Civil Engineering Study Structural Series 96-5

INRESB-3D-SUPII
Program Listing for PC

GENERAL PURPOSE PROGRAM FOR INELASTIC ANALYSIS OF RC AND STEEL BUILDING SYSTEMS FOR 3D STATIC AND DYNAMIC LOADS AND SEISMIC EXCITATIONS

by

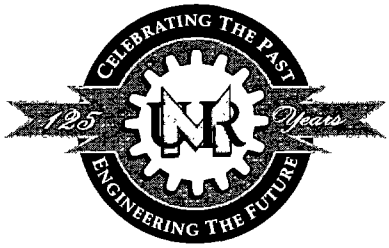
Franklin Y. Cheng
Curators' Professor

Jeng-Fuh Ger
Former Ph.D. Student and Post-Doctoral Fellow

Dan Li
Ph.D. Candidate

J.S. Yang
Former Ph.D. Student

Department of Civil Engineering
University of Missouri-Rolla
Rolla, MO 65409-0030



Report Series Prepared for the National Science Foundation
under Grants No. NSF BCS 9001494 and NSF MSS 9214664

ABSTRACT

INRESB-3D-SUPII (PC version) is a general purpose program written in FORTRAN and containing about 30,000 statements. While the PC and supercomputer versions have identical functions, PC solution capacity is limited by its memory.

INRESB-3D-SUPII is suitable for analyzing elastic and inelastic building systems subject to static loads, dynamic forces, multi-component earthquake motion, and pseudo-static cyclic loads. Also, the program is capable of calculating inelastic post-buckling behavior of steel members and systems, natural frequency, and spectral response analysis. A joint-based system is used to define the geometry of a structure. Structural members may be elastic 3D prismatic beams, nonlinear reinforced concrete shear walls, nonlinear springs, inelastic 3D-beam-column elements, finite-segment elements, and nonlinear bracing elements.

INRESB-3D-SUPII has five main functions. They are 1) elastic static analysis with multiple load cases based on control program SOL01; 2) elastic or inelastic analysis of a structure subjected to multiple ground acceleration (one-, two-, or three-dimensional seismic input) based on SOL02; 3) calculation of either natural frequencies and mode shape or buckling load and mode shape for an elastic structure based on SOL03; 4) calculation of nonlinear static cyclic response for a given loading pattern which consists of joint loads, imposed displacements or element loads for based on SOL04; and 5) calculation of maximum response for an elastic structure subject to pseudo-dynamic force obtained from the response spectrum based on SOL05.

INRESB-3D-SUPII pertains to analysis of building systems above ground level. Any connection with basements is treated as a boundary condition fixed, hinged, or with a spring.

Detail instruction of using the program is published in Civil Engineering Study Structural Series 96-3, INRESB-3D--SUPII, User's Manual.

OPERATION PROCEDURE

INRESB-3D-SUPII is written in FORTRAN and contains about 30,000 statements. In its PC version, this program comprises five independent programs: SOL01 (FEM1), SOL02 (FEM2), SOL03 (FEM3), SOL04 (FEM4), and SOL05 (FEM5). Five BAT files, LI1, LI2, LI3, LI4, and LI5, show the information needed to construct each program. These files are:

LI1.BAT

LINK FEM1+SOL01+HYST12+ELE09+STRUCT+ELELIB+MATLIB+MAEL1
+MAEL2+ELE01+MAT01+SKYLIN /ST:0X1000

LI2.BAT

LINK FEM2+SOL02+SOL02A+HYST12+ELE09+STRUCT+ELELIB+MATLIB+
MAEL1+MAEL2+ELE01+MAT01+SKYLIN /ST:0X1000

LI3.BAT

LINK FEM3+SOL03+SOL03A+HYST12+ELE09+STRUCT+ELELIB+MATLIB+
MAEL1+MAEL2+ELE01+MAT01+SKYLIN /ST:0X1000

LI4.BAT

LINK FEM4+SOL04+SOL04A+HYST12+ELE09+STRUCT+ELELIB+MATLIB+
MAEL1+MAEL2+ELE01+MAT01+SKYLIN /ST:0X1000

LI5.BAT

LINK FEM5+SOL05+SOL05A+HYST12+ELE09+STRUCT+ELELIB+MATLIB+
MAEL1+MAEL2+ELE01+MAT01+SKYLIN /ST:0X1000

ACKNOWLEDGMENTS

This is the second of two final reports on a research project sponsored by the National Science Foundation under grant numbers NSF BCS 9001494 and NSF MSS 9214664. This support is gratefully acknowledged. The authors gratefully acknowledge excellent service at University of Missouri Rolla computer center and the Cornell National Supercomputer Facility. The authors also gratefully acknowledge the advice and encouragement from Dr. S. C. Liu and Dr. K. P. Chong.

```
C-----------------------------------------------------------------------CFEM10010
C                                                                       CFEM10020
C   PROGRAM FEM - ANALYSIS OF STRUCTURES                                CFEM10030
C                                                                       CFEM10040
C-----------------------------------------------------------------------CFEM10050
C                                                                       CFEM10060
C   NOTE: USE IBM FORTVS2 COMPILER OPTION AUTODBL(DBLPAD4)              CFEM10070
C         TO COMPILE THIS PROGRAM IN DOUBLE PRECISION                   CFEM10080
C                                                                       CFEM10090
C   NOTE: USE IBM FORTVS2 COMPILER OPTION OPT(2)                        CFEM10100
C         TO OPTIMIZE THE CODE FOR FASTER EXECUTION                     CFEM10110
C                                                                       CFEM10120
C-----------------------------------------------------------------------CFEM10130
      CHARACTER*80 OPTION,INPUT,BLANK                                   FEM10140
      CHARACTER*8  TIME1,DATE1                                          FEM10150
      CHARACTER*1  CHR(0:25)                                            FEM10160
      LOGICAL TEST,HEAD,AXIAL                                           FEM10170
$INCLUDE:'ZCOMN'                                                        FEM10180
      DATA CHR/'A','B','C','D','E','F','G','H','I','J','K','L','M',    FEM10190
     &         'N','O','P','Q','R','S','T','U','V','W','Z','Y','Z'/    FEM10200
C                                                                       FEM10210
C-----------------------------                                          FEM10220
C-- ECHO THE INPUT.......   --                                          FEM10230
C-----------------------------                                          FEM10240
      I=55                                                              FEM10250
      L=0                                                               FEM10260
9995  READ(105,('(A)'),END=9996) INPUT                                  FEM10270
      IF (INDEX(INPUT,'NOECHO').NE.0) GO TO 9996                        FEM10280
      I=I+1                                                             FEM10290
      L=L+1                                                             FEM10300
      IF (  I.GT.55 ) THEN                                              FEM10310
         WRITE(108,9994) (KK,KK=1,8)                                    FEM10320
         I=0                                                            FEM10330
      ENDIF                                                             FEM10340
      WRITE(108,9997) L,INPUT                                           FEM10350
      IF (INDEX(INPUT,'STOP'  ).EQ.0) GO TO 9995                        FEM10360
9994  FORMAT ('1    ECHO OF INPUT DATA '//                              FEM10370
     &        1X,3X,'LINE  ',8('....:...',I1,'0'))                      FEM10380
9997  FORMAT (1X,I7,': ',A)                                             FEM10390
9996  REWIND(105)                                                       FEM10400
C                                                                       FEM10410
C-----------------------------                                          FEM10420
C-- INITIALIZE DATA.......   --                                         FEM10430
C-----------------------------                                          FEM10440
      TITLE(1)=' '                                                      FEM10450
      TITLE(2)=' '                                                      FEM10460
      BLANK  =' '                                                       FEM10470
      BUG =.FALSE.                                                      FEM10480
      DOUBLE =.FALSE.                                                   FEM10490
      IBUG=0                                                            FEM10500
      MAXEDF=24                                                         FEM10510
      MD(1)=1                                                           FEM10520
      DO 1 I=2,(MAXEDF+1)                                               FEM10530
    1   MD(I)=MD(I-1)+(I-1)                                             FEM10540
      IZ=0                                                              FEM10550
C                                                                       FEM10560
   20 READ(105,*,END=1000) OPTION                                       FEM10570
      RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                    FEM10580
C                                                                       FEM10590
C--------------------------                                             FEM10600
C-- DEFINE THE STRUCTURE ---                                            FEM10610
C--------------------------                                             FEM10620
      IF (INDEX(OPTION,'STR').NE.0)  THEN                               FEM10630
C ------------------------------                                        FEM10640
         IZ=1                                                           FEM10650
         TITLE(1)=' '                                                   FEM10660
         TITLE(2)=' '                                                   FEM10670
C ------- ------------------------                                     FEM10680
         GRAV=386.4                                                     FEM10690
         READ(105,*,END=1000) TITLE(1)                                  FEM10700
         CALL TIDAE(TIME1,DATE1)                                        FEM10710
         WRITE(TITLE(1)(80:),45) TIME1,DATE1                            FEM10720
         RATIZ=100 *REAL(IZ)/REAL(MAXZ)                                 FEM10730
         WRITE(*,52)' BULIDING STRUCTURAL MODEL --- ---------------*',  FEM10740
     &              IZ,RATIZ                                            FEM10750
         WRITE(*,51) TITLE(1)(1:70)                                     FEM10760
         CALL STRUCT                                                    FEM10770
         IZREL=IZ                                                       FEM10780
         RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                 FEM10790
C                                                                       FEM10800
C---------------------------------                                      FEM10810
C-- DEFINE THE LOADING AND SOLVE --                                     FEM10820
C---------------------------------                                      FEM10830
      ELSE IF (INDEX(OPTION,'SOL') NE.0)  THEN                          FEM10840
         READ(105,*,END=1000) TITLE(2)                                  FEM10850
         CALL TIDAE(TIME1,DATE1)                                        FEM10860
         WRITE(TITLE(2)(80:),45) TIME1,DATE1                            FEM10870
         WRITE(*,52)' SOLVING STRUCTURAL MODEL ------------------*',    FEM10880
     &              IZ,RATIZ                                            FEM10890
         WRITE(*,51) TITLE(1)(1:70)                                     FEM10900
         WRITE(*,51) TITLE(2)(1:70)                                     FEM10910
         CALL SOLN(OPTION)                                              FEM10920
         RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                 FEM10930
C                                                                       FEM10940
C---------------------------------                                      FEM10950
C-- STOP... END OF PROGRAM     --                                       FEM10960
C---------------------------------                                      FEM10970
      ELSE IF (INDEX(OPTION,'STOP').NE.0)  THEN                         FEM10980
         WRITE(*,52)' NORMAL STOP ------------- ----------------*',     FEM10990
     &              IZ,RATIZ                                            FEM11000
         STOP                                                           FEM11010
C                                                                       FEM11020
C---------------------------------                                      FEM11030
C--  READ RESULTS FROM INPUT FILE ---                                   FEM11040
C---------------------------------                                      FEM11050
      ELSE IF (INDEX(OPTION,'READ') NE.0)  THEN                         FEM11060
  140    CALL GETINT(OPTION,'UNIT=',IUNIT,0, TRUE ,.TRUE  )            FEM11070
         CALL GETINT(OPTION, 'INC=',INC  ,1, TRUE,.FALSE.)              FEM11080
         IF (IUNIT GT.0) THEN                                           FEM11090
           WRITE(*,52) ' READING DATA FROM FILE --------- ---------  *'FEM11100
     &               IZ,RATIZ                                           FEM11110
           IOPT=4                                                       FEM11120
           CALL DMPDAT(IOPT,IWRITE,TO,DT)                               FEM11130
           GO TO 140                                                    FEM11140
         ENDIF                                                          FEM11150
         RATIZ=100 *REAL(IZ)/REAL(MAXZ)                                 FEM11160
C                                                                       FEM11170
C------------------------                                               FEM11180
C-- SET BUG OPTION     --                                               FEM11190
C------------------------                                               FEM11200
      ELSE IF (TEST(OPTION,'BUG-',.FALSE.))  THEN                       FEM11210
         CALL GETCHR(OPTION,'BUG-',INPUT, TRUE., FALSE )                FEM11220
         IBUG=0                                                         FEM11230
         WRITE(OPTION,160) INPUT(1:70)                                  FEM11240
         DO 150 I=0,23                                                  FEM11250
           IF (TEST(INPUT,CHR(I), FALSE.)) IBUG=IBSET(IBUG,I)           FEM11260
  150    CONTINUE                                                       FEM11270
  160    FORMAT ('BUG-',A)                                              FEM11280
         WRITE(*,52)' SET BUG OPTION: --------- ---------- ----*',      FEM11290
     &              IZ,RATIZ                                            FEM11300
         WRITE(*,51)  OPTION(1:70)                                      FEM11310
         WRITE(108,51)  'SET BUG OPTION. '                              FEM11320
         WRITE(108,51)  OPTION(1:70)                                    FEM11330
C                                                                       FEM11340
C------------------------                                               FEM11350
C-- SET BUG OPTION     --                                               FEM11360
C------------------------                                               FEM11370
      ELSE IF (INDEX(OPTION,'BUG').NE.0)  THEN                          FEM11380
         WRITE(*,51) ('SET BUG OPTION  '//OPTION(1:54) )                FEM11390
         BUG= TRUE                                                      FEM11400
         IF (INDEX(OPTION,'NOBUG') NE.0) BUG= FALSE                     FEM11410
C                                                                       FEM11420
C---------------------------------------                                FEM11430
C-- RELEASE MEMORY FROM LAST SOLUTION --                                FEM11440
C---------------------------------------                                FEM11450
      ELSE IF (INDEX(OPTION,'RELEASE').NE.0)  THEN                      FEM11460
         WRITE(108,52)'-------------------------  ---------- ----*'     FEM11470
         WRITE(*,52)' RELEASING MEMORY ------------  ------ ----*'     FEM11480
         WRITE(108,52)' RELEASING MEMORY ----- -------  - ----------*'  FEM11490
         IF (INDEX(OPTION,'ELEMENT').NE.0)  THEN                        FEM11500
         WRITE(108,52)' RESETING ELEMENT FORCES ------------ ------*'   FEM11510
         WRITE(108,52)'-------------------------------------------*'    FEM11520
           DO 170 IELNO=1,NELMT                                         FEM11530
             LSTTYP=0                                                   FEM11540
             HEAD =.FALSE.                                              FEM11550
  170        CALL ELELIB                                                FEM11560
     &     (13,LSTTYP,.FALSE.,IREL,HEAD,INPUT,EESE,EPSE,DAMAGE,DUCFAG,  FEM11570
     &      IELNO,IELDOF,KGDOF,PGECM,2 ,AXIAL,IZDSP,IZDLOA,MSTGR)       FEM11580
         ELSE                                                           FEM11590
         WRITE(108,52)'-- ---------- ----------------------------*'     FEM11600
         ENDIF                                                          FEM11610
         IZ=IZREL                                                       FEM11620
           IZACC =0                                                     FEM11630
           IZDACC=0                                                     FEM11640
           IZDDSP=0                                                     FEM11650
           IZDISP=0                                                     FEM11660
           IZDLOA=0                                                     FEM11670
           IZDVEL=0                                                     FEM11680
           IZFUB =0                                                     FEM11690
           IZLOAD=0                                                     FEM11700
           IZVEL =0                                                     FEM11710
           RATIZ=100.*REAL(IZ)/REAL(MAXZ)                               FEM11720
C                                                                       FEM11730
C---------------------------------------                                FEM11740
C-- SAVE MEMORY FOR A RESTART         --                                FEM11750
C---------------------------------------                                FEM11760
      ELSE IF (INDEX(OPTION,'SAVE').NE.0)  THEN                         FEM11770
         WRITE(*,52)' SAVING DATA ------------ ------- ------- -*',     FEM11780
     &              IZ,RATIZ                                            FEM11790
         WRITE(*,51) TITLE(1)(1:70)                                     FEM11800
         WRITE(*,51) TITLE(2)(1:70)                                     FEM11810
         WRITE(108,200)                                                 FEM11820
  200    FORMAT(5X,'******** WRITING DATA TO UNIT FT09 ********')       FEM11830
         OPEN(UNIT=9,ACCESS='SEQUENTIAL',STATUS='UNKNOWN',              FEM11840
     &        FORM='UNFORMATTED')                                       FEM11850
         WRITE(9)   TITLE                                               FEM11860
         WRITE(9)   IZ    ,IZREL ,                                      FEM11870
     &    NDOF  ,NCOND ,NFREE ,NREST ,MAXNOD,NMAT  ,NELMT ,NNODE ,     FEM11880
     &    NLOAD ,NSOLN ,MAXEDF,MD,NCOS,MAXELD,NELD,                    FEM11890
     &    IZID  ,IZIDOF,IZCORD,IZCOS ,IZCNST,IZMAT ,IZELE ,            FEM11900
     &    IZJFLG,IZJCOS,                                               FEM11910
     &    IZKE  ,IZLM  ,IZSTIF,IZMD  ,IZLOAD,IZDISP,IZIELD,IZELD       FEM11920
         DO 210 J=1,IZ,256                                              FEM11930
           X=J+255                                                      FEM11940
           K=MIN(K,IZ)                                                  FEM11950
  210      WRITE(9) (Z(I),I=J,K)                                        FEM11960
         CLOSE (UNIT=9)                                                 FEM11970
         RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                 FEM11980
C                                                                       FEM11990
C---------------------------------------                                FEM12000
C-- RESTART PROGRAM                   --                                FEM12010
C---------------------------------------                                FEM12020
         WRITE(*,52)' RETREVING DATA ------------ ---------- *',       FEM12030
     &              IZ,RATIZ                                            FEM12040
       ELSE IF (INDEX(OPTION,'RESTART').NE.0)  THEN                     FEM12050
         WRITE(108,300)                                                 FEM12060
  300    FORMAT(5X,'******** READING DATA FROM UNIT FT09 ********')     FEM12070
         OPEN(UNIT=9,ACCESS='SEQUENTIAL',STATUS='OLD',                  FEM12080
     &        FORM='UNFORMATTED')                                       FEM12090
         READ  (9)   TITLE                                              FEM12100
         READ  (9)   IZ    ,IZREL ,                                     FEM12110
     &    NDOF  ,NCOND ,NFREE ,NREST ,MAXNOD,NMAT  ,NELMT ,NNODE ,     FEM12120
     &    NLOAD ,NSOLN ,MAXEDF,MD,NCOS,MAXELD,NELD,                    FEM12130
     &    IZID  ,IZIDOF,IZCORD,IZCOS ,IZCNST,IZMAT ,IZELE ,            FEM12140
     &    IZJFLG,IZJCOS,                                               FEM12150
     &    IZKE  ,IZLM  ,IZSTIF,IZMD  ,IZLOAD,IZDISP,IZIELD,IZELD       FEM12160
         DO 310 J=1,IZ,256                                              FEM12170
           X=J+255                                                      FEM12180
           K=MIN(K,IZ)                                                  FEM12190
  310      READ  (9) (Z(I),I=J,K)                                       FEM12200
         CLOSE (UNIT=9)                                                 FEM12210
         WRITE(*,52)' DATA RETRIVED  ----------------------------*',    FEM12220
     &              IZ,RATIZ                                            FEM12230
         WRITE(*,51) TITLE(1)(1:70)                                     FEM12240
         WRITE(*,51) TITLE(2)(1:70)                                     FEM12250
         RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                 FEM12260
C                                                                       FEM12270
C---------------------------------------                                FEM12280
C-- INVALID OPTION                    --                                FEM12290
C---------------------------------------                                FEM12300
      ELSE                                                              FEM12310
         IF (TEST(OPTION,'NOECHO',.FALSE.) ) GO TO 20                   FEM12320
         IF (OPTION.EQ.BLANK) GO TO 20                                  FEM12330
         WRITE(108,30) OPTION                                           FEM12340
         WRITE(*,51) ('INVALID OPTION: '//OPTION(1:54))                 FEM12350
      ENDIF                                                             FEM12360
      GO TO  20                                                         FEM12370
C                                                                       FEM12380
   45 FORMAT(' TIME. ',A,', DATE. ',A)                                  FEM12390
   50 FORMAT (A1,' STRUCTURE....  ',A,/1X,' SOLUTION ..    ',A,/)       FEM12400
   51 FORMAT (1X,A)                                                     FEM12410
   52 FORMAT (1X,'----',A45,'*',',' IZ=',I7,' MEM=',F6.3,'%')           FEM12420
   10 FORMAT ('1 INVALID OPTION= ',A)                                   FEM12430
  110 FORMAT (' ZMATRX(',I6,'):',I15,1P,G20.10,',',G20.10)              FEM12440
 1000 STOP                                                              FEM12450
      END                                                               FEM12460
C                                                                       TIDA0010
C ----- TAKE PRESENT TIME AND DATE ----                                 TIDA0020
C                                                                       TIDA0030
      SUBROUTINE TIDAE(TIME1,DATE1)                                     TIDA0040
       INTEGER*2 IYR,IMON,IDAY,IHR,IMIN,ISEC,I100TH                     TIDA0050
       CHARACTER*8 TIME1,DATE1                                          TIDA0060
       CALL GETDAT(IYR,IMON,IDAY)                                       TIDA0070
       IYR=IYR-1900                                                     TIDA0080
       CALL GETTIM(IHR,IMIN,ISEC,I100TH)                                TIDA0090
       WRITE(DATE1,'(i2,1h/,i2,1h/,i2)') IMON,IDAY,IYR                  TIDA0100
       WRITE(TIME1,'(i2,1h:,i2,1h:,i2)') IHR,IMIN,ISEC                  TIDA0110
       RETURN                                                           TIDA0120
      END                                                               TIDA0130
C                                                                       SOLN0010
C-----------------------------------------------------------------------SOLN0020
C                                                                       SOLN0030
      SUBROUTINE SOLN(OPTION)                                           SOLN0040
      CHARACTER*80 OPTION                                               SOLN0050
$INCLUDE:'ZCOMN'                                                        SOLN0060
      STEPID=' '                                                        SOLN0070
C                                                                       SOLN0080
C--------------------------                                             SOLN0090
C- DETERMINE SOLUTION NO  --                                            SOLN0100
C--------------------------                                             SOLN0110
      CALL GETINT(OPTION,'SOL',NSOLN,0,.TRUE.,.FALSE.)                  SOLN0120
C                                                                       SOLN0130
C------------------------                                               SOLN0140
C- CHOOSE THE SOLUTION --                                               SOLN0150
C------------------------                                               SOLN0160
      IF (NSOLN.EQ.1) THEN                                              SOLN0170
            CALL SOL01                                                  SOLN0180
C     ELSE IF (NSOLN.EQ.2) THEN                                         SOLN0190
C           NLOAD=1                                                     SOLN0200
C           MAXELD=0                                                    SOLN0210
C           CALL SOL02                                                  SOLN0220
C     ELSE IF (NSOLN.EQ.3) THEN                                         SOLN0230
C           NLOAD=1                                                     SOLN0240
C           MAXELD=0                                                    SOLN0250
C           CALL SOL03                                                  SOLN0260
C     ELSE IF (NSOLN.EQ.4) THEN                                         SOLN0270
C           NLOAD=1                                                     SOLN0280
C           CALL SOL04                                                  SOLN0290
C     ELSE IF (NSOLN.EQ.5) THEN                                         SOLN0300
C           NLOAD=1                                                     SOLN0310
C           MAXELD=0                                                    SOLN0320
C           CALL SOL05                                                  SOLN0330
```

```
      ELSE                                                              SOLN0340
            WRITE(108,*) 'INVALID SOLN#',NSOLN                          SOLN0350
            STOP 'INVALID SOLUTION NUMBER'                              SOLN0360
      ENDIF                                                             SOLN0370
      RETURN                                                            SOLN0380
      END                                                               SOLN0390
C                                                                       MSOL0010
C*************************************************************************MSOL0020
C  MATRIX SOLUTION OF EQUATION AX=P, WHERE THE UPPER TRIANGULAR MATRIX  MSOL0030
C  OF THE SYMMETRIC MATRIX A IS STORED BY SKYLINES. P IS STORED AS A    MSOL0040
C  FULL MATRIX.  THE MATRIX P IS REPLACED BY X AFTER BACK SUBSTITUTION  MSOL0050
C  IS COMPLEATE.  BOTH A AND P ARE ALTERED ...                          MSOL0060
C  GUYAN OPTION: IF GUYAN IS .TRUE. MASS MATRIX IS ALSO CONDENSED       MSOL0070
C*************************************************************************MSOL0080
      SUBROUTINE MSOLL(IOPT,PRINT,N,IMD,LROW,NLC,L1,L2,MD,A,P,FACTOR,   MSOL0090
     &                 GUYAN,MASS,MDMASS,IMDMAS,KGCOND,KG,MDKG,IMDKG)   MSOL0100
      REAL    MASS(IMDMAS),KG(IMDKG)                                    MSOL0110
      LOGICAL GUYAN,PRINT,KGCOND                                        MSOL0120
      DIMENSION A(IMD),FACTOR(N),P(LROW,NLC),MD(N+1)                    MSOL0130
      DIMENSION MDMASS(N+1),MDKG(N+1)                                   MSOL0140
C                                                                       MSOL0150
      IJI(I,J)=MD(J)-J+I                                                MSOL0160
      IJM(I,J)=MDMASS(J)-J+I                                            MSOL0170
      IJKG(I,J)=MDKG(J)+J-I                                             MSOL0180
C ---- ------------------------------------------------------------------MSOL0190
C   MATRIX STORAGE SCHEME:                                              MSOL0200
C  THE MATRIX A IS BANDED AND SYMMETRIC. THUS ONLY THE SKYLINE ABOVE    MSOL0210
C   AND THE MAIN DIAGONAL NEEDS TO BE STORED.                           MSOL0220
C   THE MATRIX A IS STORED IN A LINEAR ARRAY BY COLUMNS.  THE VALUE ON  MSOL0230
C   THE MAIN DIAGONAL IS STORED IN THE LINEAR ARRAY FIRST.              MSOL0240
C   ADDRESSES OF THE LINEAR ARRAY ELEMENTS CONTAINING THE MAIN DIAGONAL MSOL0250
C   TERMS ARE STORED IN MD                                              MSOL0260
C   EXAMPLE: LET B BE THE 5X5 FULL SYMETRIC MATRIX BELOW...             MSOL0270
C      :  1  :  2  :  4  :  11 :                                        MSOL0280
C      :     :  3  :  5  :  12 :  NOTE: B(1,3),B(1,4) AND B(2,4) ARE ZERO.MSOL0290
C  B=  :     :  4  :  7  :  10 :                                        MSOL0300
C      :  SYMM.    :  6  :  9  :  MD= 1, 2, 4, 6, 8, 13                MSOL0310
C      :           :     :  8  :                                        MSOL0320
C   THUS, B(3,3)=A( MD(3) )= A(4)                                       MSOL0330
C         B(I,J)=A( MD(J)+J-I )  IF J>I AND (MD(J)-J+I)<MD(J-1)         MSOL0340
C               =0               IF J>I AND (MD(J)+J-I)>MD(J+1)       MSOL0350
C  MATRIX P IS NOT SYMMETRIC, AND IS STORED AS A FULL MATRIX            MSOL0360
C-------------------------------------------------------------------------MSOL0370
C   VARIABLE TABLE:                                                     MSOL0380
C   IOPT  - OPTION NUMBER                                               MSOL0390
C   N     - NUMBER OF DEGREES OF FREEDOM OF A MATRIX                    MSOL0400
C   NLC   - NUMBER OF LOAD CASES FOR THE P MATRIX                       MSOL0410
C   L1    - FIRST ROW TO BE WORKED WITH                                 MSOL0420
C   L2    - LAST ROW TO BE WORKED WITH                                  MSOL0430
C   MD    - ADDRESS OF THE MAIN DIAGONAL TERMS OF A                     MSOL0440
C   P     - LACD MATRIX ON INPUT, SOLN (X) OF AX=P ON COMPLEATION OF    MSOL0450
C           IOPT=4                                                      MSOL0460
C   FACTOR- TEMPORARY STORAGE MATRIX                                    MSOL0470
C   GUYAN - GUYAN REDUCTION FLAG FOR MASS MATRIX                        MSOL0480
C   MASS  - MASS MATRIX                                                 MSOL0490
C   MDMASS- MASS MATRIX ADDRESSES OF MAIN DIAGONAL ELEMENTS             MSOL0500
C   KGCOND- FLAG FOR CONDENSING OUT KG                                  MSOL0510
C   KG    - GEOMETRIC STIFFNESS MATRIX                                  MSOL0520
C   MDKG  - KG MATRIX ADDRESSES OF MAIN DIAGONAL ELEMENTS               MSOL0530
C   L     - ROW NUMBER FOR ELIMINATION                                  MSOL0540
C   I     - ROW NUMBER                                                  MSOL0550
C   J     - COLUMN NUMBER                                               MSOL0560
C-------------------------------------------------------------------------MSOL0570
C                                                                       MSOL0580
      L1=MAX(L1,1)                                                      MSOL0590
      L2=MIN(L2,N)                                                      MSOL0600
C                                                                       MSOL0610
      GO TO (10,110,210,310) IOPT                                       MSOL0620
C                                                                       MSOL0630
   10 CONTINUE                                                          MSOL0640
C--------- IOPT=1, REDUCE A AND P ------------------------------------- MSOL0650
C          GAUSSIAN ELIMINIATION IS USED TO REDUCE THE A AND P MATRICIESMSOL0660
C          FROM ROW L1 TO ROW L2.  IF L1 IS NOT 1, IT IS ASSUMED THAT   MSOL0670
C          A AND P ARE ALLREADY REDUCED FROM 1 TO L1 .                  MSOL0680
C-------------------------------------------------------------------------MSOL0690
      IF (PRINT) THEN                                                   MSOL0700
        WRITE(108,500) 'IOPT=1, REDUCE STIFFNESS AND LOADS',L1,L2,N     MSOL0710
        CALL LMATRX(A,MD,N,IMD ,'INPUT STIFFNESS')                      MSOL0720
        CALL WMATRX(P,N ,NLC,'INPUT LOAD')                              MSOL0730
        IF (GUYAN ) CALL LMATRX(MASS,MDMASS,N,IMDMAS,'INPUT MASS')      MSOL0740
        IF (KGCOND) CALL LMATRX(KG  ,MDKG  ,N,IMDKG ,'INPUT KG  ')     MSOL0750
      ENDIF                                                             MSOL0760
C                                                                       MSOL0770
      DO 100 L=L1,L2                                                    MSOL0780
        IF (L.EQ.N) GO TO 100                                           MSOL0790
C                                                                       MSOL0800
C------- TEST FOR ZERO PIVOT.. ..... .....                              MSOL0810
        IF (A(MD(L)).EQ.0) THEN                                         MSOL0820
          WRITE(108,2000) L,MD(L),A(MD(L))                              MSOL0830
          STOP                                                          MSOL0840
        ENDIF                                                           MSOL0850
C                                                                       MSOL0860
C--- --- CALCULATE THE FACTORS EACH ROW IS MULTIPLIED BY                MSOL0870
        DO 30 I=(L+1),N                                                 MSOL0880
          LI=IJI(L,I)                                                   MSOL0890
          IF ( LI.LT.MD(I+1) ) THEN                                     MSOL0900
           FACTOR(I)=A(LI) /A( MD(L) )                                  MSOL0910
          ELSE                                                          MSOL0920
           FACTOR(I)=0                                                  MSOL0930
          ENDIF                                                         MSOL0940
   30     CONTINUE                                                      MSOL0950
C                                                                       MSOL0960
C---- -- REDUCE A BY COLUMNS (J) EACH ROW (I)...                       MSOL0970
        DO 40 J=(L+1),N                                                 MSOL0980
          LJ=IJI(L,J)                                                   MSOL0990
          IF (LJ GE MD(J+1)) GO TO 40                                   MSOL1000
          DO 35 I=(L+1),J                                               MSOL1010
            IJ=IJI(I,J)                                                 MSOL1020
            A(IJ)=A(IJ)-FACTOR(I)*A(LJ)                                 MSOL1030
   35       CONTINUE                                                    MSOL1040
   40   CONTINUE                                                        MSOL1050
C                                                                       MSOL1060
C---- ---- REDUCE P BY COLUMNS (J) EACH ROW (I)..                       MSOL1070
        DO 50 J=1,NLC                                                   MSOL1080
          IF (P(L,J).EQ 0) GO TO 50                                     MSOL1090
          DO 45 I=(L+1),N                                               MSOL1100
   45       P(I,J)=P(I,J)-FACTOR(I)*P(L,J)                             MSOL1110
   50   CONTINUE                                                        MSOL1120
C                                                                       MSOL1130
C-------- REDUCE MASS MATRIX BY GUYAN REDUCTION                         MSOL1140
      IF (GUYAN) THEN                                                   MSOL1150
C                                                                       MSOL1160
C------ .. M22=M22 - K21/K11 * M11 + 1/K11 * K12                        MSOL1170
        RM=MASS(MDMASS(L))                                              MSOL1180
        IF (RM.NE.0) THEN                                               MSOL1190
              DO 60 J=L+1,N                                             MSOL1200
                   IF (FACTOR(J) EQ.0) GO TO 60                         MSOL1210
                   I1=MAX(L+1,J-(MDMASS(J+1)-MDMASS(J)) )               MSOL1220
                   DO 55 I=I1,J                                         MSOL1230
                    IJ=IJM(I,J)                                         MSOL1240
   55               MASS(IJ)=MASS(IJ)+FACTOR(I)*RM*FACTOR(J)            MSOL1250
   60         CONTINUE                                                  MSOL1260
        ENDIF                                                           MSOL1270
C                                                                       MSOL1280
C------ .. M22=M22 - K21/K11 * M12                                      MSOL1290
        DO 66 J=L+1,N                                                   MSOL1300
             LJ=IJM(L,J)                                                MSOL1310
             IF (LJ.GE.MDMASS(J+1)) GO TO 66                            MSOL1320
             IF (MASS(LJ).EQ.0) GO TO 66                                MSOL1330
             DO 65 I=L+1,J                                              MSOL1340
               IJ=IJM(I,J)                                              MSOL1350
               IF (IJ.LT.MDMASS(J-1))                                   MSOL1360
     &         MASS(IJ)=MASS(IJ)-FACTOR(I)*MASS(LJ)                     MSOL1370
   65        CONTINUE                                                   MSOL1380
   66   CONTINUE                                                        MSOL1390
C                                                                       MSOL1400
C....    . M22=M22 - M21 * 1/K11 * K12                                  MSOL1410
        DO 70 J=L+1,N                                                   MSOL1420
             IF (FACTOR(J) EQ.0) GO TO 70                               MSOL1430
             DO 69 I=L+1,J                                              MSOL1440
                  LI=IJM(L,I)                                           MSOL1450
                  IF (LI.GE.MDMASS(I+1)) GO TO 69                       MSOL1460
                  IJ=IJM(I,J)                                           MSOL1470
                  IF (IJ.LT.MDMASS(J-1))                                MSOL1480
     &            MASS(IJ)=MASS(IJ)-FACTOR(J)*MASS(LI)                  MSOL1490
   69        CONTINUE                                                   MSOL1500
   70   CONTINUE                                                        MSOL1510
      ENDIF                                                             MSOL1520
C                                                                       MSOL1530
C------- REDUCE GEOMETRIC STIFFNESS MATRIX                              MSOL1540
      IF (KGCOND) THEN                                                  MSOL1550
C                                                                       MSOL1560
C........ . KG22=KG22 + K21/K11 * KG11 * 1/K11 * KG12                  MSOL1570
        RKG=KG(MDKG(L))                                                 MSOL1580
        IF (RKG.NE.0) THEN                                              MSOL1590
             DO 80 J=L+1,N                                              MSOL1600
                  IF (FACTOR(J).EQ 0) GO TO 80                          MSOL1610
                  I1=MAX(L+1,J-(MDKG(J+1)-MDKG(J)) )                    MSOL1620
                  DO 75 I=I1,J                                          MSOL1630
                   IJ=IJKG(I,J)                                         MSOL1640
   75             KG(IJ)=KG(IJ)+FACTOR(I)*RKG*FACTOR(J)                MSOL1650
   80        CONTINUE                                                   MSOL1660
        ENDIF                                                           MSOL1670
C                                                                       MSOL1680
C ........ KG22=KG22 - K21/K11 * KG12                                   MSOL1690
        DO 86 J=L+1,N                                                   MSOL1700
             LJ=IJKG(L,J)                                               MSOL1710
             IF (LJ.GE.MDKG(J+1)) GO TO 86                              MSOL1720
             IF (KG(LJ).EQ.0) GO TO 86                                  MSOL1730
             DO 85 I=L+1,J                                              MSOL1740
                  IJ=IJKG(I,J)                                          MSOL1750
                  IF (IJ.LT.MDKG(J+1))                                  MSOL1760
     &            KG(IJ)=KG(IJ)-FACTOR(I)*KG(LJ)                        MSOL1770
   85        CONTINUE                                                   MSOL1780
   86   CONTINUE                                                        MSOL1790
C                                                                       MSOL1800
C ... .... KG22=KG22 - K21 * 1/K11 * K12                                MSOL1810
        DO 90 J=L+1,N                                                   MSOL1820
             IF (FACTOR(J).EQ.0) GO TO 90                               MSOL1830
             DO 89 I=L+1,J                                              MSOL1840
                  LI=IJKG(L,I)                                          MSOL1850
                  IF (LI.GE.MDKG(I+1)) GO TO 89                         MSOL1860
                  IJ=IJKG(I,J)                                          MSOL1870
                  IF (IJ.LT.MDKG(J-1))                                  MSOL1880
     &            KG(IJ)=KG(IJ)-FACTOR(J)*KG(LI)                        MSOL1890
   89        CONTINUE                                                   MSOL1900
   90   CONTINUE                                                        MSOL1910
C                                                                       MSOL1920
      ENDIF                                                             MSOL1930
C                                                                       MSOL1940
  100 CONTINUE                                                          MSOL1950
      IF (PRINT) THEN                                                   MSOL1960
        CALL LMATRX(A,MD,N,IMD,'REDUCED STIFFNESS')                     MSOL1970
        CALL WMATRX(P,N ,NLC,'REDUCED LOAD')                            MSOL1980
        IF (GUYAN)  CALL LMATRX(MASS,MDMASS,N,IMDMAS,'REDUCED MASS')    MSOL1990
        IF (KGCOND) CALL LMATRX(KG  ,MDKG  ,N,IMDKG ,'REDUCED KG')      MSOL2000
      ENDIF                                                             MSOL2010
      RETURN                                                            MSOL2020
C                                                                       MSOL2030
  110 CONTINUE                                                          MSOL2040
C--------- IOPT=2, REDUCE P ONLY ------------------------------------- MSOL2050
C          GAUSSIAN ELIMINIATION IS USED TO REDUCE THE P MATRIX         MSOL2060
C          FROM ROW L1 TO ROW L2.  IF L1 IS NOT 1, IT IS ASSUMED THAT   MSOL2070
C          A AND P ARE ALLREADY REDUCED FROM 1 TO L1 ..                 MSOL2080
C-------------------------------------------------------------------------MSOL2090
      IF (PRINT) THEN                                                   MSOL2100
        WRITE(108,500) 'IOPT=2, REDUCE LOADS ONLY',L1,L2,N              MSOL2110
        CALL LMATRX(A,MD,N,MD(N+1),'REDUCED STIFFNESS')                 MSOL2120
        CALL WMATRX(P,N ,NLC,'NEW LOAD MATRIX ')                        MSOL2130
      ENDIF                                                             MSOL2140
C                                                                       MSOL2150
C------- LOOP FOR EACH ROW (L) TO REDUCE....                            MSOL2160
      DO 160 L=L1,L2                                                    MSOL2170
C                                                                       MSOL2180
        IF (A(MD(L)) EQ.0) THEN                                         MSOL2190
          WRITE(106,2000) L,MD(L),A(MD(L))                              MSOL2200
          STOP                                                          MSOL2210
        ENDIF                                                           MSOL2220
C                                                                       MSOL2230
C       LOOP FOR EACH COLUMN (J)                                        MSOL2240
        DO 150 J=1,NLC                                                  MSOL2250
          IF (P(L,J).EQ 0) GO TO 150                                    MSOL2260
          DO 120 I=(L+1),N                                              MSOL2270
            LI=IJI(L,I)                                                 MSOL2280
            IF (LI.LT.MD(I+1)) P(I,J)=P(I,J)-A(LI)*P(L,J)/A(MD(L))      MSOL2290
  120       CONTINUE                                                    MSOL2300
  150   CONTINUE                                                        MSOL2310
  160   CONTINUE                                                        MSOL2320
      RETURN                                                            MSOL2330
C                                                                       MSOL2340
  210 CONTINUE                                                          MSOL2350
C------- -- IOPT=3, BACK SUBSTUTION  ------------------------------------MSOL2360
C          BACK SUBSTUTION IS USED TO SOLVE FOR X IN SX=V, WHERE S IS   MSOL2370
C          THE REDUCE UPPER TRIANGULAR FACTOR OF THE A MATRIX, AND V    MSOL2380
C          IS THE REDUCED FORRM OF THE P MATRIX.  THE RESULTS (X) ARE   MSOL2390
C          STORED IN P.                                                 MSOL2400
C          BACK SUBSTUTION ONLY TAKES PLACE BETWEEN ROWS L1 AND L2.     MSOL2410
C-------------------------------------------------------------------------MSOL2420
      IF (PRINT) THEN                                                   MSOL2430
        WRITE(108,500) 'IOPT=3, BACK SUBSTUTION',L1,L2,N                MSOL2440
        CALL LMATRX(A,MD,N,MD(N+1),'REDUCED STIFFNESS')                 MSOL2450
        CALL WMATRX(P,N ,NLC,'REDUCED LOAD')                            MSOL2460
      ENDIF                                                             MSOL2470
C                                                                       MSOL2480
C------ LOOP FOR EACH ROW L                                            MSOL2490
C       CALCULAGE X FOR EACH LOAD CASE, STORE IN P                      MSOL2500
        DO 250 J=L2,L1,-1                                               MSOL2510
        IF (A(MD(J)).EQ.0) THEN                                         MSOL2520
          WRITE(108,2000) J,MD(J),A(MD(J))                              MSOL2530
          STOP                                                          MSOL2540
        ENDIF                                                           MSOL2550
        DO 215 LC=1,NLC                                                 MSOL2560
  215     P(J,LC)=P(J,LC)/A(MD(J))                                      MSOL2570
        IF (J.GT.1) THEN                                                MSOL2580
         IJ=MD(J-1)                                                     MSOL2590
         NI=J-(MD(J-1)-MD(J)) +1                                        MSOL2600
         DO 220 I=NI,J-1,+1                                             MSOL2610
            IJ=IJ-1                                                     MSOL2620
            DO 220 LC=1,NLC                                             MSOL2630
              P(I,LC)=P(I,LC)-P(J,LC)*A(IJ)                             MSOL2640
  220   CONTINUE                                                        MSOL2650
        ENDIF                                                           MSOL2660
  250   CONTINUE                                                        MSOL2670
      IF (PRINT) CALL WMATRX(P,N ,NLC,'DISPLACEMENT')                   MSOL2680
C                                                                       MSOL2690
      RETURN                                                            MSOL2700
C                                                                       MSOL2710
  310 CONTINUE                                                          MSOL2720
C------- - IOPT=4, REDUCED MULTIPLICATION ---------------- ------------MSOL2730
C          SOLVE FOR V IN S*D=P WHERE S HAS BEEN REDUCED BY GAUSSIAN    MSOL2740
C          ELIMINATION.                                                 MSOL2750
C          THE RESULTING LOAD IS STORED IN P                            MSOL2760
C          THE DISPLACEMENTS ARE INPUT IN D                             MSOL2770
C          MULTIPLACATION ONLY TAKES PLACE BETWEEN ROWS L1 AND L2       MSOL2780
C--- --- ------- ------------------------------------ -- ------- ----- MSOL2790
      IF (PRINT) THEN                                                   MSOL2800
        WRITE(108,500) 'IOPT=4, REDUCED MULTIPLACATION',L1,L2,N         MSOL2810
        CALL LMATRX(A,MD,N,MD(N+1),'REDUCED STIFFNESS')                 MSOL2820
        CALL WMATRX(FACTOR,N ,1,'DISPLACEMENT')                         MSOL2830
      ENDIF                                                             MSOL2840
C                                                                       MSOL2850
C------ ONLY ONE LOAD CASE.. .                                         MSOL2860
```

```
      LC=1                                                              MSOL2870
C       ---- ZERO P - --                                               MSOL2880
        DO 320 J=L1,L2                                                  MSOL2890
          IF (A(MD(J)).EQ 0) THEN                                       MSOL2900
            WRITE(108,2000) J,MD(J),A(MD(J))                            MSOL2910
            STOP                                                        MSOL2920
          ENDIF                                                         MSOL2930
  320   P(J,LC)=0                                                       MSOL2940
C                                                                       MSOL2950
C ----------- MULTIPLY S_RED * DISP = P                                 MSOL2960
      DO 340 J=L1,N                                                     MSOL2970
        IC1=J-(MD(J-1) MD(J)) +1                                        MSOL2980
        IC2=MAX(IC1,L1)                                                 MSOL2990
        DO 340 I=J,IC2,-1                                               MSOL3000
          IJ=IJI(I,J)                                                   MSOL3010
  340     P(I,LC)=P(I,LC)+A(IJ)*FACTOR(J)                               MSOL3020
      IF (PRINT) CALL WMATRX(P,N ,1,'REDUCED LOAD')                     MSOL3030
C                                                                       MSOL3040
C------- UNREDUCE THE LOAD MATRIX    .                                 MSOL3050
        DO 360 L=L2,L1,-1                                               MSOL3060
C                                                                       MSOL3070
C       LOOP FOR EACH COLUMN (J)                                        MSOL3080
        DO 350 J=1,NLC                                                  MSOL3090
          IF (P(L,LC).EQ 0) GO TO 350                                   MSOL3100
          DO 345 I=(L+1),N                                              MSOL3110
            LI=IJI(L,I)                                                 MSOL3120
            IF (LI LT.MD(I+1)) P(I,LC)=P(I,LC)+A(LI)*P(L,LC)/A(MD(L))   MSOL3130
  345       CONTINUE                                                    MSOL3140
  350   CONTINUE                                                        MSOL3150
  360   CONTINUE                                                        MSOL3160
  400   CONTINUE                                                        MSOL3170
C                                                                       MSOL3180
      IF (PRINT) CALL WMATRX(P,N ,1,'LOAD')                             MSOL3190
  500 FORMAT (/2X,A,2X,'L1 =',I5,'   L2 =',I5,'   N:',I5)               MSOL3200
 2000 FORMAT (///1X,50('*')/                                            MSOL3210
     &        ' * ER','ROR IN SUBROUTINE MSOLL',        T51,'*'/       MSOL3220
     &        ' *  PIVOT IS LE 0 ------- ROW =',I5,     T51,'*'/       MSOL3230
     &        ' *                    STORAGE LCN =',I5,  T51,'*'/      MSOL3240
     &        ' *                              PIVOT =',1P,G15.6,T51,'*'/MSOL3250
     &        ' *  SOLUTION IS ABORTED ',               T51,'*'/       MSOL3260
     &        1X,50('*')/)                                             MSOL3270
C                                                                       MSOL3280
      RETURN                                                            MSOL3290
      END                                                               MSOL3300
C                                                                       REAC0010
C-----------------------------------------------------------------------REAC0020
C                                                                       REAC0030
      SUBROUTINE REACTN(IOPT,PRINT,NNODE,L1,L2,L3,L4,FORCE,DISPL,       REAC0040
     &                  MD,A,KM,NDOF,NLOAD,FACTOR,                      REAC0050
     &                  MAXNOD,IDOF,COORD,ID,TITLE,STEPID,HEAD,         REAC0060
     &                  NCOS,COSINE,JTCOS,JTFLG,SUM)                    REAC0070
      CHARACTER*(*) TITLE(2),STEPID                                     REAC0080
      CHARACTER*15 CD(6)                                                REAC0090
      LOGICAL PRINT,HEAD,FLAG,BTEST                                     REAC0100
     .DIMENSION A(KM),MD(NDOF+1),IDOF(MAXNOD,6),G(6)                    REAC0110
      DIMENSION FORCE(NDOF,NLOAD),DISPL(NDOF,NLOAD)                     REAC0120
      DIMENSION ID(MAXNOD),COORD(MAXNOD,3),JTFLG(MAXNOD)                REAC0130
      DIMENSION P(6),SUM(6),COSINE(3,3,NCOS),JTCOS(MAXNOD)              REAC0140
C                                                                       REAC0150
      IF (IOPT.EQ.1) THEN                                               REAC0160
C                                                                       REAC0170
C-----------------------------------------                              REAC0180
C- MODIFY LOADS FOR RESTRAINT DISPLACEMENTS ---                         REAC0190
C-----------------------------------------                              REAC0200
C CONSIDER THE PARTITIONED STIFFNESS WHERE X2 ARE THE RESTRAINT DISPL.  REAC0210
C                                                                       REAC0220
C   |P1|   | K11 | K12 | |X1|                                         REAC0230
C   |--| = |-----+-----| |--| * FACTOR                                   REAC0240
C   |P2|   | K21 | K22 | |X2|                                         REAC0250
C                                                                       REAC0260
C   THE MODIFIED FREE LOAD IS (P1-K12*X2) - K11*X1                      REAC0270
C ---------------------------------------------------------------------C REAC0280
C     DO 10 I=L1,L2                                               C     REAC0290
C     DO 10 J=1,NLOAD                                             C     REAC0300
C     DO 10 K=L3,L4                                               C     REAC0310
C  10   FORCE(I,J)=FORCE(I,J)-K(I,K)*DISPL(K,J)                   C     REAC0320
C ---------------------------------------------------------------------C REAC0330
      DO 20 K=L3,L4                                                     REAC0340
      MDK=MD(K+1)                                                       REAC0350
        DO 20 J=1,NLOAD                                                 REAC0360
        DO 10 I=L2,L1,-1                                                REAC0370
          IK=MD(K)+K-I                                                  REAC0380
          IF (IK.GE.MDK) GO TO 20                                       REAC0390
          FIJ=FORCE(I,J)                                                REAC0400
          AIK=A(IK)                                                     REAC0410
          DKJ=DISPL(K,J)                                                REAC0420
   10     FORCE(I,J)=FORCE(I,J)-A(IK)*DISPL(K,J)                        REAC0430
   20   CONTINUE                                                        REAC0440
        IF (PRINT)                                                      REAC0450
     &  CALL WMATRX(FORCE,NDOF,NLOAD,'MODIFIED LOADS')                  REAC0460
C                                                                       REAC0470
      ELSE IF (IOPT.EQ.2) THEN                                          REAC0480
C                                                                       REAC0490
C------------------------                                               REAC0500
C- SOLVE FOR REACTIONS ---                                             REAC0510
C------------------------                                               REAC0520
C CONSIDER THE PARTITIONED STIFFNESS WHERE X2 ARE THE RESTRAINT DISPL.  REAC0530
C                                                                       REAC0540
C   |P1|   | K11 | K12 | |X1|                                         REAC0550
C   |--| = |-----+-----| |--| * FACTOR                                   REAC0560
C   |P2|   | K21 | K22 | |X2|                                         REAC0570
C                                                                       REAC0580
C   THE REACTIONS ARE  P2 = P2 + ( K21*X1 + K22*X2 ) * FACTOR           REAC0590
CC  NOTE- BOTH P1 AND P2 CONTAIN FIXED END FORCES FROM MEMBER LOADS...  REAC0600
C         THUS, X21*X1 + K22*X2 IS ADDED TO P2 TO GET THE REACTIONS     REAC0610
C------ ----- ------------ ---------- ---------------------------------C REAC0620
C     DO 120 I=L3,L4                                              C     REAC0630
C     DO 120 J=1,NLOAD                                            C     REAC0640
CC      FORCE(I,J)=0.0  * P2 CONTAINS FEF'S!!                     C     REAC0650
C       DO 110 K=L1,L2                                            C     REAC0660
C 110   FORCE(I,J)=FORCE(I,J)+K(I,K)*DISPL(K,J)                   C     REAC0670
C       DO 120 K=L3,L4                                            C     REAC0680
C 120   FORCE(I,J)=FORCE(I,J)+K(I,K)*DISPL(K,J)                   C     REAC0690
C--------- - --- --------- ------------ ---------------------------C REAC0700
C                                                                       REAC0710
      DO 120 I=L3,L4                                                    REAC0720
      DO 120 J=1,NLOAD                                                  REAC0730
      DO 110 K=L2,L1, 1                                                 REAC0740
        KI=MD(I)+I-K                                                    REAC0750
        IF (KI.GE.MD(I+1)) GO TO 120                                    REAC0760
        AKI=A(KI)                                                       REAC0770
  110   FORCE(I,J)=FORCE(I,J)+A(KI)*DISPL(K,J)*FACTOR                   REAC0780
  120 CONTINUE                                                          REAC0790
C                                                                       REAC0800
      DO 140 I=L3,L4                                                    REAC0810
      DO 140 K=L3,L4                                                    REAC0820
        IF (I.LE.K) THEN                                                REAC0830
          IK=MD(K)+K-I                                                  REAC0840
          MDK=MD(K+1)                                                   REAC0850
        ELSE                                                            REAC0860
          IK=MD(I)+I-K                                                  REAC0870
          MDK=MD(I+1)                                                   REAC0880
        ENDIF                                                           REAC0890
        IF (IK.GE.MDK)  GO TO 140                                       REAC0900
        AIK=A(IK)                                                       REAC0910
        DO 130 J=1,NLOAD                                                REAC0920
  130     FORCE(I,J)=FORCE(I,J)+A(IK)*DISPL(K,J)*FACTOR                 REAC0930
  140 CONTINUE                                                          REAC0940
C                                                                       REAC0950
      IF (PRINT)                                                        REAC0960
     & CALL WMATRX(FORCE,NDOF,NLOAD,'FINAL LOADS AND REACTIONS')        REAC0970
C                                                                       REAC0980
      ELSE IF (IOPT.EQ 3 .OR. IOPT EQ 4) THEN                           REAC0990
        NF=L2                                                           REAC1000
C------------------------------                                         REAC1010
C- LOOP FOR EACH LOAD CASE  ---                                         REAC1020
C------------------------------                                         REAC1030
      DO 300 L=1,NLOAD                                                  REAC1040
C                                                                       REAC1050
C------------------------------                                         REAC1060
C- PRINT GLOBAL REACTIONS    ---                                        REAC1070
C------------------------------                                         REAC1080
C                                                                       REAC1090
      IF (PRINT) THEN                                                   REAC1100
        IF (HEAD) WRITE(108,320) TITLE(1),TITLE(2)                      REAC1110
          NREACT=0                                                      REAC1120
          IF (IOPT.NE.4) WRITE(108,321) L,STEPID                        REAC1130
          IF (IOPT.EQ.4) WRITE(108,421)                                 REAC1140
      ENDIF                                                             REAC1150
        IF (IOPT.EQ.3) THEN                                             REAC1160
          DO 190 I=1,6                                                  REAC1170
  190       SUM(I)=0                                                    REAC1180
        ENDIF                                                           REAC1190
C                                                                       REAC1200
      IF ( .NOT.(NCOS.EQ 1  AND  COSINE(1,1,1).EQ.1 .AND.COSINE(2,2,1)  REAC1210
     &     .EQ.1  AND  COSINE(3,3,1) EQ 1).AND. IOPT.EQ 4  ) GO TO 279  REAC1220
C                                                                       REAC1230
        DO 200 I=1,NNODE                                                REAC1240
C     .                                                                 REAC1250
C. ............ GET REACTNS IN JOINT COORDINATE SYSTEM                 REAC1260
          FLAG=.FALSE.                                                  REAC1270
          DO 209 J=1,6                                                  REAC1280
            IJ=IDOF(I,J)                                                REAC1290
            IF (IJ.LE.NF  OR.  BTEST(JTFLG(I),J+11)) THEN               REAC1300
              F(J)=0                                                    REAC1310
            ELSE                                                        REAC1320
              F(J)=FORCE(IJ,L)                                          REAC1330
              FLAG=.TRUE                                                REAC1340
            ENDIF                                                       REAC1350
  209     CONTINUE                                                      REAC1360
          IF (.NOT.FLAG) GO TO 200                                      REAC1370
C                                                                       REAC1380
C... .... .... TRANSFER REACTNS TO GLOBAL COORDINATE SYSTEM             REAC1390
          DO 229 I1=1,3                                                 REAC1400
            I2=I1+3                                                     REAC1410
            G(I1)=0                                                     REAC1420
            G(I2)=0                                                     REAC1430
            DO 219 J1=1,3                                               REAC1440
              J2=J1+3                                                   REAC1450
              G(I1)=G(I1) + COSINE(J1,I1,JTCOS(I))*F(J1)                REAC1460
  219         G(I2)=G(I2) + COSINE(J1,I1,JTCOS(I))*F(J2)                REAC1470
            IF (PRINT) THEN                                             REAC1480
                WRITE(CD(I1),210) G(I1)                                 REAC1490
                WRITE(CD(I2),210) G(I2)                                 REAC1500
            ENDIF                                                       REAC1510
  229       CONTINUE                                                    REAC1520
C                                                                       REAC1530
C.............. SUM REACTIONS ....... ...                              REAC1540
          IF (IOPT.EQ.3) THEN                                           REAC1550
            DO 240 I1=1,6                                               REAC1560
  240         SUM(I1)=SUM(I1) + G(I1)                                   REAC1570
            SUM(4)= SUM(4) + G(2)*COORD(I,3) - G(3)*COORD(I,2)          REAC1580
            SUM(5)= SUM(5) + G(1)*COORD(I,3) - G(3)*COORD(I,1)          REAC1590
            SUM(6)= SUM(6) - G(1)*COORD(I,2) + G(2)*COORD(I,1)          REAC1600
          ENDIF                                                         REAC1610
C                                                                       REAC1620
C.... .. ..... PRINT GLOBAL REACTIONS .  ...  .                        REAC1630
          NREACT=NREACT+1                                               REAC1640
          IF (PRINT) WRITE(108,220) ID(I),(CD(K),K=1,6)                 REAC1650
  200   CONTINUE                                                        REAC1660
        IF (IOPT.EQ.3  AND. PRINT ) WRITE(108,230) SUM                  REAC1670
  279   IF (IOPT.EQ.4 .AND. PRINT ) WRITE(108,231) SUM                  REAC1680
C                                                                       REAC1690
C-----------------------------------------------                        REAC1700
C- PRINT LOCAL JOINT COORDINATE SYSTEMS REACTIONS  ---                  REAC1710
C-----------------------------------------------                        REAC1720
C                                                                       REAC1730
      IF (.NOT. PRINT ) GO TO 300                                       REAC1740
      IF (NCOS.EQ.1 .AND. COSINE(1,1,1).EQ.1 .AND. COSINE(2,2,1).EQ.1   REAC1750
     &              .AND. COSINE(3,3,1).EQ.1 ) GO TO 300                REAC1760
C                                                                       REAC1770
      IF (HEAD.AND. NREACT.GT.20) WRITE(108,320) TITLE(1),TITLE(2)      REAC1780
        IF (IOPT NE.4) WRITE(108,322) L,STEPID                          REAC1790
        IF (IOPT.EQ 4) WRITE(108,422)                                   REAC1800
        DO 290 I=1,NNODE                                                REAC1810
C                                                                       REAC1820
C..... ....... GET REACTNS IN JOINT COORDINATE SYSTEM                   REAC1830
          FLAG=.FALSE.                                                  REAC1840
          DO 280 J=1,6                                                  REAC1850
            IJ=IDOF(I,J)                                                REAC1860
            IF (IJ.LE.NF .OR.  BTEST(JTFLG(I),J+11)) THEN               REAC1870
              CD(J)=' '                                                 REAC1880
            ELSE                                                        REAC1890
              WRITE(CD(J),210) FORCE(IJ,L)                              REAC1900
              FLAG=.TRUE.                                               REAC1910
            ENDIF                                                       REAC1920
  280     CONTINUE                                                      REAC1930
          IF (.NOT.FLAG) GO TO 290                                      REAC1940
C                                                                       REAC1950
C.. .. .. .. PRINT REACTNS IN LOCAL COORDINATE SYSTEM ..  .             REAC1960
          WRITE(108,221) ID(I),(CD(K),K=1,6),JTCOS(I)                   REAC1970
  290   CONTINUE                                                        REAC1980
  300 CONTINUE                                                          REAC1990
C                                                                       REAC2000
  210 FORMAT (1P,G15.7)                                                 REAC2010
  220 FORMAT (5X,I5,6A)                                                 REAC2020
  221 FORMAT (5X,I5,6A,I10)                                             REAC2030
  230 FORMAT (/'  ',90('-')/' SUMMATION',6G15.7)                        REAC2040
  231 FORMAT (' MAX OF ALL',89('-')/' GCS SUMM.',6G15.7 )               REAC2050
  320 FORMAT ('1 STRUCTURE. . .  ',A,/'  SOLUTION . . . . ',A)          REAC2060
  321 FORMAT (//'  GCS RESTRAINT REACTIONS,   LOADING #',I5,5X,A,/      REAC2070
     &          ' ------------------------'//6X,'NODE',7X,'FX',        REAC2080
     &          13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ'/)          REAC2090
  322 FORMAT (//'  JCS RESTRAINT REACTIONS,   LOADING #',I5,5X,A,/     REAC2100
     &          ' ------------------------'//6X,'NODE',7X,'FX',        REAC2110
     &          13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ',          REAC2120
     &          6X,'  COSINE #'/)                                     REAC2130
  421 FORMAT (//'  MAXIMUM GCS RESTRAINT REACTIONS',                   REAC2140
     &       5X,'NOTE  MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'/   REAC2150
     &          ' ------------------------------'//6X,'NODE',7X,'FX',  REAC2160
     &       13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ'/)           REAC2170
  422 FORMAT (//'  MAXIMUM JCS RESTRAINT REACTIONS ',                  REAC2180
     &       5X,'NOTE: MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'/   REAC2190
     &       ' ------------------------------'//6X,'NODE',7X,'FX',     REAC2200
     &       13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ',             REAC2210
     &       6X,'  COSINE #'/)                                        REAC2220
C                                                                       REAC2230
      ELSE                                                              REAC2240
       WRITE(108,*) 'INVALID OPTION IN SUBROUTINE  REA_CTN, IOPT=',IOPT REAC2250
      ENDIF                                                             REAC2260
      RETURN                                                            REAC2270
      END                                                               REAC2280
C                                                                       FORM0010
C-----------------------------------------------------------------------FORM0020
C                                                                       FORM0030
      SUBROUTINE FORM(IOPT)                                             FORM0040
      CHARACTER  NAME*80,ELNO*130                                       FORM0050
      LOGICAL PRINT,HEAD,FALSE,AXIAL,BTEST,BUG5,BUG57                   FORM0060
$INCLUDE:'ZCOMN'                                                        FORM0070
C                                                                       FORM0080
C------------------------------------------- ------ ------------------ FORM0090
C  MATRIX STORAGE SCHEME                                                FORM0100
C  THE STIFFNESS MATRIX (K) IS BANDED AND SYMMETRIC                     FORM0110
C  THUS ONLY THE SKYLINE ABOVE THE MAIN DIAGONAL NEEDS TO BE STORED     FORM0120
C  THE MATRIX X IS STORED IN A LINEAR ARRAY BY COLUMNS.                 FORM0130
C  THE VALUE ON THE MAIN DIAGONAL IS STORED IN THE LINEAR ARRAY FIRST   FORM0140
C  RELATIVE ADDRESSES OF THE LINEAR ARRAY ELEMENTS CONTAINING THE       FORM0150
C  MAIN DIAGONAL TERMS ARE STORED IN MD                                 FORM0160
C  THE ABSOLUTE ADDRESS OF THE FIRST ELEMENT IN THE X MATRIX IS I2STIF. FORM0170
C  THE STIFFNESS MATRIX K IS STORED IN THE Z ARRAY, Z(I2STIF)=K(1,1)    FORM0180
C  THE ABSOLUTE ADDRESS OF THE FIRST ELEMENT IS THE MD MATRIX IS I2MD   FORM0190
C  THE STIFFNESS MATRIX K IS STORED IN THE N2 ARRAY, NZ(I2MD)=MD(1)     FORM0200
```

```
C   THUS: IF I LE J (UPPER TRIANGULAR MATRIX)                          FORM0210
C         IF IZMD-1+J ( IZMD-1+J-1                                     FORM0220
C                 IJ=NZ(IZMD-1+J)+J-I   RELATIVE ADDRESS IF ELEMENT (I,J)FORM0230
C                 B(I,J)=Z(IZSTIP-1 + IJ)                              FORM0240
C           ELSE                                                       FORM0250
C                 B(I,J)=0.             : (I,J) IS ABOVE THE SKYLINE   FORM0260
C                                                                      FORM0270
C   EXAMPLE: LET B BE THE 5X5 FULL SYMETRIC MATRIX BELOW...            FORM0280
C        : 1 : 3 :   :   : 12 :                                        FORM0290
C        :   : 2 : 5 :   : 11 :  NOTE: B(1,3),B(1,4) AND B(2,4) ARE ZERO. FORM0300
C   B= :     : 4 : 7 : 10 :                                           FORM0310
C        :  SYMM.   : 6 :  9 :  MD= 1, 2, 4, 6, 8, 13                 FORM0320
C        :          :   :  8 :                                        FORM0330
C   SAY IZMD=121,IZSTIP=200                                            FORM0340
C   NZ(121)=1          Z(200)=1                                        FORM0350
C   NZ(122)=2          Z(201)=2                                        FORM0360
C   NZ(123)=4          Z(202)=3                                        FORM0370
C   NZ(124)=6          Z(203)=4                                        FORM0380
C   NZ(125)=8          Z(204)=5                                        FORM0390
C   NZ(126)=13         Z(205)=6                                        FORM0400
C                      Z(206)=7                                        FORM0410
C                      Z(206)=8                                        FORM0420
C                      Z(206)=9                                        FORM0430
C                      Z(206)=10                                       FORM0440
C                      Z(206)=11                                       FORM0450
C                      Z(206)=12                                       FORM0460
C   THUS, FOR B(3,4)  I=3, J=4                                         FORM0470
C                IJ=NZ(IZMD-1+J)+J-I + NZ(121-1+4) +4-3 = 6+4-3 =7     FORM0480
C                B(I,J)=Z(IZSTIP-1 + IJ) + Z(200-1+7) = Z(206) + 7  OK FORM0490
C   THUS, FOR B(2,4)  I=2, J=4                                         FORM0500
C                IJ=NZ(IZMD-1+J)+J-I + NZ(121-1+4) +4-2 = 6+4-2 =8     FORM0510
C                NZ(IZMD -1+J-1)+NZ(121-1+4-1)=8                       FORM0520
C                SINCE IJ GE NZ(IZMD-1+J-1)                            FORM0530
C                B(I,J)=0  (I,J) IS ABOVE THE SKYLINE..                FORM0540
C-------------------------------------------------------------------   FORM0550
C                                                                      FORM0560
      BUG5 = BTEST(IBUG,5)                                             FORM0570
      BUG57= BTEST(IBUG,7) .AND. BUG5                                  FORM0580
      LSTTYP=0                                                         FORM0590
      PRINT=.FALSE.                                                    FORM0600
      HEAD =.FALSE                                                     FORM0610
C                                                                      FORM0620
C-------------------------------------------------------------------   FORM0630
C IOPT=1 FORM STIFFNESS MATRIX                                         FORM0640
C IOPT=2 FORM GEOMETRIC STIFFNESS MATRIX                               FORM0650
C-------------------------------------------------------------------   FORM0660
C                                                                      FORM0670
      IF (BUG5)  WRITE (6,*) ' IN ROUTINE FORM IOPT=',IOPT             FORM0680
C                                                                      FORM0690
   49 FORMAT (1X,A/(' I=',I3,'  MDMASS(I)=',I11))                      FORM0700
C                                                                      FORM0710
      GO TO (100,200) IOPT                                             FORM0720
C                                                                      FORM0730
C--------------------------                                            FORM0740
C- ASSEMBLE STIFFNESS  -----                                           FORM0750
C--------------------------                                            FORM0760
  100 CONTINUE                                                         FORM0770
          IST =IZSTIP                                                  FORM0780
          IEND=IZSTIP +NZ(IZMD+NDOF) -1                                FORM0790
C                                                                      FORM0800
C---------- ZERO STIFFNESS                                            FORM0810
      DO 110 I=IST,IEND                                                FORM0820
  110   Z(I)=0                                                         FORM0830
C                                                                      FORM0840
C---------- - LOOP FOR EACH ELEMENT                                   FORM0850
      DO 120 IELNO=1,NELMT                                             FORM0860
C                                                                      FORM0870
C------ ---- GET STIFFNESS OF EACH ELEMENT                            FORM0880
      FALSE=.FALSE.                                                    FORM0890
      CALL ELELIB                                                      FORM0900
     &  (3,LSTTYP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DAMAGE,DUCFAG,       FORM0910
     &  IELNO ,IELDOF,KGDOF,PGEOM,Z  ,AXIAL,IZDISP,IZLOAD,MSTOR)       FORM0920
C                                                                      FORM0930
      IF (BUG57) THEN                                                  FORM0940
        WRITE (ELNO,15)  IELNO,(NZ(I+IZLM-1),I=1,MIN(IELDOF,18))       FORM0950
   15   FORMAT (' ELEMENT #',I6,' LM=',1816)                           FORM0960
        CALL LMATRX(Z(IZKE),MD,IELDOF,MD(IELDOF+1),ELNO)               FORM0970
      ENDIF                                                            FORM0980
C                                                                      FORM0990
      CALL FORML(Z(IZSTIP),NZ(IZMD),NDOF,NZ(IZMD+NDOF),                FORM1000
     &           Z(IZKE),MD,IELDOF,MD(IELDOF+1),NZ(IZLM),1.)           FORM1010
C                                                                      FORM1020
C---------- SUBTRACT GEOMETRIC STIFFNESS                               FORM1030
      NZ(IZKGDT+3)=NZ(IZKGDT+3)                                        FORM1040
C                                                                      FORM1050
      IF (NZ(IZKGDT+3) EQ.1) THEN                                      FORM1060
        FALSE=.FALSE.                                                  FORM1070
        CALL ELELIB                                                    FORM1080
     &  (2,LSTTYP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DAMAGE,DUCFAG,       FORM1090
     &  IELNO ,IELDOF,KGDOF,PGEOM,Z  ,AXIAL,IZDISP,IZLOAD,MSTOR)       FORM1100
C                                                                      FORM1110
        KGDOF=KGDOF                                                    FORM1120
C                                                                      FORM1130
        IF (BUG57) THEN                                                FORM1140
          WRITE (ELNO,15)  IELNO,(NZ(I+IZLMKG-1),I=1,MIN(KGDOF,18))   FORM1150
          CALL LMATRX(Z(IZKEKG),MD,KGDOF,MD(KGDOF+1),ELNO)             FORM1160
        ENDIF                                                          FORM1170
C                                                                      FORM1180
        IF (.NOT. AXIAL) GO TO 120                                     FORM1190
C           DETERMINE THE FACTOR FOR KG                                FORM1200
              IF (NZ(IZKGDT+1) EQ.1) THEN                              FORM1210
                    ACCELZ= Z(IZKGDT)                                  FORM1220
                    PGEOM=PGEOM*(1+ACCELZ)                             FORM1230
              ENDIF                                                    FORM1240
C     SWAP THE SIGN ON PGEOM FOR SUBTRACTION.....                      FORM1250
      PGEOM=-PGEOM                                                     FORM1260
C                                                                      FORM1270
      CALL FORML(Z(IZSTIP),NZ(IZMD),NDOF,NZ(IZMD+NDOF),                FORM1280
     &           Z(IZKEKG),MD,KGDOF,MD(KGDOF+1),NZ(IZLMKG),PGEOM)      FORM1290
      ENDIF                                                            FORM1300
C                                                                      FORM1310
  120   CONTINUE                                                       FORM1320
C                                                                      FORM1330
      RETURN                                                           FORM1340
C                                                                      FORM1350
C----  --- - ASSEMBLE GEOMETRIC STIFNESS                               FORM1360
  200 CONTINUE                                                         FORM1370
      IF (NZ(IZKGDT+3) NE.2 ) RETURN                                   FORM1380
          IST =IZKG                                                    FORM1390
          IEND=IZKG + NZ(IZMDKG+NDOF) - 1                              FORM1400
C                                                                      FORM1410
C------ ---- ZERO STIFFNESS                                           FORM1420
      DO 210 I=IST,IEND                                                FORM1430
  210   Z(I)=0                                                         FORM1440
C                                                                      FORM1450
C------ ----- LOOP FOR EACH ELEMENT                                    FORM1460
      DO 220 IELNO=1,NELMT                                             FORM1470
C                                                                      FORM1480
C--  ------- GET STIFFNESS OF EACH ELEMENT                             FORM1490
      FALSE=.FALSE                                                     FORM1500
      CALL ELELIB                                                      FORM1510
     &  (2,LSTTYP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DAMAGE,DUCFAG,       FORM1520
     &  IELNO ,IELDOF,KGDOF,PGEOM,Z  ,AXIAL,IZDISP,IZLOAD,MSTOR)       FORM1530
C                                                                      FORM1540
      IF (.NOT. AXIAL) GO TO 220                                       FORM1550
C                                                                      FORM1560
      IF (BUG57) THEN                                                  FORM1570
        WRITE (6,*)  'PGEOM=',PGEOM                                    FORM1580
        WRITE (ELNO,15)  IELNO,(NZ(I+IZLMKG-1),I=1,MIN(KGDOF,18))     FORM1590
        CALL LMATRX(Z(IZKEKG),MD,KGDOF,MD(KGDOF+1),ELNO)               FORM1600
      ENDIF                                                            FORM1610
C                                                                      FORM1620
      CALL FORML(Z(IZKG),NZ(IZMDKG),NDOF,NZ(IZMDKG+NDOF),             FORM1630
     &       Z(IZKEKG),MD,KGDOF,MD(KGDOF+1),NZ(IZLMKG),PGEOM)        FORM1640
  220   CONTINUE                                                       FORM1650
C                                                                      FORM1660
      RETURN                                                           FORM1670
      END                                                              FORM1680
C-------------------------------------------------------------------   CFEM20010
C                                                                      CFEM20020
C   PROGRAM FEM - ANALYSIS OF STRUCTURES                               CFEM20030
C                                                                      CFEM20040
C-------------------------------------------------------------------   CFEM20050
C                                                                      CFEM20060
C   NOTE: USE IBM FORTVS2 COMPILER OPTION AUTODBL(DBLPAD4)             CFEM20070
C         TO COMPILE THIS PROGRAM IN DOUBLE PRECISION                  CFEM20080
C                                                                      CFEM20090
C   NOTE. USE IBM FORTVS2 COMPILER OPTION OPT(2)                       CFEM20100
C         TO OPTIMIZE THE CODE FOR FASTER EXECUTION                    CFEM20110
C                                                                      CFEM20120
C-------------------------------------------------------------------   CFEM20130
      CHARACTER*80 OPTION,INPUT,BLANK                                  FEM20140
      CHARACTER*8  TIME1,DATE1                                         FEM20150
      CHARACTER*1  CHR(0:25)                                           FEM20160
      LOGICAL TEST                                                     FEM20170
$INCLUDE 'ZCOMM'                                                       FEM20180
      DATA CHR/'A','B','C','D','E','F','G','H','I','J','K','L','M',   FEM20190
     &         'N','O','P','Q','R','S','T','U','V','W','Z','Y','Z'/   FEM20200
C                                                                      FEM20210
C*------------------------                                            FEM20220
C** ECHO THE INPUT.....    **                                         FEM20230
C*------------------------                                            FEM20240
      I=55                                                             FEM20250
      L=0                                                              FEM20260
9995  READ(105,('(A)'),END=9996) INPUT                                 FEM20270
      IF (INDEX(INPUT,'NOECHO').NE.0) GO TO 9996                       FEM20280
      I=I+1                                                            FEM20290
      L=L+1                                                            FEM20300
      IF ( I.GT.55 ) THEN                                              FEM20310
         WRITE(108,9994) (KK,KK=1,8)                                   FEM20320
         I=0                                                           FEM20330
      ENDIF                                                            FEM20340
      WRITE(108,9997) L,INPUT                                          FEM20350
      IF (INDEX(INPUT,'STOP'  ).EQ.0) GO TO 9995                       FEM20360
9994  FORMAT ('1     ECHO OF INPUT DATA '//                            FEM20370
     &  1X,3X,'LINE  ',8('....:... ',I1,'0'))                         FEM20380
9997  FORMAT (1X,I7,'. ',A)                                            FEM20390
9996  REWIND(105)                                                      FEM20400
C                                                                      FEM20410
C*------------------------                                            FEM20420
C** INITIALIZE DATA.....   **                                         FEM20430
C*------------------------                                            FEM20440
      TITLE(1)=' '                                                     FEM20450
      TITLE(2)=' '                                                     FEM20460
      BLANK  =' '                                                      FEM20470
      BUG =.FALSE.                                                     FEM20480
      DOUBLE =.FALSE.                                                  FEM20490
      IBUG=0                                                           FEM20500
      MAXEDF=24                                                        FEM20510
      MD(1)=1                                                          FEM20520
      DO 1 I=2,(MAXEDF+1)                                              FEM20530
    1   MD(I)=MD(I-1)+(I-1)                                            FEM20540
        IZ=0                                                           FEM20550
C                                                                      FEM20560
   20 READ(105,*,END=1000) OPTION                                      FEM20570
      RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                   FEM20580
C                                                                      FEM20590
C*--------------------------                                          FEM20600
C** DEFINE THE STRUCTURE ---                                          FEM20610
C*--------------------------                                          FEM20620
      IF (INDEX(OPTION,'STR').NE.0)  THEN                              FEM20630
C ** ------------------------                                         FEM20640
            IZ=1                                                       FEM20650
            TITLE(1)=' '                                               FEM20660
            TITLE(2)=' '                                               FEM20670
C ---------------------------                                         FEM20680
            GRAV=386.4                                                 FEM20690
            READ(105,*,END=1000) TITLE(1)                              FEM20700
            CALL TIDAY(TIME1,DATE1)                                    FEM20710
            WRITE(TITLE(1)(80:),45) TIME1,DATE1                        FEM20720
            RATIZ=100.*REAL(IZ)/REAL(MAXZ)                             FEM20730
            WRITE(*,52)' BUILDING STRUCTURAL MODEL ------------------*', FEM20740
     &                 IZ,RATIZ                                        FEM20750
            WRITE(*,51) TITLE(1)(1:70)                                 FEM20760
            CALL STRUCT                                                FEM20770
            IZREL=IZ                                                   FEM20780
            RATIZ=100.*REAL(IZ)/REAL(MAXZ)                             FEM20790
C                                                                      FEM20800
C*--------------------------------                                    FEM20810
C** DEFINE THE LOADING AND SOLVE --                                    FEM20820
C*--------------------------------                                    FEM20830
      ELSE IF (INDEX(OPTION,'SOL') NE.0)  THEN                         FEM20840
            READ(105,*,END=1000) TITLE(2)                              FEM20850
            CALL TIDAY(TIME1,DATE1)                                    FEM20860
            WRITE(TITLE(2)(80:),45) TIME1,DATE1                        FEM20870
            WRITE(*,52)' SOLVING STRUCTURAL MODEL ------------------*', FEM20880
     &                 IZ,RATIZ                                        FEM20890
            WRITE(*,51) TITLE(1)(1:70)                                 FEM20900
            WRITE(*,51) TITLE(2)(1:70)                                 FEM20910
            CALL SOLN(OPTION)                                          FEM20920
            RATIZ=100.*REAL(IZ)/REAL(MAXZ)                             FEM20930
C                                                                      FEM20940
C*--------------------------------                                    FEM20950
C** STOP.... END OF PROGRAM.       **                                 FEM20960
C*--------------------------------                                    FEM20970
      ELSE IF (INDEX(OPTION,'STOP').NE.0)  THEN                        FEM20980
            WRITE(*,52)' NORMAL STOP ----- -------------------------*', FEM20990
     &                 IZ,RATIZ                                        FEM21000
            STOP                                                       FEM21010
C                                                                      FEM21020
C*--------------------------------------                              FEM21030
C**  READ RESULTS FROM INPUT FILE ***                                  FEM21040
C*--------------------------------------                              FEM21050
      ELSE IF (INDEX(OPTION,'READ').NE.0)  THEN                        FEM21060
  140       CALL GETINT(OPTION,'UNIT=',IUNIT,0,.TRUE.,.TRUE. )         FEM21070
            CALL GETINT(OPTION,'INC=',INC  ,1,.TRUE.,.FALSE.)          FEM21080
            IF (IUNIT.GT.0) THEN                                       FEM21090
              WRITE(*,52) ' READING DATA FROM FILE ---------- --------*',FEM21100
     &                    IZ,RATIZ                                     FEM21110
              IOPT=4                                                   FEM21120
              CALL DMPDAT(IOPT,IWRITE,TO,DT)                           FEM21130
              GO TO 140                                                FEM21140
            ENDIF                                                      FEM21150
            RATIZ=100.*REAL(IZ)/REAL(MAXZ)                             FEM21160
C                                                                      FEM21170
C*------------------------                                            FEM21180
C** SET BUG OPTION     **                                            FEM21190
C*------------------------                                            FEM21200
      ELSE IF (TEST(OPTION,'BUG=', FALSE.))  THEN                      FEM21210
            CALL GETCHR(OPTION,'BUG=',INPUT,.TRUE.,.FALSE.)            FEM21220
            IBUG=0                                                     FEM21230
            WRITE(OPTION,160) INPUT(1:70)                              FEM21240
            DO 150 I=0,23                                              FEM21250
              IF (TEST(INPUT,CHR(I),.FALSE.)) IBUG=IBSET(IBUG,I)       FEM21260
  150       CONTINUE                                                   FEM21270
  160       FORMAT ('BUG=',A)                                          FEM21280
            WRITE(*,52)' SET BUG OPTION: ---- ------------ ----------*', FEM21290
     &                 IZ,RATIZ                                        FEM21300
            WRITE(*,51)  OPTION(1:70)                                  FEM21310
            WRITE(108,51)  'SET BUG OPTION= '                          FEM21320
            WRITE(108,51)  OPTION(1:70)                                FEM21330
C                                                                      FEM21340
C*------------------------                                            FEM21350
C** SET BUG OPTION     **                                            FEM21360
C*------------------------                                            FEM21370
      ELSE IF (INDEX(OPTION,'BUG') NE.0)  THEN                         FEM21380
            WRITE(*,51) ('SET BUG OPTION. '//OPTION(1:54) )            FEM21390
            BUG= TRUE.                                                 FEM21400
            IF (INDEX(OPTION,'NOBUG').NE.0) BUG=.FALSE                 FEM21410
C-------------------------------------                               FEM21420
C** INVALID OPTION                 **                                FEM21430
C*-------------------------------------                              FEM21440
```

```
      ELSE                                                              FEM21450
           IF (TEST(OPTION,'NOECHO',.FALSE.) ) GO TO 20                 FEM21460
           IF (OPTION.EQ.BLANK) GO TO 20                                FEM21470
           WRITE(108,30) OPTION                                         FEM21480
           WRITE(*,51) ('INVALID OPTION. '//OPTION(1:54))               FEM21490
      ENDIF                                                             FEM21500
      GO TO  20                                                         FEM21510
C                                                                       FEM21520
   45 FORMAT(' TIME: ',A,', DATE: ',A)                                  FEM21530
   50 FORMAT (A1,' STRUCTURE....: ',A,/1X,' SOLUTION ....: ',A,/)       FEM21540
   51 FORMAT (1X,A)                                                     FEM21550
   52 FORMAT (1X,'*---',A45,'*',:,' I2=',I7,' MEM=',F6 3,'B')           FEM21560
   30 FORMAT (' 1 INVALID OPTION= ',A)                                  FEM21570
  110 FORMAT (' ZMATRX(',I6,'):',11S,1P,G20.10,:,G20.10)                FEM21580
 1000 STOP                                                              FEM21590
      END                                                               FEM21600
C                                                                       TIDA0010
C ----- TAKE PRESENT TIME AND DATE ----                                 TIDA0020
C                                                                       TIDA0030
      SUBROUTINE TIDAE(TIME1,DATE1)                                     TIDA0040
       INTEGER*2 IYR,IMON,IDAY,IHR,IMIN,ISEC,I100TH                     TIDA0050
       CHARACTER*8 TIME1,DATE1                                          TIDA0060
       CALL GETDAT(IYR,IMON,IDAY)                                       TIDA0070
       IYR=IYR-1900                                                     TIDA0080
       CALL GETTIM(IHR,IMIN,ISEC,I100TH)                                TIDA0090
       WRITE(DATE1,'(i2,1h/,i2,1h/,i2)') IMON,IDAY,IYR                  TIDA0100
       WRITE(TIME1,'(i2,1h:,i2,1h:,i2)') IHR,IMIN,ISEC                  TIDA0110
      RETURN                                                            TIDA0120
      END                                                               TIDA0130
C                                                                       SOLN0010
C-----------------------------------------------------------------------SOLN0020
C                                                                       SOLN0030
      SUBROUTINE SOLN(OPTION)                                           SOLN0040
      CHARACTER*80 OPTION                                               SOLN0050
$INCLUDE:'ZCOMN'                                                        SOLN0060
      STEPID=' '                                                        SOLN0070
C                                                                       SOLN0080
C-------------------------                                              SOLN0090
C- DETERMINE SOLUTION NO. **                                            SOLN0100
C-------------------------                                              SOLN0110
      CALL GETINT(OPTION,'SOL',NSOLN,0,.TRUE.,.FALSE.)                  SOLN0120
C                                                                       SOLN0130
C-------------------------                                              SOLN0140
C- CHOOSE THE SOLUTION --                                               SOLN0150
C-------------------------                                              SOLN0160
C     IF (NSOLN EQ.1) THEN                                              SOLN0170
C          CALL SOL01                                                   SOLN0180
C     ELSE IF (NSOLN.EQ 2) THEN                                         SOLN0190
      IF (NSOLN.EQ.2) THEN                                              SOLN0200
           NLOAD=1                                                      SOLN0210
           MAXELD=0                                                     SOLN0220
           CALL SOL02                                                   SOLN0230
C     ELSE IF (NSOLN.EQ.3) THEN                                         SOLN0240
C          NLOAD=1                                                      SOLN0250
C          MAXELD=0                                                     SOLN0260
C          CALL SOL03                                                   SOLN0270
C     ELSE IF (NSOLN.EQ.4) THEN                                         SOLN0280
C          NLOAD=1                                                      SOLN0290
C          CALL SOL04                                                   SOLN0300
C     ELSE IF (NSOLN.EQ.5) THEN                                         SOLN0310
C          NLOAD=1                                                      SOLN0320
C          MAXELD=0                                                     SOLN0330
C          CALL SOL05                                                   SOLN0340
      ELSE                                                              SOLN0350
           WRITE(108,*) 'INVALID SOLN#',NSOLN                           SOLN0360
           STOP 'INVALID SOLUTION NUMBER'                               SOLN0370
      ENDIF                                                             SOLN0380
      RETURN                                                            SOLN0390
      END                                                               SOLN0400
C                                                                       MSOL0010
C-----------------------------------------------------------------------MSOL0020
C  MATRIX SOLUTION OF EQUATION AX=P, WHERE THE UPPER TRIANGULAR MATRIX  MSOL0030
C  OF THE SYMMETRIC MATRIX A IS STORED BY SKYLINES. P IS STORED AS A    MSOL0040
C  FULL MATRIX.  THE MATRIX P IS REPLACED BY X AFTER BACK SUBSTITUTION  MSOL0050
C  IS COMPLEATE.  BOTH A AND P ARE ALTERED. ...                         MSOL0060
C  GUYAN OPTION: IF GUYAN IS .TRUE. MASS MATRIX IS ALSO CONDENSED       MSOL0070
C-----------------------------------------------------------------------MSOL0080
      SUBROUTINE MSOLL(IOPT,PRINT,N,IMD,LROW,NLC,L1,L2,MD,A,P,FACTOR,   MSOL0090
     &                 GUYAN,MASS,MDMASS,IMDMAS,KGCOND,KG,MDKG,IMDKG)  MSOL0100
      REAL   MASS(IMDMAS),KG(IMDKG)                                     MSOL0110
      LOGICAL GUYAN,PRINT,KGCOND                                        MSOL0120
      DIMENSION A(IMD),FACTOR(N),P(LROW,NLC),MD(N+1)                    MSOL0130
      DIMENSION MDMASS(N+1),MDKG(N+1)                                   MSOL0140
C                                                                       MSOL0150
      IJI(I,J)=MD(J)+J-I                                                MSOL0160
      IJM(I,J)=MDMASS(J)+J-I                                            MSOL0170
      IJKG(I,J)=MDKG(J)+J-I                                             MSOL0180
C-----------------------------------------------------------------------MSOL0190
C   MATRIX STORAGE SCHEME:                                              MSOL0200
C   THE MATRIX A IS BANDED AND SYMMETRIC. THUS ONLY THE SKYLINE ABOVE   MSOL0210
C   AND THE MAIN DIAGONAL NEEDS TO BE STORED                            MSOL0220
C   THE MATRIX A IS STORED IN A LINEAR ARRAY BY COLUMNS.  THE VALUE ON  MSOL0230
C   THE MAIN DIAGONAL IS STORED IN THE LINEAR ARRAY FIRST.              MSOL0240
C   ADDRESSES OF THE LINEAR ARRAY ELEMENTS CONTAINING THE MAIN DIAGONAL MSOL0250
C   TERMS ARE STORED IN MD.                                             MSOL0260
C   EXAMPLE: LET B BE THE 5X5 FULL SYMETRIC MATRIX BELOW.               MSOL0270
C        |  1  |  3  |     |  12 |                                      MSOL0280
C        |     |  2  |  5  |     |  11 |  NOTE: B(1,3),B(1,4) AND B(2,4) ARE ZERO. MSOL0290
C   B= |     |     |  4  |  7  |  10 |                                  MSOL0300
C        |  SYMM.    |  6  |  9  |  MD= 1, 2, 4, 6, 8, 13              MSOL0310
C        |           |     |  8  |                                      MSOL0320
C   THUS, B(3,3)=A( MD(3) )= A(4)                                       MSOL0330
C         B(I,J)=A( MD(J)+J-I ) IF J>I AND (MD(J)+J-I)<MD(J+1)          MSOL0340
C               =0              IF J>I AND (MD(J)+J-I)>MD(J+1)        MSOL0350
C   MATRIX P IS NOT SYMMETRIC, AND IS STORED AS A FULL MATRIX           MSOL0360
C-----------------------------------------------------------------------MSOL0370
C   VARIABLE TABLE:                                                     MSOL0380
C   IOPT  - OPTION NUMBER                                               MSOL0390
C   N     - NUMBER OF DEGREES OF FREEDOM OF A MATRIX                    MSOL0400
C   NLC   - NUMBER OF LOAD CASES FOR THE P MATRIX                       MSOL0410
C   L1    - FIRST ROW TO BE WORKED WITH                                 MSOL0420
C   L2    - LAST ROW TO BE WORKED WITH                                  MSOL0430
C   MD    - ADDRESS OF THE MAIN DIAGONAL TERMS OF A                     MSOL0440
C   P     - LAOD MATRIX ON INPUT, SOLN (X) OF AX=P ON COMPLEATION OF    MSOL0450
C           IOPT=4                                                      MSOL0460
C   FACTOR- TEMPORARY STORAGE MATRIX                                    MSOL0470
C   GUYAN - GUYAN REDUCTION FLAG FOR MASS MATRIX                        MSOL0480
C   MASS  - MASS MATRIX                                                 MSOL0490
C   MDMASS- MASS MATRIX ADDRESSES OF MAIN DIAGONAL ELEMENTS             MSOL0500
C   KGCOND- FLAG FOR CONDENSING OUT KG                                  MSOL0510
C   KG    - GEOMETRIC STIFFNESS MATRIX                                  MSOL0520
C   MDKG  - KG MATRIX ADDRESSES OF MAIN DIAGONAL ELEMENTS               MSOL0530
C   L     - ROW NUMBER FOR ELIMINATION                                  MSOL0540
C   I     - ROW NUMBER                                                  MSOL0550
C   J     - COLUMN NUMBER                                               MSOL0560
C-----------------------------------------------------------------------MSOL0570
C                                                                       MSOL0580
      L1=MAX(L1,1)                                                      MSOL0590
      L2=MIN(L2,N)                                                      MSOL0600
C                                                                       MSOL0610
      GO TO (10,110,210,310) IOPT                                       MSOL0620
C                                                                       MSOL0630
   10 CONTINUE                                                          MSOL0640
C -------- IOPT=1, REDUCE A AND P ------------------------------------ MSOL0650
C          GAUSSIAN ELIMINIATION IS USED TO REDUCE THE A AND P MATRICIESMSOL0660
C          FROM ROW L1 TO ROW L2   IF L1 IS NOT 1, IT IS ASSUMED THAT   MSOL0670
C          A AND P ARE ALLREADY REDUCED FROM 1 TO L1 . . .              MSOL0680
C-----------------------------------------------------------------------MSOL0690
      IF (PRINT) THEN                                                   MSOL0700
        WRITE(108,500) 'IOPT=1, REDUCE STIFFNESS AND LOADS',L1,L2,N     MSOL0710
        CALL LMATRX(A,MD,N,IMD ,'INPUT STIFFNESS')                      MSOL0720
        CALL WMATRX(P,N ,NLC,'INPUT LOAD')                              MSOL0730
        IF (GUYAN ) CALL LMATRX(MASS,MDMASS,N,IMDMAS,'INPUT MASS')      MSOL0740
        IF (KGCOND) CALL LMATRX(KG  ,MDKG  ,N,IMDKG ,'INPUT KG  ')     MSOL0750
      ENDIF                                                             MSOL0760
C                                                                       MSOL0770
      DO 100 L=L1,L2                                                    MSOL0780
        IF (L.EQ.N) GO TO 100                                           MSOL0790
C                                                                       MSOL0800
C------- TEST FOR ZERO PIVOT..... .... .. ...                          MSOL0810
        IF (A(MD(L)).EQ.0) THEN                                         MSOL0820
          WRITE(108,2000) L,MD(L),A(MD(L))                              MSOL0830
          STOP                                                          MSOL0840
        ENDIF                                                           MSOL0850
C                                                                       MSOL0860
C------- CALCULATE THE FACTORS EACH ROW IS MULTIPLIED BY                MSOL0870
        DO 30 I=(L+1),N                                                 MSOL0880
          LI=IJI(L,I)                                                   MSOL0890
          IF ( LI.LT.MD(I+1) ) THEN                                     MSOL0900
           FACTOR(I)=A(LI) /A( MD(L) )                                  MSOL0910
          ELSE                                                          MSOL0920
           FACTOR(I)=0.                                                 MSOL0930
          ENDIF                                                         MSOL0940
   30     CONTINUE                                                      MSOL0950
C                                                                       MSOL0960
C------- REDUCE A BY COLUMNS (J) EACH ROW (I)..                         MSOL0970
        DO 40 J=(L+1),N                                                 MSOL0980
          LJ=IJI(L,J)                                                   MSOL0990
          IF (LJ.GE.MD(J+1)) GO TO 40                                   MSOL1000
          DO 35 I=(L+1),J                                               MSOL1010
            IJ=IJI(I,J)                                                 MSOL1020
            A(IJ)=A(IJ)-FACTOR(I)*A(LJ)                                 MSOL1030
   35       CONTINUE                                                    MSOL1040
   40   CONTINUE                                                        MSOL1050
C                                                                       MSOL1060
C- ----- REDUCE P BY COLUMNS (J) EACH ROW (I) .                         MSOL1070
        DO 50 J=1,NLC                                                   MSOL1080
          IF (P(L,J).EQ.0) GO TO 50                                     MSOL1090
          DO 45 I=(L+1),N                                               MSOL1100
   45       P(I,J)=P(I,J)-FACTOR(I)*P(L,J)                             MSOL1110
   50   CONTINUE                                                        MSOL1120
C                                                                       MSOL1130
C------- REDUCE MASS MATRIX BY GUYAN REDUCTION                          MSOL1140
      IF (GUYAN) THEN                                                   MSOL1150
C                                                                       MSOL1160
C......... M22-M22 - K21/K11 * M11 * 1/K11 * K12                       MSOL1170
        RM=MASS(MDMASS(L))                                              MSOL1180
        IF (RM.NE.0) THEN                                               MSOL1190
             DO 60 J=L+1,N                                              MSOL1200
                  IF (FACTOR(J).EQ.0) GO TO 60                          MSOL1210
                  I1=MAX(L+1,J-(MDMASS(J+1)-MDMASS(J)) )                MSOL1220
                  DO 55 I=I1,J                                          MSOL1230
                   IJ=IJM(I,J)                                          MSOL1240
   55             MASS(IJ)=MASS(IJ)+FACTOR(I)*RM*FACTOR(J)             MSOL1250
   60        CONTINUE                                                   MSOL1260
        ENDIF                                                           MSOL1270
C                                                                       MSOL1280
C.. ..... M22-M22 - K21/K11 * M12                                       MSOL1290
        DO 66 J=L+1,N                                                   MSOL1300
             LJ=IJM(L,J)                                                MSOL1310
             IF (LJ.GE.MDMASS(J+1)) GO TO 66                            MSOL1320
             IF (MASS(LJ).EQ.0) GO TO 66                                MSOL1330
             DO 65 I=L+1,J                                              MSOL1340
               IJ=IJM(I,J)                                              MSOL1350
               IF (IJ.LT.MDMASS(J+1))                                   MSOL1360
     &         MASS(IJ)=MASS(IJ)-FACTOR(I)*MASS(LJ)                     MSOL1370
   65        CONTINUE                                                   MSOL1380
   66   CONTINUE                                                        MSOL1390
C                                                                       MSOL1400
C..... .. M22-M22 - M21 * 1/K11 * K12                                   MSOL1410
        DO 70 J=L+1,N                                                   MSOL1420
             IF (FACTOR(J).EQ.0) GO TO 70                               MSOL1430
             DO 69 I=L+1,J                                              MSOL1440
                  LI=IJM(L,I)                                           MSOL1450
                  IF (LI.GE.MDMASS(I+1)) GO TO 69                       MSOL1460
                  IJ=IJM(I,J)                                           MSOL1470
                  IF (IJ.LT.MDMASS(J+1))                                MSOL1480
     &            MASS(IJ)=MASS(IJ)-FACTOR(J)*MASS(LI)                  MSOL1490
   69        CONTINUE                                                   MSOL1500
   70   CONTINUE                                                        MSOL1510
      ENDIF                                                             MSOL1520
C                                                                       MSOL1530
C------- REDUCE GEOMETRIC STIFFNESS MATRIX                              MSOL1540
      IF (KGCOND) THEN                                                  MSOL1550
C                                                                       MSOL1560
C......... KG22-KG22 - K21/K11 * KG11 * 1/K11 * KG12                   MSOL1570
        RKG=KG(MDKG(L))                                                 MSOL1580
        IF (RKG.NE.0) THEN                                              MSOL1590
             DO 80 J=L+1,N                                              MSOL1600
                  IF (FACTOR(J).EQ.0) GO TO 80                          MSOL1610
                  I1=MAX(L+1,J-(MDKG(J+1)-MDKG(J)) )                    MSOL1620
                  DO 75 I=I1,J                                          MSOL1630
                   IJ=IJKG(I,J)                                         MSOL1640
   75             KG(IJ)=KG(IJ)+FACTOR(I)*RKG*FACTOR(J)                MSOL1650
   80        CONTINUE                                                   MSOL1660
        ENDIF                                                           MSOL1670
C                                                                       MSOL1680
C  ... ... KG22-KG22 - K21/K11 * KG12                                   MSOL1690
        DO 86 J=L+1,N                                                   MSOL1700
             LJ=IJKG(L,J)                                               MSOL1710
             IF (LJ.GE.MDKG(J+1)) GO TO 86                              MSOL1720
             IF (KG(LJ).EQ.0) GO TO 86                                  MSOL1730
             DO 85 I=L+1,J                                              MSOL1740
                  IJ=IJKG(I,J)                                          MSOL1750
                  IF (IJ.LT.MDKG(J+1))                                  MSOL1760
     &            KG(IJ)=KG(IJ)-FACTOR(I)*KG(LJ)                        MSOL1770
   85        CONTINUE                                                   MSOL1780
   86   CONTINUE                                                        MSOL1790
C                                                                       MSOL1800
C......... KG22-KG22 - KG21 * 1/K11 * K12                              MSOL1810
        DO 90 J=L+1,N                                                   MSOL1820
             IF (FACTOR(J).EQ.0) GO TO 90                               MSOL1830
             DO 89 I=L+1,J                                              MSOL1840
                  LI=IJKG(L,I)                                          MSOL1850
                  IF (LI.GE.MDKG(I+1)) GO TO 89                         MSOL1860
                  IJ=IJKG(I,J)                                          MSOL1870
                  IF (IJ.LT.MDKG(J+1))                                  MSOL1880
     &            KG(IJ)=KG(IJ)-FACTOR(J)*KG(LI)                        MSOL1890
   89        CONTINUE                                                   MSOL1900
   90   CONTINUE                                                        MSOL1910
C                                                                       MSOL1920
      ENDIF                                                             MSOL1930
C                                                                       MSOL1940
  100 CONTINUE                                                          MSOL1950
      IF (PRINT) THEN                                                   MSOL1960
        CALL LMATRX(A,MD,N,IMD,'REDUCED STIFFNESS')                     MSOL1970
        CALL WMATRX(P,N ,NLC,'REDUCED LOAD')                            MSOL1980
        IF (GUYAN)  CALL LMATRX(MASS,MDMASS,N,IMDMAS,'REDUCED MASS')    MSOL1990
        IF (KGCOND) CALL LMATRX(KG  ,MDKG  ,N,IMDKG ,'REDUCED KG')      MSOL2000
      ENDIF                                                             MSOL2010
      RETURN                                                            MSOL2020
C                                                                       MSOL2030
  110 CONTINUE                                                          MSOL2040
C--- ------ IOPT=2, REDUCE P ONLY ------------------ ---------- ---- ---MSOL2050
C          GAUSSIAN ELIMINIATION IS USED TO REDUCE THE P MATRIX         MSOL2060
C          FROM ROW L1 TO ROW L2   IF L1 IS NOT 1, IT IS ASSUMED THAT   MSOL2070
C          A AND P ARE ALLREADY REDUCED FROM 1 TO L1 . .                MSOL2080
C----- ------------------------------------------------------------ ----MSOL2090
      IF (PRINT) THEN                                                   MSOL2100
        WRITE(108,500) 'IOPT=2, REDUCE LOADS ONLY',L1,L2,N              MSOL2110
        CALL LMATRX(A,MD,N,MD(N+1),'REDUCED STIFFNESS')                 MSOL2120
        CALL WMATRX(P,N ,NLC,'NEW LOAD MATRIX ')                        MSOL2130
      ENDIF                                                             MSOL2140
C                                                                       MSOL2150
C------- LOOP FOR EACH ROW (L) TO REDUCE.                               MSOL2160
      DO 160 L=L1,L2                                                    MSOL2170
C                                                                       MSOL2180
        IF (A(MD(L)).EQ.0) THEN                                         MSOL2190
          WRITE(108,2000) L,MD(L),A(MD(L))                              MSOL2200
          STOP                                                          MSOL2210
        ENDIF                                                           MSOL2220
C                                                                       MSOL2230
```

```
C       LOOP FOR EACH COLUMN (J)                                        MSOL2240
        DO 150 J=1,NLC                                                  MSOL2250
          IF (P(L,J).EQ.0) GO TO 150                                    MSOL2260
          DO 120 I=(L+1),N                                              MSOL2270
            LI=IJI(L,I)                                                 MSOL2280
            IF (LI.LT.MD(I+1)) P(I,J)=P(I,J)-A(LI)*P(L,J)/A(MD(L))      MSOL2290
  120       CONTINUE                                                    MSOL2300
  150     CONTINUE                                                      MSOL2310
  160     CONTINUE                                                      MSOL2320
        RETURN                                                          MSOL2330
C                                                                       MSOL2340
  210 CONTINUE                                                          MSOL2350
C--------- IOPT=3, BACK SUBSTUTION  ------------------------------------MSOL2360
C           BACK SUBSTUTION IS USED TO SOLVE FOR X IN SX=V, WHERE S IS  MSOL2370
C           THE REDUCE UPPER TRIANGULAR FACTOR OF THE A MATRIX, AND V   MSOL2380
C           IS THE REDUCED FORRM OF THE P MATRIX   THE RESULTS (X) ARE  MSOL2390
C           STORED IN P.                                                MSOL2400
C           BACK SUBSTUTION ONLY TAKES PLACE BETWEEN ROWS L1 AND L2.    MSOL2410
C ---  ------------------------------------------------------------     MSOL2420
        IF (PRINT) THEN                                                 MSOL2430
           WRITE(108,500) 'IOPT=3, BACK SUBSTUTION',L1,L2,N             MSOL2440
           CALL LMATRX(A,MD,N,MD(N+1),'REDUCED STIFFNESS')              MSOL2450
           CALL WMATRX(P,N ,NLC,'REDUCED LOAD')                         MSOL2460
        ENDIF                                                           MSOL2470
C                                                                       MSOL2480
C------ LOOP FOR EACH ROW L                                            MSOL2490
C       CALCULAGE X FOR EACH LOAD CASE, STORE IN P                      MSOL2500
        DO 250 J=L2,L1,-1                                               MSOL2510
        IF (A(MD(J)).EQ.0) THEN                                         MSOL2520
          WRITE(108,2000) J,MD(J),A(MD(J))                              MSOL2530
          STOP                                                          MSOL2540
        ENDIF                                                           MSOL2550
        DO 215 LC=1,NLC                                                 MSOL2560
  215     P(J,LC)=P(J,LC)/A(MD(J))                                      MSOL2570
        IF (J.GT.1) THEN                                                MSOL2580
          IJ=MD(J-1)                                                    MSOL2590
          NI=J-(MD(J+1)-MD(J)) +1                                       MSOL2600
          DO 220 I=NI,J-1,+1                                            MSOL2610
            IJ=IJ+1                                                     MSOL2620
            DO 220 LC=1,NLC                                             MSOL2630
              P(I,LC)=P(I,LC)-P(J,LC)*A(IJ)                             MSOL2640
  220   CONTINUE                                                        MSOL2650
        ENDIF                                                           MSOL2660
  250   CONTINUE                                                        MSOL2670
        IF (PRINT) CALL WMATRX(P,N ,NLC,'DISPLACEMENT')                 MSOL2680
C                                                                       MSOL2690
        RETURN                                                          MSOL2700
C                                                                       MSOL2710
  310 CONTINUE                                                          MSOL2720
C--------- IOPT=4, REDUCED MULTIPLICATION ------------------------------MSOL2730
C           SOLVE FOR V IN S*D-P WHERE S HAS BEEN REDUCED BY GAUSSIAN   MSOL2740
C           ELIMINATION.                                                MSOL2750
C           THE RESULTING LOAD IS STORED IN P.                          MSOL2760
C           THE DISPLACEMENTS ARE INPUT IN D                            MSOL2770
C           MULTIPLACATION ONLY TAKES PLACE BETWEEN ROWS L1 AND L2.     MSOL2780
C------- ------------------------------------------------------------   MSOL2790
        IF (PRINT) THEN                                                 MSOL2800
           WRITE(108,500) 'IOPT=4, REDUCED MULTIPLACATION',L1,L2,N      MSOL2810
           CALL LMATRX(A,MD,N,MD(N+1),'REDUCED STIFFNESS')              MSOL2820
           CALL WMATRX(FACTOR,N ,1,'DISPLACEMENT')                      MSOL2830
        ENDIF                                                           MSOL2840
C                                                                       MSOL2850
C------ ONLY ONE LOAD CASE. . .                                        MSOL2860
        LC=1                                                            MSOL2870
C       ---  ZERO P ----                                               MSOL2880
        DO 320 J=L1,L2                                                  MSOL2890
          IF (A(MD(J)).EQ.0) THEN                                       MSOL2900
            WRITE(108,2000) J,MD(J),A(MD(J))                            MSOL2910
            STOP                                                        MSOL2920
          ENDIF                                                         MSOL2930
  320     P(J,LC)=0                                                     MSOL2940
C                                                                       MSOL2950
C---------- MULTIPLY S_RED * DISP = P                                   MSOL2960
        DO 340 J=L1,N                                                   MSOL2970
          IC1=J-(MD(J+1)-MD(J)) +1                                      MSOL2980
          IC2=MAX(IC1,L1)                                               MSOL2990
          DO 340 I=J,IC2, 1                                             MSOL3000
            IJ=IJI(I,J)                                                 MSOL3010
  340       P(I,LC)=P(I,LC)+A(IJ)*FACTOR(J)                             MSOL3020
        IF (PRINT) CALL WMATRX(P,N ,1,'REDUCED LOAD')                   MSOL3030
C                                                                       MSOL3040
C------- UNREDUCE THE LOAD MATRIX ...                                   MSOL3050
        DO 360 L=L2,L1, 1                                               MSOL3060
C                                                                       MSOL3070
C       LOOP FOR EACH COLUMN (J)                                        MSOL3080
        DO 350 J=1,NLC                                                  MSOL3090
          IF (P(L,LC).EQ.0) GO TO 350                                   MSOL3100
          DO 345 I=(L+1),N                                              MSOL3110
            LI=IJI(L,I)                                                 MSOL3120
            IF (LI.LT.MD(I+1)) P(I,LC)=P(I,LC)+A(LI)*P(L,LC)/A(MD(L))   MSOL3130
  345       CONTINUE                                                    MSOL3140
  350     CONTINUE                                                      MSOL3150
  360     CONTINUE                                                      MSOL3160
  600   CONTINUE                                                        MSOL3170
C                                                                       MSOL3180
        IF (PRINT) CALL WMATRX(P,N ,1,'LOAD')                           MSOL3190
  500 FORMAT (/2X,A,2X,'L1:',I5,'   L2.',I5,'   N:',I5)                MSOL3200
 2000 FORMAT (///1X,50('*')/                                            MSOL3210
     &        ' *  ER','ROR IN SUBROUTINE MSOLL',       T51,'*'/       MSOL3220
     &        ' *  PIVOT IS LE 0 ------- ROW ',I5,     T51,'*'/       MSOL3230
     &        ' *                  STORAGE LCN.',I5,   T51,'*'/       MSOL3240
     &        ' *                        PIVOT=',1P,G15.6,T51,'*'/     MSOL3250
     &        ' *  SOLUTION IS ABORTED ',              T51,'*'/       MSOL3260
     &        1X,50('*')/)                                            MSOL3270
C                                                                       MSOL3280
        RETURN                                                          MSOL3290
        END                                                             MSOL3300
C                                                                       REAC0010
C*********************************************************************  REAC0020
C'                                                                      REAC0030
      SUBROUTINE REACTN(IOPT,PRINT,NNODE,L1,L2,L3,L4,FORCE,DISPL,       REAC0040
     &                  MD,A,KM,NDOF,NLOAD,FACTOR,                     REAC0050
     &                  MAXNOD,IDOF,COORD,ID,TITLE,STEPID,HEAD,         REAC0060
     &                  NCOS,COSINE,JTCOS,JTFLG,SUM)                    REAC0070
      CHARACTER*(*) TITLE(2),STEPID                                     REAC0080
      CHARACTER*15 CD(6)                                                REAC0090
      LOGICAL PRINT,HEAD,FLAG,BTEST                                     REAC0100
      DIMENSION A(KM),MD(NDOF+1),IDOF(MAXNOD,6),G(6)                    REAC0110
      DIMENSION FORCE(NDOF,NLOAD),DISPL(NDOF,NLOAD)                     REAC0120
      DIMENSION ID(MAXNOD),COORD(MAXNOD,3),JTFLG(MAXNOD)                REAC0130
      DIMENSION F(6),SUM(6),COSINE(3,3,NCOS),JTCOS(MAXNOD)              REAC0140
C                                                                       REAC0150
      IF (IOPT.EQ.1) THEN                                               REAC0160
C                                                                       REAC0170
C---------------------------------------                                REAC0180
C- MODIFY LOADS FOR RESTRAINT DISPLACEMENTS ---                         REAC0190
C---------------------------------------                                REAC0200
C CONSIDER THE PARTITIGNED STIFFNESS WHERE X2 ARE THE RESTRAINT DISPL.  REAC0210
C                                                                       REAC0220
C   |P1|   | K11 | K12 | |X1|                                         REAC0230
C   |--| = |-----|-----| |--|                                         REAC0240
C   |P2|   | K21 | K22 | |X2|                                         REAC0250
C                                                                       REAC0260
C   THE MODIFIED FREE LOAD IS (P1-K12*X2) - K11*X1                      REAC0270
C  ------------------------------------------------------------- C      REAC0280
C       DO 10 I=L1,L2                                          C  FULL STORAGE  REAC0290
C       DO 10 J=1,NLOAD                                        C  EQUIVALENT    REAC0300
C       DO 10 K=L3,L4                                          C                REAC0310
C 10      FORCE(I,J)=FORCE(I,J)-K(I,K)*DISPL(K,J)              C                REAC0320
C------ ------------------------------------------------------- C      REAC0330
        DO 20 K=L3,L4                                                   REAC0340
        MDK=MD(K+1)                                                     REAC0350
          DO 20 J=1,NLOAD                                               REAC0360
          DO 10 I=L2,L1,-1                                              REAC0370
            IK=MD(K)+K-I                                                REAC0380
            IF (IK.GE.MDK) GO TO 20                                     REAC0390
            FIJ=FORCE(I,J)                                              REAC0400
            AIK=A(IK)                                                   REAC0410
            DKJ=DISPL(K,J)                                              REAC0420
   10       FORCE(I,J)=FORCE(I,J)-A(IK)*DISPL(K,J)                      REAC0430
   20   CONTINUE                                                        REAC0440
        IF (PRINT)                                                      REAC0450
     &  CALL WMATRX(FORCE,NDOF,NLOAD,'MODIFIED LOADS')                  REAC0460
C                                                                       REAC0470
      ELSE IF (IOPT.EQ 2) THEN                                          REAC0480
C                                                                       REAC0490
C-------------------------                                              REAC0500
C- SOLVE FOR REACTIONS ---                                             REAC0510
C-------------------------                                              REAC0520
C CONSIDER THE PARTITIGNED STIFFNESS WHERE X2 ARE THE RESTRAINT DISPL.  REAC0530
C                                                                       REAC0540
C   |P1|   | K11 | K12 | |X1|                                         REAC0550
C   |--| = |-----|-----| |--|   * FACTOR                                REAC0560
C   |P2|   | K21 | K22 | |X2|                                         REAC0570
C                                                                       REAC0580
C   THE REACTIONS ARE  P2 = P2 + ( K21*X1 + K22*X2 ) * FACTOR           REAC0590
C   NOTE: BOTH P1 AND P2 CONTAIN FIXED END FORCES FROM MEMBER LOADS ..  REAC0600
C         THUS, K21*X1 + K22*X2 IS ADDED TO P2 TO GET THE REACTIONS.    REAC0610
C-------------------------------------------------------------C        REAC0620
C     DO 120 I=L3,L4                                          C        REAC0630
C     DO 120 J=1,NLOAD                                        C  FULL STORAGE  REAC0640
CC     FORCE(I,J)=0    * P2 CONTAINS FEF'''                   C  MODE EQUIVALENT REAC0650
C     DO 110 K=L1,L2                                          C        REAC0660
C 110  FORCE(I,J)=FORCE(I,J)+K(I,K)*DISPL(K,J)                C        REAC0670
C     DO 120 K=L3,L4                                          C        REAC0680
C 120  FORCE(I,J)=FORCE(I,J)+K(I,K)*DISPL(K,J)                C        REAC0690
C-------------------------------------------------------------C        REAC0700
C                                                                       REAC0710
        DO 120 I=L3,L4                                                  REAC0720
        DO 120 J=1,NLOAD                                                REAC0730
        DO 110 K=L2,L1,-1                                               REAC0740
          KI=MD(I)+I-K                                                  REAC0750
          IF (KI.GE.MD(I+1)) GO TO 120                                  REAC0760
          AKI=A(KI)                                                     REAC0770
  110   FORCE(I,J)=FORCE(I,J)+A(KI)*DISPL(K,J)*FACTOR                   REAC0780
  120 CONTINUE                                                          REAC0790
C                                                                       REAC0800
        DO 140 I=L3,L4                                                  REAC0810
        DO 140 K=L3,L4                                                  REAC0820
          IF (I.LE.K) THEN                                              REAC0830
            IK=MD(K)+K-I                                                REAC0840
            MDK=MD(K+1)                                                 REAC0850
          ELSE                                                          REAC0860
            IK=MD(I)+I-K                                                REAC0870
            MDK=MD(I+1)                                                 REAC0880
          ENDIF                                                         REAC0890
          IF (IK.GE.MDK)  GO TO 140                                     REAC0900
          AIK=A(IK)                                                     REAC0910
          DO 130 J=1,NLOAD                                              REAC0920
  130       FORCE(I,J)=FORCE(I,J)+A(IK)*DISPL(K,J)*FACTOR               REAC0930
  140   CONTINUE                                                        REAC0940
C                                                                       REAC0950
        IF (PRINT)                                                      REAC0960
     &  CALL WMATRX(FORCE,NDOF,NLOAD,'FINAL LOADS AND REACTIONS')       REAC0970
C                                                                       REAC0980
      ELSE IF (IOPT.EQ.3 .OR. IOPT.EQ.4) THEN                           REAC0990
        NF=L2                                                           REAC1000
C----------------------------                                           REAC1010
C- LOOP FOR EACH LOAD CASE  ---                                        REAC1020
C----------------------------                                           REAC1030
        DO 300 L=1,NLOAD                                                REAC1040
C                                                                       REAC1050
C-------------------------------                                        REAC1060
C- PRINT GLOBAL REACTIONS     ---                                      REAC1070
C-------------------------------                                        REAC1080
C                                                                       REAC1090
        IF (PRINT) THEN                                                 REAC1100
             IF (HEAD) WRITE(108,320) TITLE(1),TITLE(2)                 REAC1110
              NREACT=0                                                  REAC1120
              IF (IOPT.NE.4) WRITE(108,321) L,STEPID                    REAC1130
              IF (IOPT.EQ.4) WRITE(108,421)                             REAC1140
        ENDIF                                                           REAC1150
          IF (IOPT EQ.3) THEN                                           REAC1160
            DO 190 I=1,6                                                REAC1170
  190       SUM(I)=0                                                    REAC1180
          ENDIF                                                         REAC1190
C                                                                       REAC1200
        IF ( .NOT.(NCOS EQ.1  .AND. COSINE(1,1,1) EQ.1 .AND.COSINE(2,2,1)  REAC1210
     &      .EQ 1  AND  COSINE(3,3,1).EQ.1).AND. IOPT EQ.4  ) GO TO 279  REAC1220
C                                                                       REAC1230
          DO 200 I=1,NNODE                                              REAC1240
C                                                                       REAC1250
C  . . . . . GET REACTNS IN JOINT COORDINATE SYSTEM                     REAC1260
            FLAG=.FALSE.                                                REAC1270
            DO 209 J=1,6                                                REAC1280
              IJ=IDOF(I,J)                                              REAC1290
              IF (IJ LE.NF .OR. BTEST(JTFLG(I),J+11)) THEN              REAC1300
                F(J)=0                                                  REAC1310
              ELSE                                                      REAC1320
                F(J)=FORCE(IJ,L)                                        REAC1330
                FLAG=.TRUE.                                             REAC1340
              ENDIF                                                     REAC1350
  209       CONTINUE                                                    REAC1360
            IF (.NOT.FLAG) GO TO 200                                    REAC1370
C                                                                       REAC1380
C....  ... . . TRANSFER REACTNS TO GLOBAL COORDINATE SYSTEM             REAC1390
            DO 229 I1=1,3                                               REAC1400
              I2=I1+3                                                   REAC1410
              G(I1)=0                                                   REAC1420
              G(I2)=0                                                   REAC1430
              DO 219 J1=1,3                                             REAC1440
                J2=J1+3                                                 REAC1450
                G(I1)=G(I1) + COSINE(J1,I1,JTCOS(I))*F(J1)              REAC1460
  219           G(I2)=G(I2) + COSINE(J1,I1,JTCOS(I))*F(J2)              REAC1470
              IF (PRINT) THEN                                           REAC1480
                  WRITE(CD(I1),210) G(I1)                               REAC1490
                  WRITE(CD(I2),210) G(I2)                               REAC1500
              ENDIF                                                     REAC1510
  229         CONTINUE                                                  REAC1520
C                                                                       REAC1530
C. . . . . . SUM REACTIONS .  . .....                                   REAC1540
          IF (IOPT EQ.3) THEN                                           REAC1550
            DO 240 I1=1,6                                               REAC1560
  240         SUM(I1)=SUM(I1) + G(I1)                                   REAC1570
            SUM(4)= SUM(4)   G(2)*COORD(I,3) - G(3)*COORD(I,2)          REAC1580
            SUM(5)= SUM(5) + G(1)*COORD(I,3) - G(3)*COORD(I,1)          REAC1590
            SUM(6)= SUM(6) - G(1)*COORD(I,2) + G(2)*COORD(I,1)          REAC1600
          ENDIF                                                         REAC1610
C                                                                       REAC1620
C. . . . . . PRINT GLOBAL REACTIONS   . . . .                          REAC1630
          NREACT=NREACT+1                                               REAC1640
          IF (PRINT) WRITE(108,220) ID(I),(CD(K),K=1,6)                 REAC1650
  200   CONTINUE                                                        REAC1660
        IF (IOPT.EQ.3  AND. PRINT ) WRITE(108,230) SUM                  REAC1670
  279   IF (IOPT EQ.4 .AND. PRINT ) WRITE(108,231) SUM                  REAC1680
C                                                                       REAC1690
C-----------------------------------------------                        REAC1700
C- PRINT LOCAL JOINT COORDINATE SYSTEMS REACTIONS  ---                  REAC1710
C-----------------------------------------------                        REAC1720
C                                                                       REAC1730
        IF (.NOT. PRINT ) GO TO 300                                     REAC1740
        IF (NCOS.EQ.1  AND. COSINE(1,1,1).EQ 1  AND. COSINE(2,2,1).EQ 1 REAC1750
     &           .AND. COSINE(3,3,1) EQ.1 ) GO TO 300                   REAC1760
C                                                                       REAC1770
        IF (HEAD.AND. NREACT.GT.20) WRITE(108,320) TITLE(1),TITLE(2)    REAC1780
          IF (IOPT.NE 4) WRITE(108,322) L,STEPID                        REAC1790
          IF (IOPT.EQ 4) WRITE(108,422)                                 REAC1800
          DO 290 I=1,NNODE                                              REAC1810
C                                                                       REAC1820
C. . . . . . . . GET REACTNS IN JOINT COORDINATE SYSTEM                 REAC1830
            FLAG=.FALSE                                                 REAC1840
            DO 280 J=1,6                                                REAC1850
```

```
          IJ=IDOF(I,J)                                                  REAC1860
          IF (IJ.LE.NF  OR.  BTEST(JTFLG(I),J+11)) THEN                 REAC1870
            CD(J)=' '                                                   REAC1880
          ELSE                                                          REAC1890
            WRITE(CD(J),210) FORCE(IJ,L)                                REAC1900
            FLAG=.TRUE.                                                 REAC1910
          ENDIF                                                         REAC1920
  280   CONTINUE                                                        REAC1930
        IF (.NOT.FLAG) GO TO 290                                        REAC1940
C                                                                       REAC1950
C.    .. ....  PRINT REACTNS IN LOCAL COORDINATE SYSTEM  ... .          REAC1960
      WRITE(108,221) ID(I),(CD(K),K=1,6),JTCOS(I)                       REAC1970
  290   CONTINUE                                                        REAC1980
  300 CONTINUE                                                          REAC1990
C                                                                       REAC2000
  210 FORMAT (1F,G15.7)                                                 REAC2010
  220 FORMAT (5X,I5,6A)                                                 REAC2020
  221 FORMAT (5X,I5,6A,I10)                                             REAC2030
  230 FORMAT ('                ',90('-'),/' SUMMATION',6G15.7)          REAC2040
  231 FORMAT (' MAX OF ALL',89('-'),/' GCS SUMM.',6G15.7 )              REAC2050
  320 FORMAT ('1 STRUCTURE.....  ',A,/'  SOLUTION.....: ',A)            REAC2060
  321 FORMAT (///'  GCS RESTRAINT REACTIONS,   LOADING #',I5,5X,A,/     REAC2070
     &          ' ------------------------'//6X,'NODE',7X,'FX',        REAC2080
     &           13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ'/)       REAC2090
  322 FORMAT (///'  JCS RESTRAINT REACTIONS,   LOADING #',I5,5X,A,/     REAC2100
     &          ' ------------------------'//6X,'NODE',7X,'FX',        REAC2110
     &          13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ',         REAC2120
     &          6X,'  COSINE #'/)                                     REAC2130
  421 FORMAT (///' MAXIMUM GCS RESTRAINT REACTIONS:',                   REAC2140
     &        5X,'NOTE  MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'/  REAC2150
     &        ' ---------------------------------'//6X,'NODE',7X,'FX',  REAC2160
     &         13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ'/)         REAC2170
  422 FORMAT (///'  MAXIMUM JCS RESTRAINT REACTIONS ',                  REAC2180
     &        5X,'NOTE: MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'/  REAC2190
     &        ' ---------------------------------'//6X,'NODE',7X,'FX',  REAC2200
     &        13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ',           REAC2210
     &        6X,'  COSINE #'/)                                       REAC2220
C                                                                       REAC2230
      ELSE                                                            REAC2240
       WRITE(108,*) 'INVALID OPTION IN SUBROUTINE  REA_CTN, IOPT=',IOPT REAC2250
      ENDIF                                                           REAC2260
      RETURN                                                          REAC2270
      END                                                             REAC2280
C                                                                     PORM0010
C*********************************************************************PORM0020
C                                                                     PORM0030
      SUBROUTINE FORM(IOPT)                                           PORM0040
      CHARACTER  NAME*80,ELNO*130                                     PORM0050
      LOGICAL PRINT,HEAD,FALSE,AXIAL,BTEST,BUG5,BUG57                 PORM0060
$INCLUDE:'ZCOMN'                                                      PORM0070
C                                                                     PORM0080
C---------------------------------------------------------------------PORM0090
C   MATRIX STORAGE SCHEME:                                            PORM0100
C  -THE STIFFNESS MATRIX (K) IS BANDED AND SYMMETRIC.                 PORM0110
C   THUS ONLY THE SKYLINE ABOVE THE MAIN DIAGONAL NEEDS TO BE STORED. PORM0120
C   THE MATRIX K IS STORED IN A LINEAR ARRAY BY COLUMNS               PORM0130
C   THE VALUE ON THE MAIN DIAGONAL IS STORED IN THE LINEAR ARRAY FIRST. PORM0140
C   RELATIVE ADDRESSES OF THE LINEAR ARRAY ELEMENTS CONTAINING THE    PORM0150
C   MAIN DIAGONAL TERMS ARE STORED IN MD                              PORM0160
C   THE ABSOLUTE ADDRESS OF THE FIRST ELEMENT IN THE K MATRIX IS IZSTIP.PORM0170
C   THE STIFFNESS MATRIX K IS STORED IN THE Z ARRAY, Z(IZSTIP)=K(1,1) PORM0180
C   THE ABSOLUTE ADDRESS OF THE FIRST ELEMENT IS THE MD MATRIX IS IZMD. PORM0190
C   THE STIFFNESS MATRIX K IS STORED IN THE NZ ARRAY, NZ(IZMD)=MD(1)  PORM0200
C   THUS:  IF I .LE.J (UPPER TRIANGULAR MATRIX)                       PORM0210
C            IF IZMD-1+J .( IZMD-1+J-1                                PORM0220
C                  IJ=NZ(IZMD-1+J)+J-I : RELATIVE ADDRESS IF ELEMENT (I,J)PORM0230
C                  B(I,J)=Z(IZSTIP-1 + IJ)                            PORM0240
C             ELSE                                                    PORM0250
C                  B(I,J)=0             : (I,J) IS ABOVE THE SKYLINE  PORM0260
C                                                                     PORM0270
C   EXAMPLE: LET B BE THE 5X5 FULL SYMETRIC MATRIX BELOW...           PORM0280
C        |  1  |  3  |     |  12 |                                     PORM0290
C        |     |  2  |  5  |     |  11 |  NOTE: B(1,3),B(1,4) AND B(2,4) ARE ZERO. PORM0300
C   B=   |     |     |  4  |  7  |  10 |                              PORM0310
C        |     SYMM. |     |  6  |  9  |  MD= 1, 2, 4, 6, 8, 13       PORM0320
C        |           |     |     |  8  |                              PORM0330
C   SAY IZMD-121,IZSTIP=200                                           PORM0340
C   NZ(121)=1          Z(200)=1                                       PORM0350
C   NZ(122)=2          Z(201)=2                                       PORM0360
C   NZ(123)=4          Z(202)=3                                       PORM0370
C   NZ(124)=6          Z(203)=4                                       PORM0380
C   NZ(125)=8          Z(204)=5                                       PORM0390
C   NZ(126)=13         Z(205)=6                                       PORM0400
C                      Z(206)=7                                       PORM0410
C                      Z(206)=8                                       PORM0420
C                      Z(206)=9                                       PORM0430
C                      Z(206)=10                                      PORM0440
C                      Z(206)=11                                      PORM0450
C                      Z(206)=12                                      PORM0460
C   THUS, FOR B(3,4)   I=3, J=4                                       PORM0470
C              IJ=NZ(IZMD-1+J)+J-I = NZ(121-1+4) +4-3 = 6+4-3 =7      PORM0480
C              B(I,J)=Z(IZSTIP-1 + IJ) = Z(200-1+7) = Z(206) = 7  OK  PORM0490
C   THUS, FOR B(2,4)   I=2, J=4                                       PORM0500
C              IJ=NZ(IZMD-1+J)+J-I = NZ(121-1+4) +4-2 = 6+4-2 =8      PORM0510
C              NZ(IZMD -1+J+1)+NZ(121-1+4+1)=8                        PORM0520
C              SINCE IJ GE NZ(IZMD-1+J-1)                             PORM0530
C              B(I,J)=0  (I,J) IS ABOVE THE SKYLINE..                 PORM0540
C---------------------------------------------------------------------PORM0550
C                                                                     PORM0560
      BUG5 = BTEST(IBUG,5)                                            PORM0570
      BUG57= BTEST(IBUG,7) .AND.  BUG5                                PORM0580
      LSTTYP=0                                                        PORM0590
      PRINT=.FALSE.                                                   PORM0600
      HEAD = FALSE.                                                   PORM0610
C                                                                     PORM0620
C-------------- ... ... .  ------------------------ ---------- ----------PORM0630
C IOPT=1 FORM STIFFNESS MATRIX                                        PORM0640
C IOPT=2 FORM GEOMETRIC STIFFNESS MATRIX                              PORM0650
C------ -------- ------------------  ---------------------------------PORM0660
C                                                                     PORM0670
       IF (BUG5)  WRITE (6,*) ' IN ROUTINE FORM IOPT.',IOPT           PORM0680
C                                                                     PORM0690
   49 FORMAT (1X,A/('  I=',I3,'  MDMASS(I):',I11))                   PORM0700
C                                                                     PORM0710
      GO TO (100,200) IOPT                                            PORM0720
C                                                                     PORM0730
C-------------------------                                            PORM0740
C- ASSEMBLE STIFFNESS  ------                                         PORM0750
C-------------------------                                            PORM0760
  100 CONTINUE                                                        PORM0770
          IST =IZSTIF                                                 PORM0780
          IEND=IZSTIF +NZ(IZMD+NDOF) -1                               PORM0790
C                                                                     PORM0800
C---- ------ ZERO STIFFNESS                                          PORM0810
      DO 110 I=IST,IEND                                               PORM0820
  110   Z(I)=0                                                        PORM0830
C                                                                     PORM0840
C - --- - --- LOOP FOR EACH ELEMENT                                   PORM0850
      DO 120 IELNO=1,NELMT                                            PORM0860
C                                                                     PORM0870
C---------- GET STIFFNESS OF EACH ELEMENT                             PORM0880
       FALSE=.FALSE.                                                  PORM0890
       CALL ELELIB                                                    PORM0900
     &   (3,LSTTYP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DAMAGE,DUCPAG,     PORM0910
     &   IELNO ,IELDOF,KGDOF,PGEOM,Z   ,AXIAL,IZDISP,IZLOAD,MSTOR)    PORM0920
C                                                                     PORM0930
       IF (BUG57)  THEN                                               PORM0940
         WRITE (ELNO,15)  IELNO,(NZ(I+IZLM-1),I=1,MIN(IELDOF,18))    PORM0950
   15    FORMAT ('ELEMENT #',I6,' LM=',18I6)                          PORM0960
         CALL LMATRX(Z(IZKE),MD,IELDOF,MD(IELDOF+1),ELNO)             PORM0970
       ENDIF                                                          PORM0980
C                                                                     PORM0990
      CALL FORML(Z(IZSTIF),NZ(IZMD),NDOF,NZ(IZMD-NDOF),               PORM1000
     &           Z(IZKE),MD,IELDOF,MD(IELDOF-1),NZ(IZLM),1 )          PORM1010
C                                                                     PORM1020
C  - - ----- SUBTRACT GEOMETRIC STIFFNESS                            PORM1030
      NZ(IZKGDT+3)=NZ(IZKGDT+3)                                       FORM1040
C                                                                     FORM1050
      IF (NZ(IZKGDT+3).EQ.1) THEN                                     FORM1060
        FALSE=.FALSE.                                                 FORM1070
        CALL ELELIB                                                   FORM1080
     &   (2,LSTTYP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DAMAGE,DUCPAG,     FORM1090
     &   IELNO ,IELDOF,KGDOF,PGEOM,Z   ,AXIAL,IZDISP,IZLOAD,MSTOR)    FORM1100
C                                                                     FORM1110
        KGDOF=KGDOF                                                   FORM1120
C                                                                     FORM1130
        IF (BUG57) THEN                                               FORM1140
          WRITE (ELNO,15)  IELNO,(NZ(I+IZLMKG-1),I=1,MIN(KGDOF,18))  FORM1150
          CALL LMATRX(Z(IZKEKG),MD,KGDOF,MD(KGDOF+1),ELNO)            FORM1160
        ENDIF                                                         FORM1170
C                                                                     FORM1180
        IF (.NOT  AXIAL) GO TO 120                                    FORM1190
C         DETERMINE THE FACTOR FOR KG                                 FORM1200
            IF (NZ(IZKGDT+1).EQ.1) THEN                               FORM1210
                  ACCEL2= Z(IZKGDT)                                   FORM1220
                  PGEOM=PGEOM*(1-ACCEL2)                              FORM1230
            ENDIF                                                     FORM1240
C     SWAP THE SIGN ON PGEOM FOR SUBTRACTION.....                     FORM1250
      PGEOM=-PGEOM                                                    FORM1260
C                                                                     FORM1270
      CALL FORML(Z(IZSTIP),NZ(IZMD),NDOF,NZ(IZMD+NDOF),               FORM1280
     &     Z(IZKEKG),MD,KGDOF,MD(KGDOF+1),NZ(IZLMKG),PGEOM)           FORM1290
      ENDIF                                                           FORM1300
C                                                                     FORM1310
  120   CONTINUE                                                      FORM1320
C                                                                     FORM1330
      RETURN                                                          FORM1340
C                                                                     FORM1350
C---------- ASSEMBLE GEOMETRIC STIFFNESS                             FORM1360
  200 CONTINUE                                                        FORM1370
      IF (NZ(IZKGDT+3).NE.2 ) RETURN                                  FORM1380
          IST =IZKG                                                   FORM1390
          IEND=IZKG + NZ(IZMDKG+NDOF) - 1                             FORM1400
C                                                                     FORM1410
C---- ------ ZERO STIFFNESS                                          FORM1420
      DO 210 I=IST,IEND                                               FORM1430
  210   Z(I)=0                                                        FORM1440
C                                                                     FORM1450
C---------- LOOP FOR EACH ELEMENT                                    FORM1460
      DO 220 IELNO=1,NELMT                                            FORM1470
C                                                                     FORM1480
C---------- GET STIFFNESS OF EACH ELEMENT                            FORM1490
       FALSE=.FALSE.                                                  FORM1500
       CALL ELELIB                                                    FORM1510
     &   (2,LSTTYP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DAMAGE,DUCPAG,     FORM1520
     &   IELNO ,IELDOF,KGDOF,PGEOM,Z   ,AXIAL,IZDISP,IZLOAD,MSTOR)    FORM1530
C                                                                     FORM1540
       IF (.NOT  AXIAL) GO TO 220                                     FORM1550
C                                                                     FORM1560
       IF (BUG57) THEN                                                FORM1570
         WRITE (6,*)  'PGEOM=',PGEOM                                  FORM1580
         WRITE (ELNO,15)  IELNO,(NZ(I+IZLMKG-1),I=1,MIN(KGDOF,18))   FORM1590
         CALL LMATRX(Z(IZKEKG),MD,KGDOF,MD(KGDOF+1),ELNO)             FORM1600
       ENDIF                                                          FORM1610
C                                                                     FORM1620
      CALL FORML(Z(IZKG),NZ(IZMDKG),NDOF,NZ(IZMDKG+NDOF),             FORM1630
     &     Z(IZKEKG),MD,KGDOF,MD(KGDOF+1),NZ(IZLMKG),PGEOM)           FORM1640
  220   CONTINUE                                                      FORM1650
C                                                                     FORM1660
      RETURN                                                          FORM1670
      END                                                             FORM1680
C*********************************************************************CFEM30010
C                                                                     CFEM30020
C   PROGRAM FEM - ANALYSIS OF STRUCTURES                              CFEM30030
C                                                                     CFEM30040
C*********************************************************************CFEM30050
C                                                                     CFEM30060
C   NOTE. USE IBM FORTVS2 COMPILER OPTION AUTODBL(DBLPAD4)            CFEM30070
C         TO COMPILE THIS PROGRAM IN DOUBLE PRECISION                 CFEM30080
C                                                                     CFEM30090
C   NOTE: USE IBM FORTVS2 COMPILER OPTION OPT(2)                      CFEM30100
C         TO OPTIMIZE THE CODE FOR FASTER EXECUTION                   CFEM30110
C                                                                     CFEM30120
C---------------------------------------------------------------------CFEM30130
      CHARACTER*80 OPTION,INPUT,BLANK                                 FEM30140
      CHARACTER*8  TIME1,DATE1                                        FEM30150
      CHARACTER*1  CHR(0:35)                                          FEM30160
      LOGICAL TEST,HEAD,AXIAL                                         FEM30170
$INCLUDE:'ZCOMN'                                                      FEM30180
      DATA CHR/'A','B','C','D','E','F','G','H','I','J','K','L','M',  FEM30190
     &         'N','O','P','Q','R','S','T','U','V','W','Z','Y','Z'/ FEM30200
C                                                                     FEM30210
C-----------------------------                                        FEM30220
C-- ECHO THE INPUT.......   --                                        FEM30230
C-----------------------------                                        FEM30240
      I=55                                                            FEM30250
      L=0                                                             FEM30260
9995  READ(105,'(A)',END=9996) INPUT                                  FEM30270
      IF (INDEX(INPUT,'NOECHO').NE.0) GO TO 9996                      FEM30280
      I=I+1                                                           FEM30290
      L=L+1                                                           FEM30300
      IF (  I.GT.55 ) THEN                                            FEM30310
         WRITE(108,9994) (XX,XX=1,8)                                  FEM30320
         I=0                                                          FEM30330
      ENDIF                                                           FEM30340
      WRITE(108,9997) L,INPUT                                         FEM30350
      IF (INDEX(INPUT,'STOP'  ).EQ.0) GO TO 9995                      FEM30360
9994  FORMAT ('1     ECHO OF INPUT DATA '//                           FEM30370
     &        1X,3X,'LINE  ',8('. ..:...',I1,'0'))                    FEM30380
9997  FORMAT (1X,I7,':  ',A)                                          FEM30390
9996  REWIND(105)                                                     FEM30400
C                                                                     FEM30410
C-----------------------------                                        FEM30420
C-- INITIALIZE DATA.......   --                                       FEM30430
C-----------------------------                                        FEM30440
      TITLE(1)=' '                                                    FEM30450
      TITLE(2)=' '                                                    FEM30460
      BLANK  =' '                                                     FEM30470
      BUG =.FALSE.                                                    FEM30480
      DOUBLE =.FALSE.                                                 FEM30490
      IBUG=0                                                          FEM30500
      MAXEDF=24                                                       FEM30510
      MD(1)=1                                                         FEM30520
      DO 1 I=2,(MAXEDF+1)                                             FEM30530
    1   MD(I)=MD(I-1)+(I-1)                                           FEM30540
      IZ=0                                                            FEM30550
C                                                                     FEM30560
   20 READ(105,*,END=1000) OPTION                                     FEM30570
      RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                  FEM30580
C                                                                     FEM30590
C-------------------------                                            FEM30600
C-- DEFINE THE STRUCTURE ---                                          FEM30610
C-------------------------                                            FEM30620
      IF (INDEX(OPTION,'STR').NE.0)  THEN                             FEM30630
C ------------------ --------- --                                     FEM30640
          IZ=1                                                        FEM30650
          TITLE(1)=' '                                                FEM30660
          TITLE(2)=' '                                                FEM30670
C ------------------------ ------                                     FEM30680
          GRAV=386.4                                                  FEM30690
          READ(105,*,END=1000) TITLE(1)                               FEM30700
          CALL TIDAE(TIME1,DATE1)                                     FEM30710
          WRITE(TITLE(1)(80:),45) TIME1,DATE1                         FEM30720
          RATIZ=100.*REAL(IZ)/REAL(MAXZ)                              FEM30730
          WRITE(*,52)' BULIDING STRUCTURAL MODEL - ---------  -----*', FEM30740
     &                  IZ,RATIZ                                      FEM30750
          WRITE(*,51) TITLE(1)(1:70)                                  FEM30760
          CALL STRUCT                                                 FEM30770
          IZREL=IZ                                                    FEM30780
          RATIZ=100.*REAL(IZ)/REAL(MAXZ)                              FEM30790
C                                                                     FEM30800
C-------------------------------                                      FEM30810
```

```
C-- DEFINE THE LOADING AND SOLVE --                                      FEM30820
C-------------------------------                                        FEM30830
      ELSE IF (INDEX(OPTION,'SOL').NE.0)  THEN                          FEM30840
        READ(105,*,END=1000) TITLE(2)                                   FEM30850
        CALL TIDAE(TIME1,DATE1)                                         FEM30860
        WRITE(TITLE(2)(80 ),45) TIME1,DATE1                             FEM30870
        WRITE(*,52)' SOLVING STRUCTURAL MODEL ---------------- *',      FEM30880
     &                  IZ,RATIZ                                        FEM30890
        WRITE(*,51) TITLE(1)(1:70)                                      FEM30900
        WRITE(*,51) TITLE(2)(1 70)                                      FEM30910
        CALL SOLN(OPTION)                                               FEM30920
        RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                  FEM30930
C                                                                       FEM30940
C-------------------------------                                        FEM30950
C-- STOP ... END OF PROGRAM       --                                    FEM30960
C-------------------------------                                        FEM30970
      ELSE IF (INDEX(OPTION,'STOP').NE.0)  THEN                         FEM30980
        WRITE(*,52)' NORMAL STOP ------------------------------*',      FEM30990
     &                  IZ,RATIZ                                        FEM31000
        STOP                                                            FEM31010
C                                                                       FEM31020
C-------------------------------                                        FEM31030
C--  READ RESULTS FROM INPUT FILE ---                                   FEM31040
C-------------------------------                                        FEM31050
      ELSE IF (INDEX(OPTION,'READ').NE.0)  THEN                         FEM31060
  140   CALL GETINT(OPTION,'UNIT=',IUNIT,0,.TRUE.,.TRUE. )             FEM31070
        CALL GETINT(OPTION, 'INC=',INC  ,1,.TRUE.,.FALSE.)             FEM31080
        IF (IUNIT.GT.0) THEN                                            FEM31090
          WRITE(*,52) ' READING DATA FROM FILE ------- ------------*',  FEM31100
     &                  IZ,RATIZ                                        FEM31110
          IOPT=4                                                        FEM31120
          CALL DMPDAT(IOPT,IWRITE,TC,DT)                                FEM31130
          GO TO 140                                                     FEM31140
        ENDIF                                                           FEM31150
        RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                  FEM31160
C                                                                       FEM31170
C-----------------------                                                FEM31180
C-- SET BUG OPTION     --                                               FEM31190
C-----------------------                                                FEM31200
      ELSE IF (TEST(OPTION,'BUG-',.FALSE.))  THEN                       FEM31210
        CALL GETCHR(OPTION,'BUG-',INPUT,.TRUE.,.FALSE.)                 FEM31220
        IBUG=0                                                          FEM31230
        WRITE(OPTION,160) INPUT(1:70)                                   FEM31240
        DO 150 I=0,23                                                   FEM31250
          IF (TEST(INPUT,CHR(I),.FALSE.)) IBUG=IBSET(IBUG,I)            FEM31260
  150   CONTINUE                                                        FEM31270
  160   FORMAT ('BUG-',A)                                               FEM31280
        WRITE(*,52)' SET BUG OPTION: ------------------------------*',  FEM31290
     &                  IZ,RATIZ                                        FEM31300
        WRITE(*,51)  OPTION(1:70)                                       FEM31310
        WRITE(108,51)  'SET BUG OPTION: '                               FEM31320
        WRITE(108,51)  OPTION(1:70)                                     FEM31330
C                                                                       FEM31340
C-----------------------                                                FEM31350
C-- SET BUG OPTION     --                                               FEM31360
C-----------------------                                                FEM31370
      ELSE IF (INDEX(OPTION,'BUG').NE.0)  THEN                          FEM31380
        WRITE(*,51) ('SET BUG OPTION: '//OPTION(1:54) )                 FEM31390
        BUG=.TRUE.                                                      FEM31400
        IF (INDEX(OPTION,'NOBUG').NE.0) BUG=.FALSE.                     FEM31410
C                                                                       FEM31420
C-------------------------------------                                  FEM31430
C-- RELEASE MEMORY FROM LAST SOLUTION --                                FEM31440
C-------------------------------------                                  FEM31450
    ELSE IF (INDEX(OPTION,'RELEASE').NE.0)  THEN                        FEM31460
        WRITE(108,52)'- ------------------------------ ------ --*'      FEM31470
        WRITE(*,52)' RELEASING MEMORY ---------------- ---------*'      FEM31480
        WRITE(108,52)' RELEASING MEMORY ---------------- ---------*'    FEM31490
        IF (INDEX(OPTION,'ELEMENT').NE.0)  THEN                         FEM31500
        WRITE(108,52)' RESETING ELEMENT FORCES  ----------------*'      FEM31510
        WRITE(108,52)'-------------- ------------------------ --- ----*' FEM31520
          DO 170 IELNO=1,NELMT                                          FEM31530
            LSTTYP=0                                                    FEM31540
            HEAD = FALSE.                                               FEM31550
  170       CALL ELELIB                                                 FEM31560
     &    (13,LSTTYP, FALSE.,IREL,HEAD,INPUT,EESE,EPSE,DAMAGE,DUCFAG,   FEM31570
     &     IELNO,IELDOF,XGDOF,PGEOM,Z ,AXIAL,IZDDSP,IZDLOA,MSTOR)       FEM31580
        ELSE                                                            FEM31590
        WRITE(108,52)'- ------------------------------------*'          FEM31600
        ENDIF                                                           FEM31610
        IZ=IZREL                                                        FEM31620
          IZACC =0                                                      FEM31630
          IZDACC=0                                                      FEM31640
          IZDDSP=0                                                      FEM31650
          IZDISP=0                                                      FEM31660
          IZDLOA=0                                                      FEM31670
          IZDVEL=0                                                      FEM31680
          IZFUB =0                                                      FEM31690
          IZLOAD=0                                                      FEM31700
          IZVEL =0                                                      FEM31710
        RATIZ=100 *REAL(IZ)/REAL(MAXZ)                                  FEM31720
C                                                                       FEM31730
C-------------------------------------                                  FEM31740
C-- SAVE MEMORY FOR A RESTART       --                                  FEM31750
C-------------------------------------                                  FEM31760
      ELSE IF (INDEX(OPTION,'SAVE').NE.0)  THEN                         FEM31770
        WRITE(*,52)' SAVING DATA  ---------- ------------------*',      FEM31780
     &                  IZ,RATIZ                                        FEM31790
        WRITE(*,51) TITLE(1)(1:70)                                      FEM31800
        WRITE(*,51) TITLE(2)(1:70)                                      FEM31810
        WRITE(108,200)                                                  FEM31820
  200   FORMAT(5X,'******** WRITING DATA TO UNIT FT09 ********')        FEM31830
        OPEN(UNIT=9,ACCESS='SEQUENTIAL',STATUS='UNKNOWN',               FEM31840
     &       FORM='UNFORMATTED')                                        FEM31850
        WRITE(9)   TITLE                                                FEM31860
        WRITE(9)   IZ    ,IZREL ,                                       FEM31870
     &    NDOF  ,NCOND ,NFREE ,NREST ,MAXNOD,NMAT  ,NELMT ,NNODE ,     FEM31880
     &    NLOAD ,NSOLN ,MAXEDF,MD,NCOS,MAXELD,NELD,                    FEM31890
     &    IZID  ,IZIDOF,IZCORD,IZCOS ,IZCNST,IZMAT ,IZELE ,            FEM31900
     &    IZJFLG,IZJCOS,                                               FEM31910
     &    IZKE  ,IZLM  ,IZSTIF,IZMD  ,IZLOAD,IZDISP,IZIELD,IZELD       FEM31920
        DO 210 J=1,IZ,256                                               FEM31930
          K=J+255                                                       FEM31940
          K=MIN(K,IZ)                                                   FEM31950
  210     WRITE(9) (Z(I),I=J,K)                                         FEM31960
        CLOSE (UNIT=9)                                                  FEM31970
        RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                  FEM31980
C                                                                       FEM31990
C-------------------------------------                                  FEM32000
C-- RESTART PROGRAM                 --                                  FEM32010
C-------------------------------------                                  FEM32020
        WRITE(*,52)' RETREVING DATA ------ --------------------*',      FEM32030
     &                  IZ,RATIZ                                        FEM32040
      ELSE IF (INDEX(OPTION,'RESTART') NE 0)  THEN                      FEM32050
        WRITE(108,300)                                                  FEM32060
  300   FORMAT(5X,'******** READING DATA FROM UNIT FT09 ********')      FEM32070
        OPEN(UNIT=9,ACCESS='SEQUENTIAL',STATUS='OLD',                   FEM32080
     &       FORM='UNFORMATTED')                                        FEM32090
        READ  (9)   TITLE                                               FEM32100
        READ  (9)   IZ    ,IZREL ,                                      FEM32110
     &    NDOF  ,NCOND ,NFREE ,NREST ,MAXNOD,NMAT  ,NELMT ,NNODE ,     FEM32120
     &    NLOAD ,NSOLN ,MAXEDF,MD,NCOS,MAXELD,NELD,                    FEM32130
     &    IZID  ,IZIDOF,IZCORD,IZCOS ,IZCNST,IZMAT ,IZELE ,            FEM32140
     &    IZJFLG,IZJCOS,                                               FEM32150
     &    IZKE  ,IZLM  ,IZSTIF,IZMD  ,IZLOAD,IZDISP,IZIELD,IZELD       FEM32160
        DO 310 J=1,IZ,256                                               FEM32170
          K=J+255                                                       FEM32180
          K=MIN(K,IZ)                                                   FEM32190
  310     READ  (9) (Z(I),I=J,K)                                        FEM32200
        CLOSE (UNIT=9)                                                  FEM32210
        WRITE(*,52)' DATA RETREIVED ------------------------- -*',      FEM32220
     &                  IZ,RATIZ                                        FEM32230
        WRITE(*,51) TITLE(1)(1:70)                                      FEM32240
        WRITE(*,51) TITLE(2)(1:70)                                      FEM32250
        RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                  FEM32260
C                                                                       FEM32270
C---------------------------------------                               FEM32280
C-- INVALID OPTION                    --                                FEM32290
C---------------------------------------                               FEM32300
      ELSE                                                              FEM32310
          IF (TEST(OPTION,'NOECHO',.FALSE.) ) GO TO 20                  FEM32320
          IF (OPTION.EQ.BLANK) GO TO 20                                 FEM32330
          WRITE(108,30) OPTION                                          FEM32340
          WRITE(*,51) ('INVALID OPTION: '//OPTION(1:54))                FEM32350
      ENDIF                                                             FEM32360
      GO TO  20                                                         FEM32370
C                                                                       FEM32380
   45 FORMAT(' TIME  ',A,',  DATE: ',A)                                 FEM32390
   50 FORMAT (A1,' STRUCTURE....: ',A,/1X,' SOLUTION...  : ',A,/)       FEM32400
   51 FORMAT (1X,A)                                                     FEM32410
   52 FORMAT (1X,'+----',A45,'+',/,' IZ=',I7,' MEM=',F6.3,'%')          FEM32420
   30 FORMAT ('1 INVALID OPTION- ',A)                                   FEM32430
  110 FORMAT (' ZMATRX(',I6,')''',I15,1P,G20.10,:,G20.10)               FEM32440
 1000 STOP                                                              FEM32450
      END                                                               FEM32460
C                                                                       TIDA0010
C ----- TAKE PRESENT TIME AND DATE ----                                 TIDA0020
C                                                                       TIDA0030
      SUBROUTINE TIDAE(TIME1,DATE1)                                     TIDA0040
       INTEGER*2 IYR,IMON,IDAY,IHR,IMIN,ISEC,I100TH                     TIDA0050
       CHARACTER*8 TIME1,DATE1                                          TIDA0060
       CALL GETDAT(IYR,IMON,IDAY)                                       TIDA0070
       IYR=IYR-1900                                                     TIDA0080
       CALL GETTIM(IHR,IMIN,ISEC,I100TH)                                TIDA0090
       WRITE(DATE1,'(i2,1h/,i2,1h/,i2)') IMON,IDAY,IYR                  TIDA0100
       WRITE(TIME1,'(i2,1h:,i2,1h:,i2)') IHR,IMIN,ISEC                  TIDA0110
      RETURN                                                            TIDA0120
      END                                                               TIDA0130
C                                                                       SOLN0010
C-----------------------------------------------------------------------SOLN0020
C                                                                       SOLN0030
      SUBROUTINE SOLN(OPTION)                                           SOLN0040
      CHARACTER*80 OPTION                                               SOLN0050
$INCLUDE:'ZCOMN'                                                        SOLN0060
      STEP1D=' '                                                        SOLN0070
C                                                                       SOLN0080
C-------------------------                                              SOLN0090
C- DETERMINE SOLUTION NO. --                                            SOLN0100
C-------------------------                                              SOLN0110
      CALL GETINT(OPTION,'SOL',NSOLN,0,.TRUE.,.FALSE.)                  SOLN0120
C                                                                       SOLN0130
C-------------------------                                              SOLN0140
C- CHOOSE THE SOLUTION --                                               SOLN0150
C-------------------------                                              SOLN0160
C     IF (NSOLN.EQ.1) THEN                                              SOLN0170
C           CALL SOL01                                                  SOLN0180
C     ELSE IF (NSOLN.EQ.2) THEN                                         SOLN0190
C           NLOAD=1                                                     SOLN0200
C           MAXELD=0                                                    SOLN0210
C           CALL SOL02                                                  SOLN0220
C     ELSE IF (NSOLN.EQ.3) THEN                                         SOLN0230
      IF (NSOLN.EQ.3) THEN                                              SOLN0240
            NLOAD=1                                                     SOLN0250
            MAXELD=0                                                    SOLN0260
            CALL SOL03                                                  SOLN0270
C     ELSE IF (NSOLN.EQ.4) THEN                                         SOLN0280
C           NLOAD=1                                                     SOLN0290
C           CALL SOL04                                                  SOLN0300
C     ELSE IF (NSOLN.EQ.5) THEN                                         SOLN0310
C           NLOAD=1                                                     SOLN0320
C           MAXELD=0                                                    SOLN0330
C           CALL SOL05                                                  SOLN0340
      ELSE                                                              SOLN0350
            WRITE(108,*) 'INVALID SOLN#',NSOLN                          SOLN0360
            STOP 'INVALID SOLUTION NUMBER'                              SOLN0370
      ENDIF                                                             SOLN0380
      RETURN                                                            SOLN0390
      END                                                               SOLN0400
C                                                                       MSOL0010
C-----------------------------------------------------------------------MSOL0020
C  MATRIX SOLUTION OF EQUATION AX=P, WHERE THE UPPER TRIANGULAR MATRIX   MSOL0030
C  OF THE SYMMETRIC MATRIX A IS STORED BY SKYLINES  P IS STORED AS A     MSOL0040
C  FULL MATRIX.  THE MATRIX P IS REPLACED BY X AFTER BACK SUBSTITUTION   MSOL0050
C  IS COMPLEATE.  BOTH A AND P ARE ALTERED.....                          MSOL0060
C  GUYAN OPTION. IF GUYAN IS .TRUE. MASS MATRIX IS ALSO CONDENSED        MSOL0070
C-----------------------------------------------------------------------MSOL0080
      SUBROUTINE MSOLL(IOPT,PRINT,N,IMD,LROW,NLC,L1,L2,MD,A,P,FACTOR,   MSOL0090
     &                 GUYAN,MASS,MDMASS,IMDMAS,KGCOND,KG,MDKG,IMDKG)   MSOL0100
      REAL   MASS(IMDMAS),KG(IMDKG)                                     MSOL0110
      LOGICAL GUYAN,PRINT,KGCOND                                        MSOL0120
      DIMENSION A(IMD),FACTOR(N),P(LROW,NLC),MD(N-1)                    MSOL0130
      DIMENSION MDMASS(N-1),MDKG(N-1)                                   MSOL0140
C                                                                       MSOL0150
      IJI(I,J)=MD(J)+J-I                                                MSOL0160
      IJM(I,J)=MDMASS(J)+J-I                                            MSOL0170
      IJKG(I,J)=MDKG(J)+J-I                                             MSOL0180
C---- ------------------------------ ------------------------- ---------MSOL0190
C  MATRIX STORAGE SCHEME:                                               MSOL0200
C  THE MATRIX A IS BANDED AND SYMMETRIC. THUS ONLY THE SKYLINE ABOVE    MSOL0210
C  AND THE MAIN DIAGONAL NEEDS TO BE STORED.                            MSOL0220
C  THE MATRIX A IS STORED IN A LINEAR ARRAY BY COLUMNS.  THE VALUE ON   MSOL0230
C  THE MAIN DIAGONAL IS STORED IN THE LINEAR ARRAY FIRST                MSOL0240
C  ADDRESSES OF THE LINEAR ARRAY ELEMENTS CONTAINING THE MAIN DIAGONAL  MSOL0250
C  TERMS ARE STORED IN MD.                                              MSOL0260
C  EXAMPLE: LET B BE THE 5X5 FULL SYMETRIC MATRIX BELOW. .              MSOL0270
C     |  1  |  3  |     |     |  12 |                                    MSOL0280
C     |     |  2  |  5  |     |  11 |  NOTE. B(1,3),B(1,4) AND B(2,4) ARE ZERO. MSOL0290
C  B= |     |     |  4  |  7  |  10 |                                    MSOL0300
C     |     | SYMM|     |  6  |  9  |  MD= 1, 2, 4, 6, 8, 13            MSOL0310
C     |     |     |     |     |  8  |                                    MSOL0320
C  THUS, B(1,3)=A( MD(3) )= A(4)                                        MSOL0330
C        B(I,J)=A( MD(J)-J-I ) IF J>I AND (MD(J)-J-I)<MD(J-1)           MSOL0340
C              =0                IF J>I AND (MD(J)-J-I)>MD(J-1)        MSOL0350
C  MATRIX P IS NOT SYMMETRIC, AND IS STORED AS A FULL MATRIX            MSOL0360
C---- ------------------------------ ------------------------- ---------MSOL0370
C  VARIABLE TABLE                                                       MSOL0380
C  IOPT  - OPTION NUMBER                                                MSOL0390
C  N     - NUMBER OF DEGREES OF FREEDOM OF A MATRIX                     MSOL0400
C  NLC   - NUMBER OF LOAD CASES FOR THE P MATRIX                        MSOL0410
C  L1    - FIRST ROW TO BE WORKED WITH                                  MSOL0420
C  L2    - LAST ROW TO BE WORKED WITH                                   MSOL0430
C  MD    - ADDRESS OF THE MAIN DIAGONAL TERMS OF A                      MSOL0440
C  P     - LOAD MATRIX ON INPUT, SOLN (X) OF AX=P ON COMPLEATION OF      MSOL0450
C          IOPT=4                                                       MSOL0460
C  FACTOR- TEMPORARY STORAGE MATRIX                                     MSOL0470
C  GUYAN - GUYAN REDUCTION FLAG FOR MASS MATRIX                         MSOL0480
C  MASS  - MASS MATRIX                                                  MSOL0490
C  MDMASS- MASS MATRIX ADDRESSES OF MAIN DIAGONAL ELEMENTS              MSOL0500
C  KGCOND- FLAG FOR CONDENSING OUT KG                                   MSOL0510
C  KG    - GEOMETRIC STIFFNESS MATRIX                                   MSOL0520
C  MDKG  - KG MATRIX ADDRESSES OF MAIN DIAGONAL ELEMENTS                MSOL0530
C  L     - ROW NUMBER FOR ELIMINATION                                   MSOL0540
C  I     - ROW NUMBER                                                   MSOL0550
C  J     - COLUMN NUMBER                                                MSOL0560
C---- ------ ---- ------- ---------- ------------------------ ---------MSOL0570
C                                                                       MSOL0580
      L1=MAX(L1,1)                                                      MSOL0590
      L2=MIN(L2,N)                                                      MSOL0600
C                                                                       MSOL0610
      GO TO (10,110,210,310) IOPT                                       MSOL0620
C                                                                       MSOL0630
   10 CONTINUE                                                          MSOL0640
C--------- IOPT=1, REDUCE A AND P  ------------------------- ---  -----MSOL0650
C         GAUSSIAN ELIMINIATION IS USED TO REDUCE THE A AND P MATRICIESMSOL0660
C         FROM ROW L1 TO ROW L2.  IF L1 IS NOT 1, IT IS ASSUMED THAT   MSOL0670
C         A AND P ARE ALLREADY REDUCED FROM 1 TO L1 ..                 MSOL0680
C---- ------ ---- ------- ---------- ------------------------ ---------MSOL0690
      IF (PRINT) THEN                                                   MSOL0700
        WRITE(108,500) 'IOPT=1, REDUCE STIFFNESS AND LOADS',L1,L2,N     MSOL0710
        CALL LMATRX(A,MD,N,IMD ,'INPUT STIFFNESS')                      MSOL0720
        CALL WMATRX(P,N ,NLC,'INPUT LOAD')                              MSOL0730
        IF (GUYAN ) CALL LMATRX(MASS,MDMASS,N,IMDMAS,'INPUT MASS')      MSOL0740
```

```
      IF (KGCOND) CALL LMATRX(KG  ,MDKG  ,N,IMDKG ,'INPUT KG  ')
      ENDIF
C
      DO 100 L=L1,L2
        IF (L.EQ.N) GO TO 100
C
C------- TEST FOR ZERO PIVOT ... ... ... .....
        IF (A(MD(L)).EQ.0) THEN
          WRITE(108,2000) L,MD(L),A(MD(L))
          STOP
        ENDIF
C
C-- -- -- CALCULATE THE FACTORS EACH ROW IS MULTIPLIED BY
        DO 30 I=(L+1),N
          LI=IJI(L,I)
          IF ( LI.LT.MD(I+1) ) THEN
           FACTOR(I)=A(LI) /A( MD(L) )
          ELSE
           FACTOR(I)=0
          ENDIF
   30     CONTINUE
C
C------- REDUCE A BY COLUMNS (J) EACH ROW (I) ..
        DO 40 J=(L+1),N
          LJ=IJI(L,J)
          IF (LJ.GE.MD(J+1)) GO TO 40
          DO 35 I=(L+1),J
            IJ=IJI(I,J)
            A(IJ)=A(IJ)-FACTOR(I)*A(LJ)
   35       CONTINUE
   40   CONTINUE
C
C------- REDUCE P BY COLUMNS (J) EACH ROW (I)...
        DO 50 J=1,NLC
          IF (P(L,J).EQ.0) GO TO 50
          DO 45 I=(L+1),N
   45       P(I,J)=P(I,J)-FACTOR(I)*P(L,J)
   50   CONTINUE
C
C------- REDUCE MASS MATRIX BY GUYAN REDUCTION
      IF (GUYAN) THEN
C
C.......  M22=M22 - K21/K11 * M11 - M11 * 1/K11 * K12
        RM=MASS(MDMASS(L))
        IF (RM.NE.0) THEN
            DO 60 J=L+1,N
                IF (FACTOR(J).EQ.0) GO TO 60
                II=MAX(L+1,J-(MDMASS(J+1)-MDMASS(J)) )
                DO 55 I=II,J
                  IJ=IJM(I,J)
   55             MASS(IJ)=MASS(IJ)+FACTOR(I)*RM*FACTOR(J)
   60         CONTINUE
        ENDIF
C
C.. .. .  M22=M22 - K21/K11 * M12
        DO 66 J=L+1,N
            LJ=IJM(L,J)
            IF (LJ.GE.MDMASS(J+1)) GO TO 66
            IF (MASS(LJ).EQ.0) GO TO 66
            DO 65 I=L+1,J
              IJ=IJM(I,J)
              IF (IJ.LT.MDMASS(J+1))
     &          MASS(IJ)=MASS(IJ)-FACTOR(I)*MASS(LJ)
   65       CONTINUE
   66     CONTINUE
C
C     ..... M22=M22 - M21 * 1/K11 * K12
        DO 70 J=L+1,N
            IF (FACTOR(J).EQ.0) GO TO 70
            DO 69 I=L+1,J
                LI=IJM(L,I)
                IF (LI.GE.MDMASS(I+1)) GO TO 69
                IJ=IJM(I,J)
                IF (IJ.LT.MDMASS(J+1))
     &            MASS(IJ)=MASS(IJ)-FACTOR(J)*MASS(LI)
   69       CONTINUE
   70     CONTINUE
      ENDIF
C
C------- REDUCE GEOMETRIC STIFFNESS MATRIX
      IF (KGCOND) THEN
C
C. ....... KG22=KG22 - K21/K11 * KG11 * 1/K11 * KG12
        RKG=KG(MDKG(L))
        IF (RKG.NE.0) THEN
            DO 80 J=L+1,N
                IF (FACTOR(J).EQ.0) GO TO 80
                II=MAX(L+1,J-(MDKG(J+1)-MDKG(J)) )
                DO 75 I=II,J
                  IJ=IJKG(I,J)
   75             KG(IJ)=KG(IJ)+FACTOR(I)*RKG*FACTOR(J)
   80         CONTINUE
        ENDIF
C
C... . . .  KG22=KG22 - K21/K11 * KG12
        DO 86 J=L+1,N
            LJ=IJKG(L,J)
            IF (LJ.GE.MDKG(J+1)) GO TO 86
            IF (KG(LJ).EQ.0) GO TO 86
            DO 85 I=L+1,J
                IJ=IJKG(I,J)
                IF (IJ.LT.MDKG(J+1))
     &            KG(IJ)=KG(IJ)-FACTOR(I)*KG(LJ)
   85       CONTINUE
   86     CONTINUE
C
C. .......  KG22=KG22 - KG21 * 1/K11 * K12
        DO 90 J=L+1,N
            IF (FACTOR(J).EQ.0) GO TO 90
            DO 89 I=L+1,J
                LI=IJKG(L,I)
                IF (LI.GE.MDKG(I+1)) GO TO 89
                IJ=IJKG(I,J)
                IF (IJ.LT.MDKG(J+1))
     &            KG(IJ)=KG(IJ)-FACTOR(J)*KG(LI)
   89       CONTINUE
   90     CONTINUE
C
      ENDIF
C
  100 CONTINUE
      IF (PRINT) THEN
        CALL LMATRX(A,MD,N,IMD,'REDUCED STIFFNESS')
        CALL WMATRX(P,N ,NLC,'REDUCED LOAD')
        IF (GUYAN)  CALL LMATRX(MASS,MDMASS,N,IMDMAS,'REDUCED MASS')
        IF (KGCOND) CALL LMATRX(KG  ,MDKG  ,N,IMDKG ,'REDUCED KG')
      ENDIF
      RETURN
C
  110 CONTINUE
C---- ---- IOPT-2, REDUCE P ONLY ------- ... ---------- - ------ ----
C           GAUSSIAN ELIMINIATION IS USED TO REDUCE THE P MATRIX
C           FROM ROW L1 TO ROW L2.  IF L1 IS NOT 1, IT IS ASSUMED THAT
C           A AND P ARE ALLREADY REDUCED FROM 1 TO L1 ..
C------- ---- ---------- ---------- ---------- ----------
      IF (PRINT) THEN
        WRITE(108,500) 'IOPT-2, REDUCE LOADS ONLY',L1,L2,N
        CALL LMATRX(A,MD,N,MD(N+1),'REDUCED STIFFNESS')
        CALL WMATRX(P,N ,NLC,'NEW LOAD MATRIX ')
      ENDIF
C
C------- LOOP FOR EACH ROW (L) TO REDUCE
        DO 160 L=L1,L2
C
        IF (A(MD(L)).EQ.0) THEN
          WRITE(108,2000) L,MD(L),A(MD(L))
          STOP
        ENDIF
C
C       LOOP FOR EACH COLUMN (J)
        DO 150 J=1,NLC
          IF (P(L,J).EQ.0) GO TO 150
          DO 120 I=(L+1),N
            LI=IJI(L,I)
            IF (LI.LT.MD(I+1)) P(I,J)=P(I,J)-A(LI)*P(L,J)/A(MD(L))
  120       CONTINUE
  150   CONTINUE
  160   CONTINUE
      RETURN
C
  210 CONTINUE
C--------- IOPT-3, BACK SUBSTUTION -------- ------- --------- ----------
C           BACK SUBSTUTION IS USED TO SOLVE FOR X IN SX=V, WHERE S IS
C           THE REDUCE UPPER TRIANGULAR FACTOR OF THE A MATRIX, AND V
C           IS THE REDUCED FORRM OF THE P MATRIX.  THE RESULTS (X) ARE
C           STORED IN P
C           BACK SUBSTUTION ONLY TAKES PLACE BETWEEN ROWS L1 AND L2.
C------------------------------------------------------------ ----------
      IF (PRINT) THEN
        WRITE(108,500) 'IOPT-3, BACK SUBSTUTION',L1,L2,N
        CALL LMATRX(A,MD,N,MD(N+1),'REDUCED STIFFNESS')
        CALL WMATRX(P,N ,NLC,'REDUCED LOAD')
      ENDIF
C
C------ LOOP FOR EACH ROW L
C     CALCULAGE X FOR EACH LOAD CASE, STORE IN P
      DO 250 J=L2,L1,-1
      IF (A(MD(J)).EQ.0) THEN
        WRITE(108,2000) J,MD(J),A(MD(J))
        STOP
      ENDIF
      DO 215 LC=1,NLC
  215   P(J,LC)=P(J,LC)/A(MD(J))
      IF (J.GT.1) THEN
       IJ=MD(J+1)
       NI=J-(MD(J+1)-MD(J)) +1
       DO 220 I=NI,J-1,+1
         IJ=IJ-1
         DO 220 LC=1,NLC
           P(I,LC)=P(I,LC)-P(J,LC)*A(IJ)
  220  CONTINUE
      ENDIF
  250  CONTINUE
      IF (PRINT) CALL WMATRX(P,N ,NLC,'DISPLACEMENT')
C
      RETURN
C
  310 CONTINUE
C--------- IOPT-4, REDUCED MULTIPLICATION ------------- ----------
C           SOLVE FOR V IN S*D-P WHERE S HAS BEEN REDUCED BY GAUSSIAN
C           ELIMINATION.
C           THE RESULTING LOAD IS STORED IN P.
C           THE DISPLACEMENTS ARE INPUT IN D.
C           MULTIPLACATION ONLY TAKES PLACE BETWEEN ROWS L1 AND L2
C------------------------------------------------------------ ----------
      IF (PRINT) THEN
        WRITE(108,500) 'IOPT-4, REDUCED MULTIPLACATION',L1,L2,N
        CALL LMATRX(A,MD,N,MD(N+1),'REDUCED STIFFNESS')
        CALL WMATRX(FACTOR,N ,1,'DISPLACEMENT')
      ENDIF
C
C------ ONLY ONE LOAD CASE....
      LC=1
C     ---- ZERO P ----
      DO 320 J=L1,L2
        IF (A(MD(J)).EQ 0) THEN
          WRITE(108,2000) J,MD(J),A(MD(J))
          STOP
        ENDIF
  320   P(J,LC)=0
C
C------------ MULTIPLY S_RED * DISP = P
      DO 340 J=L1,N
        IC1=J-(MD(J+1)-MD(J)) +1
        IC2=MAX(IC1,L1)
        DO 340 I=J,IC2,-1
          IJ=IJI(I,J)
  340     P(I,LC)=P(I,LC)+A(IJ)*FACTOR(J)
      IF (PRINT) CALL WMATRX(P,N ,1,'REDUCED LOAD')
C
C--- --- UNREDUCE THE LOAD MATRIX ...
      DO 360 L=L2,L1,-1
C
C       LOOP FOR EACH COLUMN (J)
        DO 350 J=1,NLC
          IF (P(L,LC).EQ.0) GO TO 350
          DO 345 I=(L+1),N
            LI=IJI(L,I)
            IF (LI.LT.MD(I+1)) P(I,LC)=P(I,LC)+A(LI)*P(L,LC)/A(MD(L))
  345       CONTINUE
  350   CONTINUE
  360   CONTINUE
  400   CONTINUE
C
      IF (PRINT) CALL WMATRX(P,N ,1,'LOAD')
  500 FORMAT (/2X,A,2X,'L1 :',I5,'   L2:',I5,'   N:',I5)
 2000 FORMAT (///1X,50('*')/
     &        ' *  ER','ROR IN SUBROUTINE MSOLL',       T51,'*'/
     &        ' *  PIVOT IS LE 0 -------- ROW:',I5,     T51,'*'/
     &        ' *                  STORAGE LCN:',I5,    T51,'*'/
     &        ' *                        PIVOT.',1P,G15.6,T51,'*'/
     &        ' *  SOLUTION IS ABORTED ',               T51,'*'/
     &        1X,50('*')/)
C
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE REACTN(IOPT,PRINT,NNODE,L1,L2,L3,L4,FORCE,DISPL,
     &                  MD,A,KM,NDOF,NLOAD,FACTOR,
     &                  MAXNOD,IDOF,COORD,ID,TITLE,STEPID,HEAD,
     &                  NCOS,COSINE,JTCOS,JTPLG,SUM)
      CHARACTER*(*) TITLE(2),STEPID
      CHARACTER*15 CD(6)
      LOGICAL PRINT,HEAD,PLAG,BTEST
      DIMENSION A(KM),MD(NDOF+1),IDOF(MAXNOD,6),G(6)
      DIMENSION FORCE(NDOF,NLOAD),DISPL(NDOF,NLOAD)
      DIMENSION ID(MAXNOD),COORD(MAXNOD,3),JTPLG(MAXNOD)
      DIMENSION F(6),SUM(6),COSINE(3,3,NCOS),JTCOS(MAXNOD)
C
      IF (IOPT.EQ 1) THEN
C
C-----------------------------------------
C- MODIFY LOADS FOR RESTRAINT DISPLACEMENTS ---
C-----------------------------------------
C CONSIDER THE PARTITIGNED STIFFNESS WHERE X2 ARE THE RESTRAINT DISPL.
C
C   |F1|   | K11 | K12 |  |X1|
C   |--| = |---- +-----| *|--|
C   |F2|   | K21 | K22 |  |X2|
C
C   THE MODIFIED FREE LOAD IS (F1-K12*X2) - K11*X1
C------------  --------- ------ ---- ------------C
C     DO 10 I=L1,L2                                C  FULL STORAGE
C     DO 10 J=1,NLOAD                              C  EQUIVALENT
C     DO 10 K=L3,L4                                C
C  10   FORCE(I,J)=FORCE(I,J)-K(I,K)*DISPL(K,J)    C
C-------------------------------------------------C
      DO 20 K=L3,L4
      MDK=MD(K+1)
        DO 20 J=1,NLOAD
          STOP
        ENDIF
C
C       LOOP FOR EACH COLUMN (J)
        DO 150 J=1,NLC
          IF (P(L,J).EQ.0) GO TO 150
          DO 120 I=(L+1),N
            LI=IJI(L,I)
            IF (LI.LT.MD(I+1)) P(I,J)=P(I,J)-A(LI)*P(L,J)/A(MD(L))
  120       CONTINUE
  150   CONTINUE
  160   CONTINUE
      RETURN
C
  210 CONTINUE
C--------- IOPT-3, BACK SUBSTUTION -------- ------- --------- ----------
C           BACK SUBSTUTION IS USED TO SOLVE FOR X IN SX=V, WHERE S IS
C           THE REDUCE UPPER TRIANGULAR FACTOR OF THE A MATRIX, AND V
C           IS THE REDUCED FORRM OF THE P MATRIX.  THE RESULTS (X) ARE
C           STORED IN P
C           BACK SUBSTUTION ONLY TAKES PLACE BETWEEN ROWS L1 AND L2.
C------------------------------------------------------------ ----------
      IF (PRINT) THEN
        WRITE(108,500) 'IOPT-3, BACK SUBSTUTION',L1,L2,N
        CALL LMATRX(A,MD,N,MD(N+1),'REDUCED STIFFNESS')
        CALL WMATRX(P,N ,NLC,'REDUCED LOAD')
      ENDIF
C
C------ LOOP FOR EACH ROW L
C     CALCULAGE X FOR EACH LOAD CASE, STORE IN P
      DO 250 J=L2,L1,-1
      IF (A(MD(J)).EQ.0) THEN
        WRITE(108,2000) J,MD(J),A(MD(J))
        STOP
      ENDIF
      DO 215 LC=1,NLC
  215   P(J,LC)=P(J,LC)/A(MD(J))
      IF (J.GT.1) THEN
       IJ=MD(J+1)
       NI=J-(MD(J+1)-MD(J)) +1
       DO 220 I=NI,J-1,+1
         IJ=IJ-1
         DO 220 LC=1,NLC
           P(I,LC)=P(I,LC)-P(J,LC)*A(IJ)
  220  CONTINUE
      ENDIF
  250  CONTINUE
      IF (PRINT) CALL WMATRX(P,N ,NLC,'DISPLACEMENT')
C
      RETURN
C
  310 CONTINUE
C--------- IOPT-4, REDUCED MULTIPLICATION ------------- ----------
C           SOLVE FOR V IN S*D-P WHERE S HAS BEEN REDUCED BY GAUSSIAN
C           ELIMINATION.
C           THE RESULTING LOAD IS STORED IN P.
C           THE DISPLACEMENTS ARE INPUT IN D.
C           MULTIPLACATION ONLY TAKES PLACE BETWEEN ROWS L1 AND L2
C------------------------------------------------------------ ----------
      IF (PRINT) THEN
        WRITE(108,500) 'IOPT-4, REDUCED MULTIPLACATION',L1,L2,N
        CALL LMATRX(A,MD,N,MD(N+1),'REDUCED STIFFNESS')
        CALL WMATRX(FACTOR,N ,1,'DISPLACEMENT')
      ENDIF
C
C------ ONLY ONE LOAD CASE....
      LC=1
C     ---- ZERO P ----
      DO 320 J=L1,L2
        IF (A(MD(J)).EQ 0) THEN
          WRITE(108,2000) J,MD(J),A(MD(J))
          STOP
        ENDIF
  320   P(J,LC)=0
C
C------------ MULTIPLY S_RED * DISP = P
      DO 340 J=L1,N
        IC1=J-(MD(J+1)-MD(J)) +1
        IC2=MAX(IC1,L1)
        DO 340 I=J,IC2,-1
          IJ=IJI(I,J)
  340     P(I,LC)=P(I,LC)+A(IJ)*FACTOR(J)
      IF (PRINT) CALL WMATRX(P,N ,1,'REDUCED LOAD')
C
C--- --- UNREDUCE THE LOAD MATRIX ...
      DO 360 L=L2,L1,-1
C
C       LOOP FOR EACH COLUMN (J)
        DO 350 J=1,NLC
          IF (P(L,LC).EQ.0) GO TO 350
          DO 345 I=(L+1),N
            LI=IJI(L,I)
            IF (LI.LT.MD(I+1)) P(I,LC)=P(I,LC)+A(LI)*P(L,LC)/A(MD(L))
  345       CONTINUE
  350   CONTINUE
  360   CONTINUE
  400   CONTINUE
C
      IF (PRINT) CALL WMATRX(P,N ,1,'LOAD')
  500 FORMAT (/2X,A,2X,'L1 :',I5,'   L2:',I5,'   N:',I5)
 2000 FORMAT (///1X,50('*')/
     &        ' *  ER','ROR IN SUBROUTINE MSOLL',       T51,'*'/
     &        ' *  PIVOT IS LE 0 -------- ROW:',I5,     T51,'*'/
     &        ' *                  STORAGE LCN:',I5,    T51,'*'/
     &        ' *                        PIVOT.',1P,G15.6,T51,'*'/
     &        ' *  SOLUTION IS ABORTED ',               T51,'*'/
     &        1X,50('*')/)
C
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE REACTN(IOPT,PRINT,NNODE,L1,L2,L3,L4,FORCE,DISPL,
     &                  MD,A,KM,NDOF,NLOAD,FACTOR,
     &                  MAXNOD,IDOF,COORD,ID,TITLE,STEPID,HEAD,
     &                  NCOS,COSINE,JTCOS,JTPLG,SUM)
      CHARACTER*(*) TITLE(2),STEPID
      CHARACTER*15 CD(6)
      LOGICAL PRINT,HEAD,PLAG,BTEST
      DIMENSION A(KM),MD(NDOF+1),IDOF(MAXNOD,6),G(6)
      DIMENSION FORCE(NDOF,NLOAD),DISPL(NDOF,NLOAD)
      DIMENSION ID(MAXNOD),COORD(MAXNOD,3),JTPLG(MAXNOD)
      DIMENSION F(6),SUM(6),COSINE(3,3,NCOS),JTCOS(MAXNOD)
C
      IF (IOPT.EQ 1) THEN
C
C-----------------------------------------
C- MODIFY LOADS FOR RESTRAINT DISPLACEMENTS ---
C-----------------------------------------
C CONSIDER THE PARTITIGNED STIFFNESS WHERE X2 ARE THE RESTRAINT DISPL.
C
C   |F1|   | K11 | K12 |  |X1|
C   |--| = |---- +-----| *|--|
C   |F2|   | K21 | K22 |  |X2|
C
C   THE MODIFIED FREE LOAD IS (F1-K12*X2) - K11*X1
C------------  --------- ------ ---- ------------C
C     DO 10 I=L1,L2                                C  FULL STORAGE
C     DO 10 J=1,NLOAD                              C  EQUIVALENT
C     DO 10 K=L3,L4                                C
C  10   FORCE(I,J)=FORCE(I,J)-K(I,K)*DISPL(K,J)    C
C-------------------------------------------------C
      DO 20 K=L3,L4
      MDK=MD(K+1)
        DO 20 J=1,NLOAD
```

```
      DO 10 I=L2,L1,-1                                                  REAC0370
        IK=MD(K)+K-I                                                    REAC0380
        IF (IK.GE.MDK) GO TO 20                                         REAC0390
        FIJ=FORCE(I,J)                                                  REAC0400
        AIK=A(IK)                                                       REAC0410
        DKJ=DISPL(K,J)                                                  REAC0420
   10   FORCE(I,J)=FORCE(I,J)-AIK*DISPL(K,J)                            REAC0430
   20   CONTINUE                                                        REAC0440
        IF (PRINT)                                                      REAC0450
     &  CALL WMATRX(FORCE,NDOF,NLOAD,'MODIFIED LOADS')                  REAC0460
C                                                                       REAC0470
      ELSE IF (IOPT.EQ.2) THEN                                          REAC0480
C                                                                       REAC0490
C-------------------------                                              REAC0500
C- SOLVE FOR REACTIONS ---                                             REAC0510
C-------------------------                                              REAC0520
C CONSIDER THE PARTITIONED STIFFNESS WHERE X2 ARE THE RESTRAINT DISPL.  REAC0530
C                                                                       REAC0540
C   |P1|   | K11 | K12 | |X1|                                         REAC0550
C   |--| = |-----|-----| |--| * FACTOR                                  REAC0560
C   |P2|   | K21 | K22 | |X2|                                         REAC0570
C                                                                       REAC0580
C   THE REACTIONS ARE  P2 = P2 + ( K21*X1 + K22*X2 ) * FACTOR           REAC0590
C   NOTE. BOTH P1 AND P2 CONTAIN FIXED END FORCES FROM MEMBER LOADS..   REAC0600
C         THUS, K21*X1 + K22*X2 IS ADDED TO P2 TO GET THE REACTIONS     REAC0610
C-------------------------------------------------C                     REAC0620
C     DO 120 I=L3,L4                              C    FULL STORAGE     REAC0630
C     DO 120 J=1,NLOAD                            C    MODE EQUIVALENT  REAC0640
CC    FORCE(I,J)=0    * P2 CONTAINS PEF'''        C                     REAC0650
C     DO 110 K=L1,L2                              C                     REAC0660
C 110   FORCE(I,J)=FORCE(I,J)+K(I,K)*DISPL(K,J)   C                     REAC0670
C     DO 120 K=L3,L4                              C                     REAC0680
C 120   FORCE(I,J)=FORCE(I,J)+K(I,K)*DISPL(K,J)   C                     REAC0690
C-------------------------------------------------C                     REAC0700
C                                                                       REAC0710
      DO 120 I=L3,L4                                                    REAC0720
      DO 120 J=1,NLOAD                                                  REAC0730
      DO 110 K=L2,L1,-1                                                 REAC0740
        KI=MD(I)-I-K                                                    REAC0750
        IF (KI.GE.MD(I+1)) GO TO 120                                    REAC0760
        AKI=A(KI)                                                       REAC0770
  110   FORCE(I,J)=FORCE(I,J)+A(KI)*DISPL(K,J)*FACTOR                   REAC0780
  120 CONTINUE                                                          REAC0790
C                                                                       REAC0800
      DO 140 I=L3,L4                                                    REAC0810
      DO 140 K=L3,L4                                                    REAC0820
        IF (I.LE.K) THEN                                                REAC0830
          IK=MD(K)+K-I                                                  REAC0840
          MDK=MD(K+1)                                                   REAC0850
        ELSE                                                            REAC0860
          IK=MD(I)-I-K                                                  REAC0870
          MDK=MD(I+1)                                                   REAC0880
        ENDIF                                                           REAC0890
        IF (IK.GE.MDK)  GO TO 140                                       REAC0900
        AIK=A(IK)                                                       REAC0910
        DO 130 J=1,NLOAD                                                REAC0920
  130     FORCE(I,J)=FORCE(I,J)+A(IK)*DISPL(K,J)*FACTOR                 REAC0930
  140 CONTINUE                                                          REAC0940
C                                                                       REAC0950
      IF (PRINT)                                                        REAC0960
     & CALL WMATRX(FORCE,NDOF,NLOAD,'FINAL LOADS AND REACTIONS')        REAC0970
C                                                                       REAC0980
      ELSE IF (IOPT.EQ.3 .OR.  IOPT.EQ.4) THEN                          REAC0990
        NF=L2                                                           REAC1000
C-----------------------------                                          REAC1010
C- LOOP FOR EACH LOAD CASE  ---                                         REAC1020
C-----------------------------                                          REAC1030
      DO 300 L=1,NLOAD                                                  REAC1040
C                                                                       REAC1050
C-----------------------------                                          REAC1060
C- PRINT GLOBAL REACTIONS      ---                                      REAC1070
C-----------------------------                                          REAC1080
C                                                                       REAC1090
      IF (PRINT) THEN                                                   REAC1100
          IF (HEAD) WRITE(108,320) TITLE(1),TITLE(2)                    REAC1110
            NREACT=0                                                    REAC1120
            IF (IOPT.NE.4) WRITE(108,321) L,STEPID                      REAC1130
            IF (IOPT.EQ.4) WRITE(108,421)                               REAC1140
      ENDIF                                                             REAC1150
        IF (IOPT.EQ.3) THEN                                             REAC1160
          DO 190 I=1,6                                                  REAC1170
  190     SUM(I)=0                                                      REAC1180
        ENDIF                                                           REAC1190
C                                                                       REAC1200
      IF ( .NOT.(NCOS.EQ.1 .AND. COSINE(1,1,1).EQ.1 .AND.COSINE(2,2,1)  REAC1210
     &     .EQ.1 .AND  COSINE(3,3,1).EQ.1).AND. IOPT.EQ.4  ) GO TO 279  REAC1220
C                                                                       REAC1230
        DO 200 I=1,NNODE                                                REAC1240
C                                                                       REAC1250
C..... ..... GET REACTNS IN JOINT COORDINATE SYSTEM                     REAC1260
          FLAG= FALSE.                                                  REAC1270
          DO 209 J=1,6                                                  REAC1280
            IJ=IDOF(I,J)                                                REAC1290
            IF (IJ.LE.NF .OR. BTEST(JTFLG(I),J+11)) THEN                REAC1300
              F(J)=0                                                    REAC1310
            ELSE                                                        REAC1320
              F(J)=FORCE(IJ,L)                                          REAC1330
              FLAG=.TRUE.                                               REAC1340
            ENDIF                                                       REAC1350
  209     CONTINUE                                                      REAC1360
          IF (.NOT.FLAG) GO TO 200                                      REAC1370
C                                                                       REAC1380
C . .... ...  TRANSFER REACTNS TO GLOBAL COORDINATE SYSTEM              REAC1390
          DO 229 I1=1,3                                                 REAC1400
            I2=I1+3                                                     REAC1410
            G(I1)=0                                                     REAC1420
            G(I2)=0                                                     REAC1430
            DO 219 J1=1,3                                               REAC1440
              J2=J1+3                                                   REAC1450
              G(I1)=G(I1) + COSINE(J1,I1,JTCOS(I))*F(J1)                REAC1460
  219         G(I2)=G(I2) + COSINE(J1,I1,JTCOS(I))*F(J2)                REAC1470
            IF (PRINT) THEN                                             REAC1480
                WRITE(CD(I1),210) G(I1)                                 REAC1490
                WRITE(CD(I2),210) G(I2)                                 REAC1500
            ENDIF                                                       REAC1510
  229       CONTINUE                                                    REAC1520
C                                                                       REAC1530
C .     ... ... SUM REACTIONS      .....                                REAC1540
          IF (IOPT.EQ.3) THEN                                           REAC1550
            DO 240 I1=1,6                                               REAC1560
  240         SUM(I1)=SUM(I1) + G(I1)                                   REAC1570
            SUM(4)= SUM(4) - G(2)*COORD(I,3) + G(3)*COORD(I,2)          REAC1580
            SUM(5)= SUM(5) + G(1)*COORD(I,3) - G(3)*COORD(I,1)          REAC1590
            SUM(6)= SUM(6) - G(1)*COORD(I,2) + G(2)*COORD(I,1)          REAC1600
          ENDIF                                                         REAC1610
C                                                                       REAC1620
C     ...   ... PRINT GLOBAL REACTIONS       ...                      REAC1630
          NREACT=NREACT+1                                               REAC1640
          IF (PRINT) WRITE(108,220) ID(I),(CD(K),K=1,6)                 REAC1650
  200   CONTINUE                                                        REAC1660
        IF (IOPT.EQ.3 .AND. PRINT ) WRITE(108,230) SUM                  REAC1670
  279   IF (IOPT.EQ.4 .AND. PRINT ) WRITE(108,231) SUM                  REAC1680
C                                                                       REAC1690
C-----------------------------------------------                        REAC1700
C- PRINT LOCAL JOINT COORDINATE SYSTEMS REACTIONS  ---                  REAC1710
C-----------------------------------------------                        REAC1720
C                                                                       REAC1730
      IF (.NOT. PRINT ) GO TO 300                                       REAC1740
      IF (NCOS.EQ.1 .AND. COSINE(1,1,1) EQ.1 .AND  COSINE(2,2,1) EQ.1   REAC1750
     &                .AND. COSINE(3,3,1).EQ.1 ) GO TO 300             REAC1760
C                                                                       REAC1770
      IF (HEAD.AND  NREACT.GT.20) WRITE(108,320) TITLE(1),TITLE(2)      REAC1780
        IF (IOPT.NE.4) WRITE(108,322) L,STEPID                          REAC1790
        IF (IOPT.EQ.4) WRITE(108,422)                                   REAC1800
        DO 290 I=1,NNODE                                                REAC1810
C                                                                       REAC1820
C. .....  GET REACTNS IN JOINT COORDINATE SYSTEM                        REAC1830
          FLAG= FALSE                                                   REAC1840
          DO 280 J=1,6                                                  REAC1850
            IJ=IDOF(I,J)                                                REAC1860
            IF (IJ.LE.NF  OR. BTEST(JTFLG(I),J+11)) THEN                REAC1870
              CD(J)=' '                                                 REAC1880
            ELSE                                                        REAC1890
              WRITE(CD(J),210) FORCE(IJ,L)                              REAC1900
              FLAG=.TRUE.                                               REAC1910
            ENDIF                                                       REAC1920
  280     CONTINUE                                                      REAC1930
          IF (.NOT.FLAG) GO TO 290                                      REAC1940
C                                                                       REAC1950
C .... ... ... PRINT REACTNS IN LOCAL COORDINATE SYSTEM .... .          REAC1960
          WRITE(108,221) ID(I),(CD(K),K=1,6),JTCOS(I)                   REAC1970
  290   CONTINUE                                                        REAC1980
  300 CONTINUE                                                          REAC1990
C                                                                       REAC2000
  210 FORMAT (1F,G15.7)                                                 REAC2010
  220 FORMAT (5X,I5,6A)                                                 REAC2020
  221 FORMAT (5X,I5,6A,I10)                                             REAC2030
  230 FORMAT ('                ',90('-'),/' SUMMATION',6G15.7)          REAC2040
  231 FORMAT (' MAX OF ALL',89('-'),/' GCS SUMM.',6G15.7 )              REAC2050
  320 FORMAT ('1 STRUCTURE.....: ',A,/'  SOLUTION.....: ',A)            REAC2060
  321 FORMAT (//'  GCS RESTRAINT REACTIONS,    LOADING #',I5,5X,A,/     REAC2070
     &        ' ---------------------------//6X,'NODE',7X,'FX',        REAC2080
     &          13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ'/)        REAC2090
  322 FORMAT (//'  JCS RESTRAINT REACTIONS,    LOADING #',I5,5X,A,/     REAC2100
     &        ' ---------------------------//6X,'NODE',7X,'FX',        REAC2110
     &          13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ',         REAC2120
     &          6X,'  COSINE #'/)                                     REAC2130
  421 FORMAT (//'  MAXIMUM GCS RESTRAINT REACTIONS',                   REAC2140
     &       5X,'NOTE. MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'/   REAC2150
     &        ' ---------------------------//6X,'NODE',7X,'FX',        REAC2160
     &         13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ'/)         REAC2170
  422 FORMAT (//'  MAXIMUM JCS RESTRAINT REACTIONS ',                  REAC2180
     &       5X,'NOTE: MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'/   REAC2190
     &        ' ---------------------------//6X,'NODE',7X,'FX',        REAC2200
     &        13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ',           REAC2210
     &        6X,'  COSINE #'/)                                       REAC2220
C                                                                       REAC2230
       ELSE                                                             REAC2240
        WRITE(108,*) 'INVALID OPTION IN SUBROUTINE  REA_CTN, IOPT=',IOPT REAC2250
       ENDIF                                                            REAC2260
      RETURN                                                            REAC2270
      END                                                               REAC2280
C                                                                       PORM0010
C-----------------------------------------------------------------------PORM0020
C                                                                       PORM0030
      SUBROUTINE PORM(IOPT)                                             PORM0040
      CHARACTER  NAME*80,ELNO*130                                       PORM0050
      LOGICAL PRINT,HEAD,FALSE,AXIAL,BTEST,BUG5,BUG57                   PORM0060
$INCLUDE:'ZCOMN'                                                        PORM0070
C                                                                       PORM0080
C-------------------------------------------------------------- ------PORM0090
C  MATRIX STORAGE SCHEME.                                               PORM0100
C -THE STIFFNESS MATRIX (K) IS BANDED AND SYMMETRIC                     PORM0110
C  THUS ONLY THE SKYLINE ABOVE THE MAIN DIAGONAL NEEDS TO BE STORED.    PORM0120
C  THE MATRIX K IS STORED IN A LINEAR ARRAY BY COLUMNS.                 PORM0130
C  THE VALUE ON THE MAIN DIAGONAL IS STORED IN THE LINEAR ARRAY FIRST.  PORM0140
C  RELATIVE ADDRESSES OF THE LINEAR ARRAY ELEMENTS CONTAINING THE       PORM0150
C  MAIN DIAGONAL TERMS ARE STORED IN MD.                                PORM0160
C  THE ABSOLUTE ADDRESS OF THE FIRST ELEMENT IN THE K MATRIX IS IZSTIF  PORM0170
C  THE STIFFNESS MATRIX K IS STORED IN THE Z ARRAY, Z(IZSTIP)=K(1,1)    PORM0180
C  THE ABSOLUTE ADDRESS OF THE FIRST ELEMENT IS THE MD MATRIX IS IZMD   PORM0190
C  THE STIFFNESS MATRIX K IS STORED IN THE NZ ARRAY, NZ(IZMD)=MD(1)     PORM0200
C  THUS. IF I LE J (UPPER TRIANGULAR MATRIX)                            PORM0210
C        IF IZMD-1+J < IZMD-1+J-1                                       PORM0220
C              IJ=NZ(IZMD-1+J)+J-I : RELATIVE ADDRESS IP ELEMENT (I,J)  PORM0230
C              B(I,J)=Z(IZSTIP-1 + IJ)                                  PORM0240
C         ELSE                                                          PORM0250
C              B(I,J)=0             : (I,J) IS ABOVE THE SKYLINE        PORM0260
C                                                                       PORM0270
C  EXAMPLE: LET B BE THE 5X5 FULL SYMETRIC MATRIX BELOW..               PORM0280
C     | 1 | 3 |   |  12 |                                              PORM0290
C     |   | 2 | 5 |   11 |   NOTE: B(1,3),B(1,4) AND B(2,4) ARE ZERO.  PORM0300
C  B= |   | 4 | 7 |  10 |                                              PORM0310
C     |   | SYMM. | 6 |  9 |  MD= 1, 2, 4, 6, 8, 13                   PORM0320
C     |   |   |   |  8 |                                              PORM0330
C  SAY IZMD-121,IZSTIP-200                                              PORM0340
C  NZ(121)-1          Z(200)-1                                          PORM0350
C  NZ(122)-2          Z(201)-3                                          PORM0360
C  NZ(123)-4          Z(202)-2                                          PORM0370
C  NZ(124)-6          Z(203)-4                                          PORM0380
C  NZ(125)-8          Z(204)-5                                          PORM0390
C  NZ(126)-13         Z(205)-6                                          PORM0400
C                     Z(206)-7                                          PORM0410
C                     Z(206)-8                                          PORM0420
C                     Z(206)-9                                          PORM0430
C                     Z(206)-10                                         PORM0440
C                     Z(206)-11                                         PORM0450
C                     Z(206)-12                                         PORM0460
C  THUS, FOR B(3,4)  I-3, J-4                                           PORM0470
C              IJ-NZ(IZMD-1+J)+J-I = NZ(121-1+4) +4-3 = 6+4-3 -7        PORM0480
C              B(I,J)-Z(IZSTIP-1 + IJ) - Z(200-1+7) = Z(206) - 7  OK    PORM0490
C  THUS, FOR B(2,4)  I-2,  J-4                                          PORM0500
C              IJ-NZ(IZMD-1+J)+J-I = NZ(121-1+4) +4-2 = 6+4-2 =8        PORM0510
C              NZ(IZMD -1+J-1)=NZ(121-1+4+1)=8                          PORM0520
C              SINCE IJ GE NZ(IZMD-1+J-1)                               PORM0530
C              B(I,J)=0  (I,J) IS ABOVE THE SKYLINE.                    PORM0540
C-----------------------------------------------------------------------PORM0550
C                                                                       PORM0560
      BUG5 = BTEST(IBUG,5)                                              PORM0570
      BUG57= BTEST(IBUG,7) .AND. BUG5                                   PORM0580
      LSTTYP=0                                                          PORM0590
      PRINT=.FALSE.                                                     PORM0600
      HEAD =.FALSE                                                      PORM0610
C                                                                       PORM0620
C-------------------------------------------------------------------    PORM0630
C IOPT=1 FORM STIFFNESS MATRIX                                          PORM0640
C IOPT=2 FORM GEOMETRIC STIFFNESS MATRIX                                PORM0650
C-------------------------------------------------------- ------------PORM0660
C                                                                       PORM0670
       IF (BUG5)  WRITE (6,*) ' IN ROUTINE FORM IOPT=',IOPT             PORM0680
C                                                                       PORM0690
  49 FORMAT (1X,A/('  I=',I3,'  MDMASS(I)=',I11))                       PORM0700
C                                                                       PORM0710
      GO TO (100,200) IOPT                                              PORM0720
C                                                                       PORM0730
C--------------------------                                             PORM0740
C- ASSEMBLE STIFFNESS  -----                                            PORM0750
C--------------------------                                             PORM0760
  100 CONTINUE                                                          PORM0770
          IST =IZSTIP                                                   PORM0780
          IEND=IZSTIP +NZ(IZMD+NDOF) -1                                 PORM0790
C                                                                       PORM0800
C---------- ZERO STIFFNESS                                              PORM0810
      DO 110 I=IST,IEND                                                 PORM0820
  110   Z(I)=0                                                          PORM0830
C                                                                       PORM0840
C----------- LOOP FOR EACH ELEMENT                                      PORM0850
      DO 120 IELNO=1,NELMT                                              PORM0860
C                                                                       PORM0870
C---------- GET STIFFNESS OF EACH ELEMENT                               PORM0880
        FALSE= FALSE.                                                   PORM0890
        CALL ELELIB                                                     PORM0900
     &   (3,LSTTYP,FALSE,IREL,FALSE,NAME,EPSE,EPSE,DAMAGE,DUCFAG,      PORM0910
     &    IELNO ,IELDOF,XGDOF,PGEOM,Z   ,AXIAL,IZDISP,IZLOAD,MSTOR)    PORM0920
C                                                                       PORM0930
        IF (BUG57) THEN                                                 PORM0940
          WRITE (ELNO,15)  IELNO,(NZ(I+IZLM-1),I=1,MIN(IELDOF,18))      PORM0950
   15     FORMAT ('ELEMENT #',I6,' LM ',1816)                           PORM0960
          CALL LMATRX(Z(IZKE),MD,IELDOF,MD(IELDOF+1),ELNO)              PORM0970
        ENDIF                                                           PORM0980
C                                                                       PORM0990
      CALL PORML(Z(IZSTIP),NZ(IZMD),NDOF,NZ(IZMD+NDOF),                 PORM1000
```

```
     &           Z(IZKE),MD,IELDOF,MD(IELDOF+1),NZ(IZLM),1.)                FORM1010
C                                                                           FORM1020
C---------- SUBTRACT GEOMETRIC STIFFNESS                                    FORM1030
      NZ(IZKGDT+3)=NZ(IZKGDT+3)                                             FORM1040
C                                                                           FORM1050
      IF (NZ(IZKGDT+3).EQ.1) THEN                                           FORM1060
        FALSE=.FALSE.                                                       FORM1070
        CALL ELELIB                                                         FORM1080
     &  (2,LSTTYP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DAMAGE,DUCFAG,            FORM1090
     &   IELNO ,IELDOF,KGDOF,PGEOM,2   ,AXIAL,IZDISP,IZLOAD,MSTOR)          FORM1100
C                                                                           FORM1110
        KGDOF=KGDOF                                                         FORM1120
C                                                                           FORM1130
        IF (BUGS7) THEN                                                     FORM1140
          WRITE (ELNO,15)  IELNO,(NZ(I+IZLMKG-1),I=1,MIN(KGDOF,18))        FORM1150
          CALL LMATRX(Z(IZKEKG),MD,KGDOF,MD(KGDOF+1),ELNO)                  FORM1160
        ENDIF                                                               FORM1170
C                                                                           FORM1180
        IF (.NOT. AXIAL) GO TO 120                                          FORM1190
C           DETERMINE THE FACTOR FOR KG                                     FORM1200
              IF (NZ(IZKGDT+1).EQ.1) THEN                                   FORM1210
                    ACCEL2= Z(IZKGDT)                                       FORM1220
                    PGEOM=PGEOM*(1+ACCEL2)                                  FORM1230
                ENDIF                                                       FORM1240
C     SWAP THE SIGN ON PGEOM FOR SUBTRACTION. ...                           FORM1250
      PGEOM=-PGEOM                                                          FORM1260
C                                                                           FORM1270
      CALL FORML(Z(IZSTIF),NZ(IZMD),NDOF,NZ(IZMD+NDOF),                    FORM1280
     &     Z(IZKEKG),MD,KGDOF,MD(KGDOF+1),NZ(IZLMXG),PGEOM)                FORM1290
      ENDIF                                                                 FORM1300
C                                                                           FORM1310
  120   CONTINUE                                                            FORM1320
C                                                                           FORM1330
      RETURN                                                                FORM1340
C                                                                           FORM1350
C---------- ASSEMBLE GEOMETRIC STIFFNESS                                    FORM1360
  200 CONTINUE                                                              FORM1370
      IF (NZ(IZKGDT+3).NE.2 ) RETURN                                        FORM1380
        IST =IZKG                                                           FORM1390
        IEND=IZKG + NZ(IZMDKG+NDOF) - 1                                     FORM1400
C                                                                           FORM1410
C---------- ZERO STIFFNESS                                                  FORM1420
      DO 210 I=IST,IEND                                                     FORM1430
  210   Z(I)=0                                                              FORM1440
C                                                                           FORM1450
C----------  LOOP FOR EACH ELEMENT                                          FORM1460
      DO 220 IELNO=1,NELMT                                                  FORM1470
C                                                                           FORM1480
C--------  GET STIFFNESS OF EACH ELEMENT                                    FORM1490
      FALSE=.FALSE                                                          FORM1500
      CALL ELELIB                                                           FORM1510
     &  (2,LSTTYP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DAMAGE,DUCFAG,            FORM1520
     &  IELNO ,IELDOF,KGDOF,PGEOM,2   ,AXIAL,IZDISP,IZLOAD,MSTOR)          FORM1530
C                                                                           FORM1540
      IF (.NOT. AXIAL) GO TO 220                                            FORM1550
C                                                                           FORM1560
      IF (BUGS7) THEN                                                       FORM1570
        WRITE (6,*)  'PGEOM=',PGEOM                                         FORM1580
        WRITE (ELNO,15)  IELNO,(NZ(I+IZLMKG-1),I=1,MIN(KGDOF,18))          FORM1590
        CALL LMATRX(Z(IZKEKG),MD,KGDOF,MD(KGDOF+1),ELNO)                    FORM1600
      ENDIF                                                                 FORM1610
C                                                                           FORM1620
      CALL FORML(Z(IZKG),NZ(IZMDKG),NDOF,NZ(IZMDKG+NDOF),                   FORM1630
     &     Z(IZKEKG),MD,KGDOF,MD(KGDOF+1),NZ(IZLMXG),PGEOM)                FORM1640
  220 CONTINUE                                                              FORM1650
C                                                                           FORM1660
      RETURN                                                                FORM1670
      END                                                                   FORM1680
C*************************************************************************CFEM40010
C                                                                           CFEM40020
C   PROGRAM FEM - ANALYSIS OF STRUCTURES                                    CFEM40030
C                                                                           CFEM40040
C*************************************************************************CFEM40050
C                                                                           CFEM40060
C   NOTE. USE IBM FORTVS2 COMPILER OPTION AUTODBL(DBLPAD4)                  CFEM40070
C         TO COMPILE THIS PROGRAM IN DOUBLE PRECISION                       CFEM40080
C                                                                           CFEM40090
C   NOTE: USE IBM FORTVS2 COMPILER OPTION OPT(2)                            CFEM40100
C         TO OPTIMIZE THE CODE FOR FASTER EXECUTION                         CFEM40110
C                                                                           CFEM40120
C*************************************************************************CFEM40130
      CHARACTER*80 OPTION,INPUT,BLANK                                       FEM40140
      CHARACTER*8  TIME1,DATE1                                              FEM40150
      CHARACTER*1  CHR(0:25)                                                FEM40160
      LOGICAL TEST,HEAD,AXIAL                                               FEM40170
$INCLUDE:'ZCOMX'                                                            FEM40180
      DATA CHR/'A','B','C','D','E','F','G','H','I','J','K','L','M',        FEM40190
     &         'N','O','P','Q','R','S','T','U','V','W','X','Y','Z'/        FEM40200
C                                                                           FEM40210
C---------------------------                                                FEM40220
C-- ECHO THE INPUT.....     --                                              FEM40230
C---------------------------                                                FEM40240
      I=55                                                                  FEM40250
      L=0                                                                   FEM40260
9995  READ(105,('(A)'),END=9996) INPUT                                      FEM40270
      IF (INDEX(INPUT,'NOECHO').NE.0) GO TO 9996                            FEM40280
      I=I+1                                                                 FEM40290
      L=L+1                                                                 FEM40300
      IF ( I.GT.55 ) THEN                                                   FEM40310
         WRITE(108,9994) (KK,KK=1,8)                                        FEM40320
         I=0                                                                FEM40330
      ENDIF                                                                 FEM40340
      WRITE(108,9997) L,INPUT                                               FEM40350
      IF (INDEX(INPUT,'STOP'  ).EQ.0) GO TO 9995                            FEM40360
9994  FORMAT ('1    ECHO OF INPUT DATA '//                                  FEM40370
     &        1X,3X,'LINE  ',8('....:  ..'),I1,'0'))                        FEM40380
9997  FORMAT (1X,I7,': ',A)                                                 FEM40390
9996  REWIND(105)                                                           FEM40400
C                                                                           FEM40410
C---------------------------                                                FEM40420
C-- INITIALIZE DATA .  .    --                                              FEM40430
C---------------------------                                                FEM40440
      TITLE(1)=' '                                                          FEM40450
      TITLE(2)=' '                                                          FEM40460
      BLANK  =' '                                                           FEM40470
      BUG = FALSE                                                           FEM40480
      DOUBLE =.FALSE.                                                       FEM40490
      IBUG=0                                                                FEM40500
      MAXEDF=24                                                             FEM40510
      MD(1)=1                                                               FEM40520
      DO 1 I=2,(MAXEDF+1)                                                   FEM40530
    1   MD(I)=MD(I-1)+(I-1)                                                 FEM40540
        IZ=0                                                                FEM40550
C                                                                           FEM40560
   20 READ(105,*,END=1000) OPTION                                           FEM40570
      RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                        FEM40580
C                                                                           FEM40590
C---------------------------                                                FEM40600
C-- DEFINE THE STRUCTURE ---                                                FEM40610
C---------------------------                                                FEM40620
      IF (INDEX(OPTION,'STR').NE.0)  THEN                                   FEM40630
C ---------------  ---------------                                          FEM40640
            IZ=1                                                            FEM40650
            TITLE(1)=' '                                                    FEM40660
            TITLE(2)=' '                                                    FEM40670
C ---------  ---  ------------------                                        FEM40680
            GRAV=386.4                                                      FEM40690
            READ(105,*,END=1000) TITLE(1)                                   FEM40700
            CALL TIDAB(TIME1,DATE1)                                         FEM40710
            WRITE(TITLE(1)(60:),45) TIME1,DATE1                             FEM40720
            RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                  FEM40730
            WRITE(*,52)' BULIDING STRUCTURAL MODEL ----------------*',      FEM40740
     &                 IZ,RATIZ                                             FEM40750
            WRITE(*,51) TITLE(1)(1:70)                                      FEM40760
            CALL STRUCT                                                     FEM40770
            IZREL=IZ                                                        FEM40780
            RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                  FEM40790
C                                                                           FEM40800
C----------------------------------                                         FEM40810
C-- DEFINE THE LOADING AND SOLVE --                                         FEM40820
C----------------------------------                                         FEM40830
      ELSE IF (INDEX(OPTION,'SOL').NE.0)  THEN                              FEM40840
            READ(105,*,END=1000) TITLE(2)                                   FEM40850
            CALL TIDAB(TIME1,DATE1)                                         FEM40860
            WRITE(TITLE(2)(80:),45) TIME1,DATE1                             FEM40870
            WRITE(*,52)' SOLVING STRUCTURAL MODEL ------------------*',     FEM40880
     &                 IZ,RATIZ                                             FEM40890
            WRITE(*,51) TITLE(1)(1:70)                                      FEM40900
            WRITE(*,51) TITLE(2)(1:70)                                      FEM40910
            CALL SOLN(OPTION)                                               FEM40920
            RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                  FEM40930
C                                                                           FEM40940
C----------------------------------                                         FEM40950
C-- STOP... END OF PROGRAM       --                                         FEM40960
C----------------------------------                                         FEM40970
      ELSE IF (INDEX(OPTION,'STOP').NE.0)  THEN                             FEM40980
            WRITE(*,52)' NORMAL STOP ------------------- ------- --*',     FEM40990
     &                 IZ,RATIZ                                             FEM41000
            STOP                                                            FEM41010
C                                                                           FEM41020
C----------------------------------                                         FEM41030
C--  READ RESULTS FROM INPUT FILE ---                                       FEM41040
C----------------------------------                                         FEM41050
      ELSE IF (INDEX(OPTION,'READ').NE.0)  THEN                             FEM41060
  140       CALL GETINT(OPTION,'UNIT=',IUNIT,0,.TRUE.,.TRUE. )              FEM41070
            CALL GETINT(OPTION, 'INC=',INC  ,1, TRUE.,.FALSE.)              FEM41080
            IF (IUNIT GT.0) THEN                                            FEM41090
              WRITE(*,52) ' READING DATA FROM FILE ------------------*',    FEM41100
     &                    IZ,RATIZ                                          FEM41110
              IOPT=4                                                        FEM41120
              CALL DMPDAT(IOPT,IWRITE,TO,DT)                                FEM41130
              GO TO 140                                                     FEM41140
            ENDIF                                                           FEM41150
            RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                  FEM41160
C                                                                           FEM41170
C------------------------                                                   FEM41180
C-- SET BUG OPTION     --                                                   FEM41190
C------------------------                                                   FEM41200
      ELSE IF (TEST(OPTION,'BUG=', FALSE.))  THEN                           FEM41210
            CALL GETCHR(OPTION,'BUG=',INPUT,.TRUE.,.FALSE.)                 FEM41220
            IBUG=0                                                          FEM41230
            WRITE(OPTION,160) INPUT(1:70)                                   FEM41240
            DO 150 I=0,23                                                   FEM41250
              IF (TEST(INPUT,CHR(I),.FALSE.)) IBUG=IBSET(IBUG,I)            FEM41260
  150       CONTINUE                                                        FEM41270
  160       FORMAT ('BUG=',A)                                               FEM41280
            WRITE(*,52)' SET BUG OPTION- ------------------------------*',  FEM41290
     &                 IZ,RATIZ                                             FEM41300
            WRITE(*,51)  OPTION(1:70)                                       FEM41310
            WRITE(108,51)  'SET BUG OPTION. '                               FEM41320
            WRITE(108,51)  OPTION(1:70)                                     FEM41330
C                                                                           FEM41340
C------------------------                                                   FEM41350
C-- SET BUG OPTION     --                                                   FEM41360
C------------------------                                                   FEM41370
      ELSE IF (INDEX(OPTION,'BUG').NE.0)  THEN                              FEM41380
            WRITE(*,51) ('SET BUG OPTION  '//OPTION(1:54) )                 FEM41390
            BUG=.TRUE.                                                      FEM41400
            IF (INDEX(OPTION,'NOBUG').NE.0) BUG= FALSE.                     FEM41410
C                                                                           FEM41420
C-------------------------------------                                      FEM41430
C-- RELEASE MEMORY FROM LAST SOLUTION --                                    FEM41440
C-------------------------------------                                      FEM41450
      ELSE IF (INDEX(OPTION,'RELEASE').NE.0)  THEN                          FEM41460
           WRITE(108,52)'------------ -- ------- ----------------------*'    FEM41470
           WRITE(*,52)' RELEASING MEMORY ------ ---------------------*'     FEM41480
           WRITE(108,52)' RELEASING MEMORY  ------ ---------------------*'  FEM41490
           IF (INDEX(OPTION,'ELEMENT').NE.0)  THEN                          FEM41500
           WRITE(108,52)' RESETTING ELEMENT FORCES ------------------*'     FEM41510
           WRITE(108,52)'------------------------------ -- -------- --*'    FEM41520
             DO 170 IELNO=1,NELMT                                           FEM41530
               LSTTYP=0                                                     FEM41540
               HEAD =.FALSE                                                 FEM41550
  170          CALL ELELIB                                                 FEM41560
     &       (13,LSTTYP,.FALSE.,IREL,HEAD,INPUT,EESE,EPSE,DAMAGE,DUCFAG,    FEM41570
     &        IELNO,IELDOF,KGDOF,PGEOM,2 ,AXIAL,IZDDSP,IZDLOA,MSTOR)       FEM41580
           ELSE                                                             FEM41590
           WRITE(108,52)'------------- ------------------------ *'          FEM41600
           ENDIF                                                            FEM41610
           IZ=IZREL                                                         FEM41620
             IZACC =0                                                       FEM41630
             IZDACC=0                                                       FEM41640
             IZDDSP=0                                                       FEM41650
             IZDISP=0                                                       FEM41660
             IZDLOA=0                                                       FEM41670
             IZDVEL=0                                                       FEM41680
             IZFUB =0                                                       FEM41690
             IZLOAD=0                                                       FEM41700
             IZVEL =0                                                       FEM41710
             RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                 FEM41720
C                                                                           FEM41730
C-------------------------------------                                      FEM41740
C-- SAVE MEMORY FOR A RESTART       --                                      FEM41750
C-------------------------------------                                      FEM41760
      ELSE IF (INDEX(OPTION,'SAVE').NE.0)  THEN                             FEM41770
            WRITE(*,52)' SAVING DATA  ---- ------------------------ --*',   FEM41780
     &                 IZ,RATIZ                                             FEM41790
            WRITE(*,51) TITLE(1)(1:70)                                      FEM41800
            WRITE(*,51) TITLE(2)(1:70)                                      FEM41810
            WRITE(108,200)                                                  FEM41820
  200       FORMAT(5X,'******** WRITING DATA TO UNIT FT09 ********')        FEM41830
            OPEN(UNIT=9,ACCESS='SEQUENTIAL',STATUS='UNKNOWN',               FEM41840
     &           FORM='UNFORMATTED')                                        FEM41850
            WRITE(9)   TITLE                                                FEM41860
            WRITE(9)   IZ     ,IZREL ,                                      FEM41870
     &        NDOF  ,NCOND ,NFREE ,NREST ,MAXNOD,NMAT  ,NELMT ,NNODE ,     FEM41880
     &        NLOAD ,NSOLN ,MAXEDF,MD,NCOS,MAXELD,NELD,                     FEM41890
     &        IZID  ,IZIDOF,IZCORD,IZCOS ,IZCNST,IZMAT ,IZELE ,             FEM41900
     &        IZJFLG,IZJCOS,                                                FEM41910
     &        IZXE  ,IZLM  ,IZSTIF,IZMD  ,IZLOAD,IZDISP,IZIELD,IZELD        FEM41920
            DO 210 J=1,IZ,256                                               FEM41930
              K=J+255                                                       FEM41940
              K=MIN(K,IZ)                                                   FEM41950
  210         WRITE(9) (Z(I),I=J,K)                                         FEM41960
            CLOSE (UNIT=9)                                                  FEM41970
            RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                  FEM41980
C                                                                           FEM41990
C----------------------------------                                         FEM42000
C-- RESTART PROGRAM              --                                         FEM42010
C----------------------------------                                         FEM42020
            WRITE(*,52)' RETREVING DATA ------ ----------------------*',    FEM42030
     &                 IZ,RATIZ                                             FEM42040
      ELSE IF (INDEX(OPTION,'RESTART').NE.0)  THEN                          FEM42050
            WRITE(108,300)                                                  FEM42060
  300       FORMAT(5X,'******** READING DATA FROM UNIT FT09 ********')      FEM42070
            OPEN(UNIT=9,ACCESS='SEQUENTIAL',STATUS='OLD',                   FEM42080
     &           FORM='UNFORMATTED')                                        FEM42090
            READ  (9)   TITLE                                               FEM42100
            READ  (9)   IZ     ,IZREL ,                                     FEM42110
     &        NDOF  ,NCOND ,NFREE ,NREST ,MAXNOD,NMAT  ,NELMT ,NNODE ,     FEM42120
     &        NLOAD ,NSOLN ,MAXEDF,MD,NCOS,MAXELD,NELD,                     FEM42130
     &        IZID  ,IZIDOF,IZCORD,IZCOS ,IZCNST,IZMAT ,IZELE ,             FEM42140
     &        IZJFLG,IZJCOS,                                                FEM42150
     &        IZXE  ,IZLM  ,IZSTIF,IZMD  ,IZLOAD,IZDISP,IZIELD,IZELD        FEM42160
            DO 310 J=1,IZ,256                                               FEM42170
              K=J+255                                                       FEM42180
              K=MIN(K,IZ)                                                   FEM42190
  310         READ  (9) (Z(I),I=J,K)                                        FEM42200
            CLOSE (UNIT=9)                                                  FEM42210
            WRITE(*,52)' DATA RETREIVED ------------ --- ------------*',    FEM42220
     &                 IZ,RATIZ                                             FEM42230
            WRITE(*,51) TITLE(1)(1:70)                                      FEM42240
```

```
          WRITE(*,51) TITLE(2)(1:70)                                    FEM42250
          RATIZ=100 *REAL(IZ)/REAL(MAXZ)                                FEM42260
C                                                                       FEM42270
C*-------------------------------------                                FEM42280
C-- INVALID OPTION                     --                               FEM42290
C*-------------------------------------                                FEM42300
       ELSE                                                             FEM42310
          IF (TEST(OPTION,'NOECHO',.FALSE.) ) GO TO 20                  FEM42320
          IF (OPTION.EQ.BLANK) GO TO 20                                 FEM42330
          WRITE(108,30) OPTION                                          FEM42340
          WRITE(*,51) ('INVALID OPTION: '//OPTION(1:54))                FEM42350
       ENDIF                                                            FEM42360
       GO TO  20                                                        FEM42370
C                                                                       FEM42380
   45 FORMAT(' TIME. ',A,', DATE: ',A)                                  FEM42390
   50 FORMAT (A1,' STRUCTURE. . : ',A,/1X,' SOLUTION.. ..: ',A,//)      FEM42400
   51 FORMAT (1X,A)                                                     FEM42410
   52 FORMAT (1X,'*---',A65,'*',;,' IZ=',I7,' MEM=',F6.3,'%')           FEM42420
   30 FORMAT ('1 INVALID OPTION= ',A)                                   FEM42430
  110 FORMAT (' ZMATRX(',16,').',I15,1P,G20.10,',G20.10)                FEM42440
 1000 STOP                                                              FEM42450
       END                                                              FEM42460
C                                                                       TIDA0010
C ----- TAKE PRESENT TIME AND DATE ----                                 TIDA0020
C                                                                       TIDA0030
       SUBROUTINE TIDAE(TIME1,DATE1)                                    TIDA0040
        INTEGER*2 IYR,IMON,IDAY,IHR,IMIN,ISEC,I100TH                    TIDA0050
        CHARACTER*8 TIME1,DATE1                                         TIDA0060
        CALL GETDAT(IYR,IMON,IDAY)                                      TIDA0070
        IYR=IYR-1900                                                    TIDA0080
        CALL GETTIM(IHR,IMIN,ISEC,I100TH)                               TIDA0090
        WRITE(DATE1,'(i2,1h/,i2,1h/,i2)') IMON,IDAY,IYR                 TIDA0100
        WRITE(TIME1,'(i2,1h:,i2,1h:,i2)') IHR,IMIN,ISEC                 TIDA0110
       RETURN                                                           TIDA0120
       END                                                              TIDA0130
C                                                                       SOLN0010
C*--------------------------------------------------------------------*SOLN0020
C                                                                       SOLN0030
      SUBROUTINE SOLN(OPTION)                                           SOLN0040
      CHARACTER*80 OPTION                                               SOLN0050
$INCLUDE:'ZCOMM'                                                        SOLN0060
      STEPID=' '                                                        SOLN0070
C                                                                       SOLN0080
C-------------------------                                             SOLN0090
C- DETERMINE SOLUTION NO. --                                            SOLN0100
C-------------------------                                             SOLN0110
      CALL GETINT(OPTION,'SOL',NSOLN,0,.TRUE.,.FALSE.)                  SOLN0120
C                                                                       SOLN0130
C-----------------------                                               SOLN0140
C- CHOOSE THE SOLUTION --                                               SOLN0150
C-----------------------                                               SOLN0160
C     IF (NSOLN.EQ.1) THEN                                              SOLN0170
C           CALL SOL01                                                  SOLN0180
C     ELSE IF (NSOLN.EQ.2) THEN                                         SOLN0190
C           NLOAD=1                                                     SOLN0200
C           MAXELD=0                                                    SOLN0210
C           CALL SOL02                                                  SOLN0220
C     ELSE IF (NSOLN.EQ 3) THEN                                         SOLN0230
C           NLOAD=1                                                     SOLN0240
C           MAXELD=0                                                    SOLN0250
C           CALL SOL03                                                  SOLN0260
C     ELSE IF (NSOLN.EQ.4) THEN                                         SOLN0270
      IF (NSOLN.EQ.4) THEN                                              SOLN0280
           NLOAD=1                                                      SOLN0290
           CALL SOL04                                                   SOLN0300
C     ELSE IF (NSOLN.EQ.5) THEN                                         SOLN0310
C           NLOAD=1                                                     SOLN0320
C           MAXELD=0                                                    SOLN0330
C           CALL SOL05                                                  SOLN0340
      ELSE                                                              SOLN0350
           WRITE(108,*) 'INVALID SOLN#',NSOLN                           SOLN0360
           STOP 'INVALID SOLUTION NUMBER'                               SOLN0370
      ENDIF                                                             SOLN0380
      RETURN                                                            SOLN0390
      END                                                               SOLN0400
C                                                                       MSOL0010
C*--------------------------------------------------------------------*MSOL0020
C  MATRIX SOLUTION OF EQUATION AX=P, WHERE THE UPPER TRIANGULAR MATRIX  MSOL0030
C  OF THE SYMMETRIC MATRIX A IS STORED BY SKYLINES. P IS STORED AS A    MSOL0040
C  FULL MATRIX.  THE MATRIX P IS REPLACED BY X AFTER BACK SUBSTITUTION  MSOL0050
C  IS COMPLEATE.  BOTH A AND P ARE ALTERED ....                         MSOL0060
C  GUYAN OPTION: IF GUYAN IS .TRUE. MASS MATRIX IS ALSO CONDENSED       MSOL0070
C*--------------------------------------------------------------------*MSOL0080
      SUBROUTINE MSOLL(IOPT,PRINT,N,IMD,LROW,NLC,L1,L2,MD,A,P,FACTOR,   MSOL0090
     &                 GUYAN,MASS,MDMASS,IMDMAS,KGCOND,KG,MDKG,IMDKG)  MSOL0100
      REAL   MASS(IMDMAS),KG(IMDKG)                                     MSOL0110
      LOGICAL GUYAN,PRINT,KGCOND                                        MSOL0120
      DIMENSION A(IMD),FACTOR(N),P(LROW,NLC),MD(N+1)                    MSOL0130
      DIMENSION MDMASS(N+1),MDKG(N+1)                                   MSOL0140
C                                                                       MSOL0150
      IJI(I,J)=MD(J)+J-I                                                MSOL0160
      IJM(I,J)=MDMASS(J)+J-I                                            MSOL0170
      IJKG(I,J)=MDKG(J)+J-I                                             MSOL0180
C-- ---------- ---------- ---------- ---------- ---------- ----------  MSOL0190
C   MATRIX STORAGE SCHEME.                                              MSOL0200
C  -THE MATRIX A IS BANDED AND SYMMETRIC  THUS ONLY THE SKYLINE ABOVE   MSOL0210
C   AND THE MAIN DIAGONAL NEEDS TO BE STORED.                          MSOL0220
C   THE MATRIX A IS STORED IN A LINEAR ARRAY BY COLUMNS   THE VALUE ON  MSOL0230
C   THE MAIN DIAGONAL IS STORED IN THE LINEAR ARRAY FIRST.              MSOL0240
C   ADDRESSES OF THE LINEAR ARRAY ELEMENTS CONTAINING THE MAIN DIAGONAL MSOL0250
C   TERMS ARE STORED IN MD.                                             MSOL0260
C   EXAMPLE: LET B BE THE 5X5 FULL SYMETRIC MATRIX BELOW. .  .          MSOL0270
C       |  1  |  3  |  7  |  12 |                                       MSOL0280
C       |     |  2  |  5  |  11 |  NOTE: B(1,3),B(1,4) AND B(2,4) ARE 2ERO. MSOL0290
C   B-  |     |     |  4  |  10 |                                       MSOL0300
C       |     SYMM.       |  6  |  9  |  MD= 1, 2, 4, 6, 8, 13        MSOL0310
C       |     |     |     |     |  8  |                                 MSOL0320
C   THUS, B(3,3)=A( MD(3) )= A(4)                                       MSOL0330
C         B(1,J)=A( MD(J)+J-I )  IF J>I AND (MD(J)+J-I)<MD(J-1)         MSOL0340
C               =0               IF J>I AND (MD(J)+J-I)>MD(J+1)       MSOL0350
C  -MATRIX P IS NOT SYMETRIC, AND IS STORED AS A FULL MATRIX            MSOL0360
C---------- ---------- ---------- ---------- ---------- ----------     MSOL0370
C   VARIABLE TABLE.                                                     MSOL0380
C   IOPT  - OPTION NUMBER                                               MSOL0390
C   N     - NUMBER OF DEGREES OF FREEDOM OF A MATRIX                    MSOL0400
C   NLC   - NUMBER OF LOAD CASES FOR THE P MATRIX                       MSOL0410
C   L1    - FIRST ROW TO BE WORKED WITH                                 MSOL0420
C   L2    - LAST ROW TO BE WORKED WITH                                  MSOL0430
C   MD    - ADDRESS OF THE MAIN DIAGONAL TERMS OF A                     MSOL0440
C   P     - LAOD MATRIX ON INPUT, SOLN (X) OF AX=P ON COMPLEATION OF    MSOL0450
C           IOPT=6                                                      MSOL0460
C   FACTOR- TEMPORARY STORAGE MATRIX                                    MSOL0470
C   GUYAN - GUYAN REDUCTION FLAG FOR MASS MATRIX                        MSOL0480
C   MASS  - MASS MATRIX                                                 MSOL0490
C   MDMASS- MASS MATRIX ADDRESSES OF MAIN DIAGONAL ELEMENTS             MSOL0500
C   KGCOND- FLAG FOR CONDENSING OUT KG                                  MSOL0510
C   KG    - GEOMETRIC STIFFNESS MATRIX                                  MSOL0520
C   MDKG  - KG MATRIX ADDRESSES OF MAIN DIAGONAL ELEMENTS               MSOL0530
C   L     - ROW NUMBER FOR ELIMINATION                                  MSOL0540
C   I     - ROW NUMBER                                                  MSOL0550
C   J     - COLUMN NUMBER                                               MSOL0560
C-- - ---------- ------- -------- - ---------- ----- -------            MSOL0570
C                                                                       MSOL0580
      L1=MAX(L1,1)                                                      MSOL0590
      L2=MIN(L2,N)                                                      MSOL0600
C                                                                       MSOL0610
      GO TO (10,110,210,310) IOPT                                       MSOL0620
C                                                                       MSOL0630
   10 CONTINUE                                                          MSOL0640
C--------- IOPT=1, REDUCE A AND P ---------- ---------- ----------     MSOL0650
C          GAUSSIAN ELIMINIATION IS USED TO REDUCE THE A AND P MATRICIESMSOL0660
C          FROM ROW L1 TO ROW L2   IF L1 IS NOT 1, IT IS ASSUMED THAT   MSOL0670
C          A AND P ARE ALLREADY REDUCED FROM 1 TO L1 ...                MSOL0680
C-------- ------ ------ ---------- ---------- ------ ---------- ------ MSOL0690
      IF (PRINT) THEN                                                   MSOL0700
        WRITE(108,500) 'IOPT=1, REDUCE STIFFNESS AND LOADS',L1,L2,N     MSOL0710
        CALL LMATRX(A,MD,N,IMD ,'INPUT STIFFNESS')                      MSOL0720
        CALL WMATRX(P,N ,NLC,'INPUT LOAD')                              MSOL0730
        IF (GUYAN ) CALL LMATRX(MASS,MDMASS,N,IMDMAS,'INPUT MASS')      MSOL0740
        IF (KGCOND) CALL LMATRX(KG  ,MDKG  ,N,IMDKG ,'INPUT KG  ')      MSOL0750
      ENDIF                                                             MSOL0760
C                                                                       MSOL0770
      DO 100 L=L1,L2                                                    MSOL0780
        IF (L EQ.N) GO TO 100                                           MSOL0790
C                                                                       MSOL0800
C------- TEST FOR ZERO PIVOT.... .... ...                               MSOL0810
        IF (A(MD(L)).EQ.0) THEN                                         MSOL0820
          WRITE(108,2000) L,MD(L),A(MD(L))                              MSOL0830
          STOP                                                          MSOL0840
        ENDIF                                                           MSOL0850
C                                                                       MSOL0860
C-- ---- CALCULATE THE FACTORS EACH ROW IS MULTIPLIED BY                MSOL0870
        DO 30 I=(L+1),N                                                 MSOL0880
          LI=IJI(L,I)                                                   MSOL0890
          IF ( LI.LT.MD(I-1) ) THEN                                     MSOL0900
            FACTOR(I)=A(LI) /A( MD(L) )                                 MSOL0910
          ELSE                                                          MSOL0920
            FACTOR(I)=0                                                 MSOL0930
          ENDIF                                                         MSOL0940
   30     CONTINUE                                                      MSOL0950
C                                                                       MSOL0960
C------- REDUCE A BY COLUMNS (J) EACH ROW (I)...                        MSOL0970
        DO 40 J=(L+1),N                                                 MSOL0980
          LJ=IJI(L,J)                                                   MSOL0990
          IF (LJ.GE.MD(J+1)) GO TO 40                                   MSOL1000
          DO 35 I=(L+1),J                                               MSOL1010
            IJ=IJI(I,J)                                                 MSOL1020
            A(IJ)=A(IJ)-FACTOR(I)*A(LJ)                                 MSOL1030
   35       CONTINUE                                                    MSOL1040
   40   CONTINUE                                                        MSOL1050
C                                                                       MSOL1060
C------- REDUCE P BY COLUMNS (J) EACH ROW (I)...                        MSOL1070
        DO 50 J=1,NLC                                                   MSOL1080
          IF (P(L,J).EQ.0) GO TO 50                                     MSOL1090
          DO 45 I=(L+1),N                                               MSOL1100
   45       P(I,J)=P(I,J)-FACTOR(I)*P(L,J)                             MSOL1110
   50   CONTINUE                                                        MSOL1120
C                                                                       MSOL1130
C------- REDUCE MASS MATRIX BY GUYAN REDUCTION                          MSOL1140
      IF (GUYAN) THEN                                                   MSOL1150
C                                                                       MSOL1160
C  ------- M22=M22 - K21/K11 * M11 * 1/K11 * K12                        MSOL1170
        RM=MASS(MDMASS(L))                                              MSOL1180
        IF (RM.NE.0) THEN                                               MSOL1190
          DO 60 J=L+1,N                                                 MSOL1200
            IF (FACTOR(J).EQ.0) GO TO 60                                MSOL1210
            I1=MAX(L+1,J-(MDMASS(J+1)-MDMASS(J)) )                      MSOL1220
            DO 55 I=I1,J                                                MSOL1230
              IJ=IJM(I,J)                                               MSOL1240
   55         MASS(IJ)=MASS(IJ)-FACTOR(I)*RM*FACTOR(J)                  MSOL1250
   60     CONTINUE                                                      MSOL1260
        ENDIF                                                           MSOL1270
C                                                                       MSOL1280
C--------- M22=M22 - K21/K11 * M12                                      MSOL1290
        DO 66 J=L+1,N                                                   MSOL1300
          LJ=IJM(L,J)                                                   MSOL1310
          IF (LJ.GE.MDMASS(J+1)) GO TO 66                               MSOL1320
          IF (MASS(LJ).EQ.0) GO TO 66                                   MSOL1330
          DO 65 I=L+1,J                                                 MSOL1340
            IJ=IJM(I,J)                                                 MSOL1350
            IF (IJ.LT.MDMASS(J+1))                                      MSOL1360
     &      MASS(IJ)=MASS(IJ)-FACTOR(I)*MASS(LJ)                        MSOL1370
   65     CONTINUE                                                      MSOL1380
   66   CONTINUE                                                        MSOL1390
C                                                                       MSOL1400
C--------- M22=M22 - M21 * 1/K11 * K12                                  MSOL1410
        DO 70 J=L+1,N                                                   MSOL1420
          IF (FACTOR(J).EQ.0) GO TO 70                                  MSOL1430
          DO 69 I=L+1,J                                                 MSOL1440
            LJ=IJM(L,I)                                                 MSOL1450
            IF (LJ.GE.MDMASS(I+1)) GO TO 69                             MSOL1460
            IJ=IJM(I,J)                                                 MSOL1470
            IF (IJ.LT.MDMASS(J+1))                                      MSOL1480
     &      MASS(IJ)=MASS(IJ)-FACTOR(J)*MASS(LI)                        MSOL1490
   69     CONTINUE                                                      MSOL1500
   70   CONTINUE                                                        MSOL1510
      ENDIF                                                             MSOL1520
C                                                                       MSOL1530
C------- REDUCE GEOMETRIC STIFFNESS MATRIX                              MSOL1540
      IF (KGCOND) THEN                                                  MSOL1550
C                                                                       MSOL1560
C  ------ .. KG22=KG22 - K21/K11 * KG11 * 1/K11 * K12                  MSOL1570
        RKG=KG(MDKG(L))                                                 MSOL1580
        IF (RKG.NE.0) THEN                                              MSOL1590
          DO 80 J=L+1,N                                                 MSOL1600
            IF (FACTOR(J).EQ.0) GO TO 80                                MSOL1610
            I1=MAX(L+1,J-(MDKG(J+1)-MDKG(J)) )                          MSOL1620
            DO 75 I=I1,J                                                MSOL1630
              IJ=IJKG(I,J)                                              MSOL1640
   75         KG(IJ)=KG(IJ)-FACTOR(I)*RKG*FACTOR(J)                     MSOL1650
   80     CONTINUE                                                      MSOL1660
        ENDIF                                                           MSOL1670
C                                                                       MSOL1680
C.. ------ KG22=KG22 - K21/K11 * KG12                                   MSOL1690
        DO 86 J=L+1,N                                                   MSOL1700
          LJ=IJKG(L,J)                                                  MSOL1710
          IF (LJ.GE.MDKG(J+1)) GO TO 86                                 MSOL1720
          IF (KG(LJ).EQ.0) GO TO 86                                     MSOL1730
          DO 85 I=L+1,J                                                 MSOL1740
            IJ=IJKG(I,J)                                                MSOL1750
            IF (IJ.LT.MDKG(J+1))                                        MSOL1760
     &      KG(IJ)=KG(IJ)-FACTOR(I)*KG(LJ)                              MSOL1770
   85     CONTINUE                                                      MSOL1780
   86   CONTINUE                                                        MSOL1790
C                                                                       MSOL1800
C ------ .. KG22=KG22 - KG21 * 1/K11 * K12                            MSOL1810
        DO 90 J=L+1,N                                                   MSOL1820
          IF (FACTOR(J).EQ.0) GO TO 90                                  MSOL1830
          DO 89 I=L+1,J                                                 MSOL1840
            LI=IJKG(L,I)                                                MSOL1850
            IF (LI.GE.MDKG(I+1)) GO TO 89                               MSOL1860
            IJ=IJKG(I,J)                                                MSOL1870
            IF (IJ.LT.MDKG(J+1))                                        MSOL1880
     &      KG(IJ)=KG(IJ)-FACTOR(J)*KG(LI)                              MSOL1890
   89     CONTINUE                                                      MSOL1900
   90   CONTINUE                                                        MSOL1910
C                                                                       MSOL1920
      ENDIF                                                             MSOL1930
C                                                                       MSOL1940
  100 CONTINUE                                                          MSOL1950
      IF (PRINT) THEN                                                   MSOL1960
        CALL LMATRX(A,MD,N,IMD,'REDUCED STIFFNESS')                     MSOL1970
        CALL WMATRX(P,N ,NLC,'REDUCED LOAD')                            MSOL1980
        IF (GUYAN)  CALL LMATRX(MASS,MDMASS,N,IMDMAS,'REDUCED MASS')    MSOL1990
        IF (KGCOND) CALL LMATRX(KG  ,MDKG  ,N,IMDKG ,'REDUCED KG')      MSOL2000
      ENDIF                                                             MSOL2010
      RETURN                                                            MSOL2020
C                                                                       MSOL2030
  110 CONTINUE                                                          MSOL2040
C--------- IOPT=2, REDUCE P ONLY -- ---------- ---------- -- ----------MSOL2050
C          GAUSSIAN ELIMINIATION IS USED TO REDUCE THE P MATRIX         MSOL2060
C          FROM ROW L1 TO ROW L2   IF L1 IS NOT 1, IT IS ASSUMED THAT   MSOL2070
C          A AND P ARE ALLREADY REDUCED FROM 1 TO L1..                  MSOL2080
C---------- ---------- ---------- ---------- ---------- -- ----------MSOL2090
      IF (PRINT) THEN                                                   MSOL2100
        WRITE(108,500) 'IOPT=2, REDUCE LOADS ONLY',L1,L2,N              MSOL2110
        CALL LMATRX(A,MD,N,MD(N+1),'REDUCED STIFFNESS')                 MSOL2120
        CALL WMATRX(P,N ,NLC,'NEW LOAD MATRIX ')                        MSOL2130
      ENDIF                                                             MSOL2140
C                                                                       MSOL2150
C ------ LOOP FOR EACH ROW (L) TO REDUCE . .                            MSOL2160
      DO 160 L=L1,L2                                                    MSOL2170
```

```
C                                                                       MSOL2180
        IF (A(MD(L)).EQ.0) THEN                                         MSOL2190
          WRITE(108,2000) L,MD(L),A(MD(L))                              MSOL2200
          STOP                                                          MSOL2210
        ENDIF                                                           MSOL2220
C                                                                       MSOL2230
C       - LOOP FOR EACH COLUMN (J)                                      MSOL2240
        DO 150 J=1,NLC                                                  MSOL2250
          IF (P(L,J).EQ.0) GO TO 150                                    MSOL2260
          DO 120 I=(L+1),N                                              MSOL2270
            LI=IJI(L,I)                                                 MSOL2280
            IF (LI.LT.MD(I-1)) P(I,J)=P(I,J)-A(LI)*P(L,J)/A(MD(L))      MSOL2290
  120       CONTINUE                                                    MSOL2300
  150   CONTINUE                                                        MSOL2310
  160   CONTINUE                                                        MSOL2320
      RETURN                                                            MSOL2330
C                                                                       MSOL2340
  210 CONTINUE                                                          MSOL2350
C--------- IOPT=3, BACK SUBSTUTION ------------------------------------MSOL2360
C         BACK SUBSTUTION IS USED TO SOLVE FOR X IN SX=V, WHERE S IS    MSOL2370
C         THE REDUCE UPPER TRIANGULAR FACTOR OF THE A MATRIX, AND V     MSOL2380
C         IS THE REDUCED FORRM OF THE P MATRIX.  THE RESULTS (X) ARE    MSOL2390
C         STORED IN P.                                                  MSOL2400
C         BACK SUBSTUTION ONLY TAKES PLACE BETWEEN ROWS L1 AND L2.      MSOL2410
C------- ---------- ---------- ---------- ---------- ---------- ------MSOL2420
      IF (PRINT) THEN                                                   MSOL2430
        WRITE(108,500) 'IOPT=3, BACK SUBSTUTION',L1,L2,N                MSOL2440
        CALL LMATRX(A,MD,N,MD(N+1),'REDUCED STIFFNESS')                 MSOL2450
        CALL WMATRX(P,N ,NLC,'REDUCED LOAD')                            MSOL2460
      ENDIF                                                             MSOL2470
C                                                                       MSOL2480
C------ LOOP FOR EACH ROW L                                             MSOL2490
C       CALCULAGE X FOR EACH LOAD CASE, STORE IN P                      MSOL2500
        DO 250 J=L2,L1,-1                                               MSOL2510
        IF (A(MD(J)).EQ.0) THEN                                         MSOL2520
          WRITE(108,2000) J,MD(J),A(MD(J))                              MSOL2530
          STOP                                                          MSOL2540
        ENDIF                                                           MSOL2550
        DO 215 LC=1,NLC                                                 MSOL2560
  215     P(J,LC)=P(J,LC)/A(MD(J))                                      MSOL2570
        IF (J.GT.1) THEN                                                MSOL2580
          IJ=MD(J-1)                                                    MSOL2590
          NI=J-(MD(J+1)-MD(J)) +1                                       MSOL2600
          DO 220 I=NI,J-1,-1                                            MSOL2610
            IJ=IJ-1                                                     MSOL2620
            DO 220 LC=1,NLC                                             MSOL2630
              P(I,LC)=P(I,LC)-P(J,LC)*A(IJ)                             MSOL2640
  220     CONTINUE                                                      MSOL2650
        ENDIF                                                           MSOL2660
  250   CONTINUE                                                        MSOL2670
      IF (PRINT) CALL WMATRX(P,N ,NLC,'DISPLACEMENT')                   MSOL2680
C                                                                       MSOL2690
      RETURN                                                            MSOL2700
C                                                                       MSOL2710
  310 CONTINUE                                                          MSOL2720
C--------- IOPT=4, REDUCED MULTIPLICATION -----------------------------MSOL2730
C         SOLVE FOR V IN S'D-P WHERE S HAS BEEN REDUCED BY GAUSSIAN     MSOL2740
C         ELIMINATION.                                                  MSOL2750
C         THE RESULTING LOAD IS STORED IN P.                            MSOL2760
C         THE DISPLACEMENTS ARE INPUT IN D                              MSOL2770
C         MULTIPLACATION ONLY TAKES PLACE BETWEEN ROWS L1 AND L2.       MSOL2780
C------- ---------- ---------- ---------- ---------- ---------- ------MSOL2790
      IF (PRINT) THEN                                                   MSOL2800
        WRITE(108,500) 'IOPT=4, REDUCED MULTIPLACATION',L1,L2,N         MSOL2810
        CALL LMATRX(A,MD,N,MD(N+1),'REDUCED STIFFNESS')                 MSOL2820
        CALL WMATRX(FACTOR,N ,1,'DISPLACEMENT')                         MSOL2830
      ENDIF                                                             MSOL2840
C                                                                       MSOL2850
C---- - ONLY ONE LOAD CASE. ..                                          MSOL2860
      LC=1                                                              MSOL2870
C       ---- ZERO P ----                                               MSOL2880
        DO 320 J=L1,L2                                                  MSOL2890
          IF (A(MD(J)).EQ.0) THEN                                       MSOL2900
            WRITE(108,2000) J,MD(J),A(MD(J))                            MSOL2910
            STOP                                                        MSOL2920
          ENDIF                                                         MSOL2930
  320     P(J,LC)=0                                                     MSOL2940
C                                                                       MSOL2950
C---------- MULTIPLY S_RED * DISP = P                                   MSOL2960
      DO 340 J=L1,N                                                     MSOL2970
        IC1=J-(MD(J+1)-MD(J)) +1                                        MSOL2980
        IC2=MAX(IC1,L1)                                                 MSOL2990
        DO 340 I=J,IC2,-1                                               MSOL3000
          IJ=IJI(I,J)                                                   MSOL3010
  340     P(I,LC)=P(I,LC)+A(IJ)*FACTOR(J)                               MSOL3020
      IF (PRINT) CALL WMATRX(P,N ,1,'REDUCED LOAD')                     MSOL3030
C                                                                       MSOL3040
C---- - UNREDUCE THE LOAD MATRIX ..                                     MSOL3050
      DO 360 L=L2,L1,-1                                                 MSOL3060
C                                                                       MSOL3070
C       LOOP FOR EACH COLUMN (J)                                        MSOL3080
        DO 350 J=1,NLC                                                  MSOL3090
          IF (P(L,LC) EQ.0) GO TO 350                                   MSOL3100
          DO 345 I=(L+1),N                                              MSOL3110
            LI=IJI(L,I)                                                 MSOL3120
            IF (LI.LT.MD(I-1)) P(I,LC)=P(I,LC)+A(LI)*P(L,LC)/A(MD(L))   MSOL3130
  345     CONTINUE                                                      MSOL3140
  350   CONTINUE                                                        MSOL3150
  360   CONTINUE                                                        MSOL3160
  400   CONTINUE                                                        MSOL3170
C                                                                       MSOL3180
      IF (PRINT) CALL WMATRX(P,N ,1,'LOAD')                             MSOL3190
  500 FORMAT (/29A,2X,'L1 =',I5,'   L2 =',I5,'   N =',I5)               MSOL3200
 2000 FORMAT (///1X,50('*')/                                            MSOL3210
     &        ' *  ER','ROR IN SUBROUTINE MSOLL',        T51,'*'/      MSOL3220
     &        ' *  PIVOT IS LE 0 ------- ROW ',I5,      T51,'*'/      MSOL3230
     &        ' *                STORAGE LCN ',I5,      T51,'*'/      MSOL3240
     &        ' *                            PIVOT=',1P,G15.6,T51,'*'/ MSOL3250
     &        ' *  SOLUTION IS ABORTED ',               T51,'*'/      MSOL3260
     &        1X,50('*')/)                                            MSOL3270
C                                                                       MSOL3280
      RETURN                                                            MSOL3290
      END                                                               MSOL3300
C                                                                       REAC0010
C-----------------------------------------------------------------------REAC0020
C                                                                       REAC0030
      SUBROUTINE REACTN(IOPT,PRINT,NNODE,L1,L2,L3,L4,FORCE,DISPL,       REAC0040
     &                  MD,A,KM,NDOF,NLOAD,FACTOR,                      REAC0050
     &                  MAXNOD,IDOF,COORD,ID,TITLE,STEPID,HEAD,         REAC0060
     &                  NCOS,COSINE,JTCOS,JTFLG,SUM)                    REAC0070
      CHARACTER*(*) TITLE(2),STEPID                                     REAC0080
      CHARACTER*15 CD(6)                                                REAC0090
      LOGICAL PRINT,HEAD,FLAG,BTEST                                     REAC0100
      DIMENSION A(KM),MD(NDOF+1),IDOF(MAXNOD,6),G(6)                    REAC0110
      DIMENSION FORCE(NDOF,NLOAD),DISPL(NDOF,NLOAD)                     REAC0120
      DIMENSION ID(MAXNOD),COORD(MAXNOD,3),JTFLG(MAXNOD)                REAC0130
      DIMENSION F(6),SUM(6),COSINE(3,3,NCOS),JTCOS(MAXNOD)              REAC0140
C                                                                       REAC0150
      IF (IOPT.EQ.1) THEN                                               REAC0160
C                                                                       REAC0170
C-----------------------------------------                              REAC0180
C- MODIFY LOADS FOR RESTRAINT DISPLACEMENTS ---                         REAC0190
C-----------------------------------------                              REAC0200
C CONSIDER THE PARTITIONED STIFFNESS WHERE X2 ARE THE RESTRAINT DISPL.  REAC0210
C                                                                       REAC0220
C   :P1:   : K11 : K12 : :X1:                                          REAC0230
C   :---: = :-----+-----: :--: * FACTOR                                 REAC0240
C   :P2:   : K21 : K22 : :X2:                                          REAC0250
C                                                                       REAC0260
C   THE MODIFIED FREE LOAD IS (P1-K12*X2) - K11*X1                      REAC0270
C------ --- ------ ---- -- ---------------- --------                    REAC0280
C     DO 10 I=L1,L2                                     C  FULL STORAGE REAC0290
C     DO 10 J=1,NLOAD                                   C  EQUIVALENT   REAC0300
C     DO 10 K=L3,L4                                     C               REAC0310
C  10   FORCE(I,J)=FORCE(I,J)-K(I,K)*DISPL(K,J)         C               REAC0320
C------------------------- ----- --------- ------------C               REAC0330
        DO 20 K=L3,L4                                                   REAC0340
        MDK=MD(K+1)                                                     REAC0350
          DO 20 J=1,NLOAD                                               REAC0360
          DO 10 I=L2,L1,-1                                              REAC0370
            IK=MD(K)+K-I                                                REAC0380
            IF (IK.GE.MDK) GO TO 20                                     REAC0390
            FIJ=FORCE(I,J)                                              REAC0400
            AIK=A(IK)                                                   REAC0410
            DKJ=DISPL(K,J)                                              REAC0420
   10       FORCE(I,J)=FORCE(I,J)-A(IK)*DISPL(K,J)                     REAC0430
   20   CONTINUE                                                        REAC0440
        IF (PRINT)                                                      REAC0450
     &  CALL WMATRX(FORCE,NDOF,NLOAD,'MODIFIED LOADS')                  REAC0460
C                                                                       REAC0470
      ELSE IF (IOPT.EQ.2) THEN                                          REAC0480
C                                                                       REAC0490
C-------------------------                                              REAC0500
C- SOLVE FOR REACTIONS ---                                              REAC0510
C-------------------------                                              REAC0520
C CONSIDER THE PARTITIONED STIFFNESS WHERE X2 ARE THE RESTRAINT DISPL.  REAC0530
C                                                                       REAC0540
C   :P1:   : K11 : K12 : :X1:                                          REAC0550
C   :---: = :-----+-----: :--: * FACTOR                                 REAC0560
C   :P2:   : K21 : K22 : :X2:                                          REAC0570
C                                                                       REAC0580
C  THE REACTIONS ARE  P2 = P2 + ( K21*X1 + K22*X2 ) * FACTOR            REAC0590
C   NOTE: BOTH P1 AND P2 CONTAIN FIXED END FORCES FROM MEMBER LOADS.    REAC0600
C         THUS, K21*X1 + K22*X2 IS ADDED TO P2 TO GET THE REACTIONS.    REAC0610
C-------------------------------------------C                           REAC0620
C     DO 120 I=L3,L4                        C    FULL STORAGE         REAC0630
C     DO 120 J=1,NLOAD                      C    MODE EQUIVALENT      REAC0640
CC     FORCE(I,J)=0   * P2 CONTAINS FEF!!!  C                           REAC0650
C     DO 110 K=L1,L2                        C                           REAC0660
C 110   FORCE(I,J)=FORCE(I,J)+K(I,K)*DISPL(K,J) C                       REAC0670
C     DO 120 K=L3,L4                        C                           REAC0680
C 120   FORCE(I,J)=FORCE(I,J)+K(I,K)*DISPL(K,J) C                       REAC0690
C-------------------------------------------C                           REAC0700
C                                                                       REAC0710
        DO 120 I=L3,L4                                                  REAC0720
        DO 120 J=1,NLOAD                                                REAC0730
        DO 110 K=L2,L1,-1                                               REAC0740
          KI=MD(I)+I-K                                                  REAC0750
          IF (KI.GE.MD(I+1)) GO TO 120                                  REAC0760
          AKI=A(KI)                                                     REAC0770
  110   FORCE(I,J)=FORCE(I,J)+A(KI)*DISPL(K,J)*FACTOR                   REAC0780
  120 CONTINUE                                                          REAC0790
C                                                                       REAC0800
        DO 140 I=L3,L4                                                  REAC0810
        DO 140 K=L3,L4                                                  REAC0820
          IF (I.LE.K) THEN                                              REAC0830
            IK=MD(K)+K-I                                                REAC0840
            MDK=MD(K+1)                                                 REAC0850
          ELSE                                                          REAC0860
            IK=MD(I)+I-K                                                REAC0870
            MDK=MD(I+1)                                                 REAC0880
          ENDIF                                                         REAC0890
          IF (IK.GE.MDK)  GO TO 140                                     REAC0900
          AIK=A(IK)                                                     REAC0910
          DO 130 J=1,NLOAD                                              REAC0920
  130       FORCE(I,J)=FORCE(I,J)+A(IK)*DISPL(K,J)*FACTOR               REAC0930
  140   CONTINUE                                                        REAC0940
C                                                                       REAC0950
        IF (PRINT)                                                      REAC0960
     &  CALL WMATRX(FORCE,NDOF,NLOAD,'FINAL LOADS AND REACTIONS')       REAC0970
C                                                                       REAC0980
      ELSE IF (IOPT.EQ.3 .OR. IOPT.EQ.4) THEN                           REAC0990
        NP=L2                                                           REAC1000
C-----------------------------                                          REAC1010
C- LOOP FOR EACH LOAD CASE  ---                                         REAC1020
C-----------------------------                                          REAC1030
        DO 300 L=1,NLOAD                                                REAC1040
C                                                                       REAC1050
C-------------------------------                                        REAC1060
C- PRINT GLOBAL REACTIONS     ---                                       REAC1070
C-------------------------------                                        REAC1080
C                                                                       REAC1090
        IF (PRINT) THEN                                                 REAC1100
          IF (HEAD) WRITE(108,320) TITLE(1),TITLE(2)                    REAC1110
            NREACT=0                                                    REAC1120
            IF (IOPT.NE.4) WRITE(108,321) L,STEPID                      REAC1130
            IF (IOPT.EQ.4) WRITE(108,421)                               REAC1140
        ENDIF                                                           REAC1150
          IF (IOPT.EQ.3) THEN                                           REAC1160
            DO 190 I=1,6                                                REAC1170
  190       SUM(I)=0                                                    REAC1180
          ENDIF                                                         REAC1190
C                                                                       REAC1200
        IF ( .NOT.(NCOS.EQ.1 .AND. COSINE(1,1,1).EQ.1 .AND.COSINE(2,2,1)REAC1210
     &      .EQ.1 .AND. COSINE(3,3,1).EQ.1).AND. IOPT.EQ 4  ) GO TO 279 REAC1220
C                                                                       REAC1230
          DO 200 I=1,NNODE                                              REAC1240
C                                                                       REAC1250
C ...   ..... GET REACTNS IN JOINT COORDINATE SYSTEM                    REAC1260
            FLAG=.FALSE.                                                REAC1270
            DO 209 J=1,6                                                REAC1280
              IJ=IDOF(I,J)                                              REAC1290
              IF (IJ.LE.NP  OR. BTEST(JTFLG(I),J+11)) THEN              REAC1300
                F(J)=0                                                  REAC1310
              ELSE                                                      REAC1320
                F(J)=FORCE(IJ,L)                                        REAC1330
                FLAG=.TRUE.                                             REAC1340
              ENDIF                                                     REAC1350
  209       CONTINUE                                                    REAC1360
            IF ( NOT FLAG) GO TO 200                                    REAC1370
C                                                                       REAC1380
C... .  .. .... TRANSFER REACTNS TO GLOBAL COORDINATE SYSTEM            REAC1390
            DO 229 I1=1,3                                               REAC1400
              I2=I1+3                                                   REAC1410
              G(I1)=0                                                   REAC1420
              G(I2)=0                                                   REAC1430
              DO 219 J1=1,3                                             REAC1440
                J2=J1+3                                                 REAC1450
                G(I1)=G(I1) + COSINE(J1,I1,JTCOS(I))*F(J1)              REAC1460
  219           G(I2)=G(I2) + COSINE(J1,I1,JTCOS(I))*F(J2)              REAC1470
              IF (PRINT) THEN                                           REAC1480
                  WRITE(CD(I1),210) G(I1)                               REAC1490
                  WRITE(CD(I2),210) G(I2)                               REAC1500
              ENDIF                                                     REAC1510
  229         CONTINUE                                                  REAC1520
C                                                                       REAC1530
C.... . .  .. SUM REACTIONS  ...  . ..                                  REAC1540
          IF (IOPT.EQ.3) THEN                                           REAC1550
            DO 240 I1=1,6                                               REAC1560
  240         SUM(I1)=SUM(I1) + G(I1)                                   REAC1570
            SUM(4)= SUM(4) - G(2)*COORD(I,3) + G(3)*COORD(I,2)          REAC1580
            SUM(5)= SUM(5) + G(1)*COORD(I,3) - G(3)*COORD(I,1)          REAC1590
            SUM(6)= SUM(6) - G(1)*COORD(I,2) + G(2)*COORD(I,1)          REAC1600
          ENDIF                                                         REAC1610
C                                                                       REAC1620
C.. ..... .. PRINT GLOBAL REACTIONS .... ....                          REAC1630
          NREACT=NREACT-1                                               REAC1640
          IF (PRINT) WRITE(108,220) ID(I),(CD(K),K=1,6)                 REAC1650
  200   CONTINUE                                                        REAC1660
        IF (IOPT.EQ.3  AND  PRINT ) WRITE(108,230) SUM                  REAC1670
  279   IF (IOPT EQ 4 .AND  PRINT ) WRITE(108,231) SUM                  REAC1680
C                                                                       REAC1690
C-----------------------------------------------                        REAC1700
C- PRINT LOCAL JOINT COORDINATE SYSTEMS REACTIONS  ---                  REAC1710
C-----------------------------------------------                        REAC1720
C                                                                       REAC1730
        IF ( NOT  PRINT ) GO TO 300                                     REAC1740
        IF (NCOS EQ.1 .AND. COSINE(1,1,1).EQ.1 .AND. COSINE(2,2,1) EQ 1 REAC1750
     &                  AND. COSINE(3,3,1) EQ.1 ) GO TO 300             REAC1760
C                                                                       REAC1770
        IF (HEAD.AND. NREACT.GT 20) WRITE(108,320) TITLE(1),TITLE(2)    REAC1780
          IF (IOPT NE 4) WRITE(108,322) L,STEPID                        REAC1790
```

```
      IF (IOPT.EQ.4) WRITE(108,422)                                     REAC1800
      DO 290 I=1,NNODE                                                  REAC1810
C                                                                       REAC1820
C  . . . . . . GET REACTNS IN JOINT COORDINATE SYSTEM                   REAC1830
        FLAG=.FALSE                                                     REAC1840
        DO 280 J=1,6                                                    REAC1850
          IJ=IDOF(I,J)                                                  REAC1860
          IF (IJ.LE.NP .OR. BTEST(JTFLG(I),J+11)) THEN                  REAC1870
            CD(J)=' '                                                   REAC1880
          ELSE                                                          REAC1890
            WRITE(CD(J),210) FORCE(IJ,L)                                REAC1900
            FLAG=.TRUE                                                  REAC1910
          ENDIF                                                         REAC1920
  280   CONTINUE                                                        REAC1930
        IF (.NOT.FLAG) GO TO 290                                        REAC1940
C                                                                       REAC1950
C ...     . . . . PRINT REACTNS IN LOCAL COORDINATE SYSTEM . ...        REAC1960
        WRITE(108,221) ID(I),(CD(K),K=1,6),JTCOS(I)                     REAC1970
  290   CONTINUE                                                        REAC1980
  300 CONTINUE                                                          REAC1990
C                                                                       REAC2000
  210 FORMAT (1P,G15.7)                                                 REAC2010
  220 FORMAT (5X,I5,6A)                                                 REAC2020
  221 FORMAT (5X,I5,6A,I10)                                             REAC2030
  230 FORMAT (' ',90('-'),/' SUMMATION',6G15.7)                         REAC2040
  231 FORMAT (' MAX OF ALL',89('-'),/' GCS SUMM.',6G15.7 )              REAC2050
  320 FORMAT ('1 STRUCTURE. . . . ',A,/'  SOLUTION. . . : ',A)          REAC2060
  321 FORMAT (//'  GCS RESTRAINT REACTIONS,   LOADING #',I5,5X,A,/      REAC2070
     &        ' ------------------------'//6X,'NODE',7X,'FX',          REAC2080
     &          13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ'/)        REAC2090
  322 FORMAT (//'  JCS RESTRAINT REACTIONS,   LOADING #',I5,5X,A,/     REAC2100
     &        ' ------------------------'//6X,'NODE',7X,'FX',          REAC2110
     &          13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ',         REAC2120
     &          6X,'  COSINE #'/)                                     REAC2130
  421 FORMAT (//'  MAXIMUM GCS RESTRAINT REACTIONS',                   REAC2140
     &       5X,'NOTE. MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'/  REAC2150
     &        ' ------------------------------'//6X,'NODE',7X,'FX',    REAC2160
     &          13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ'/)        REAC2170
  422 FORMAT (//'  MAXIMUM JCS RESTRAINT REACTIONS ',                  REAC2180
     &       5X,'NOTE: MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'/  REAC2190
     &        ' ------------------------------'//6X,'NODE',7X,'FX',    REAC2200
     &       13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ',            REAC2210
     &       6X,'  COSINE #'/)                                        REAC2220
C                                                                       REAC2230
      ELSE                                                              REAC2240
       WRITE(108,*) 'INVALID OPTION IN SUBROUTINE  REA_CTN, IOPT=',IOPT REAC2250
      ENDIF                                                             REAC2260
      RETURN                                                            REAC2270
      END                                                               REAC2280
C                                                                       PORM0010
C*********************************************************************  PORM0020
C                                                                       PORM0030
      SUBROUTINE FORM(IOPT)                                             PORM0040
      CHARACTER  NAME*80,ELNO*130                                       PORM0050
      LOGICAL PRINT,HEAD,FALSE,AXIAL,BTEST,BUGS,BUG5?                   PORM0060
$INCLUDE:'ZCOMN'                                                        PORM0070
C                                                                       PORM0080
C ------------------------------------------------------------------    PORM0090
C   MATRIX STORAGE SCHEME:                                              PORM0100
C  -THE STIFFNESS MATRIX (K) IS BANDED AND SYMMETRIC.                   PORM0110
C   THUS ONLY THE SKYLINE ABOVE THE MAIN DIAGONAL NEEDS TO BE STORED.   PORM0120
C   THE MATRIX K IS STORED IN A LINEAR ARRAY BY COLUMNS                 PORM0130
C   THE VALUE ON THE MAIN DIAGONAL IS STORED IN THE LINEAR ARRAY FIRST. PORM0140
C   RELATIVE ADDRESSES OF THE LINEAR ARRAY ELEMENTS CONTAINING THE      PORM0150
C   MAIN DIAGONAL TERMS ARE STORED IN MD                                PORM0160
C   THE ABSOLUTE ADDRESS OF THE FIRST ELEMENT IN THE K MATRIX IS IZSTIP.PORM0170
C   THE STIFFNESS MATRIX K IS STORED IN THE Z ARRAY, Z(IZSTIP)=K(1,1)   PORM0180
C   THE ABSOLUTE ADDRESS OF THE FIRST ELEMENT IS THE MD MATRIX IS IZMD. PORM0190
C   THE STIFFNESS MATRIX K IS STORED IN THE NZ ARRAY, NZ(IZMD)=MD(1)    PORM0200
C   THUS  IF I LE J (UPPER TRIANGULAR MATRIX)                           PORM0210
C         IF IZMD-1+J < IZMD-1+J+1                                      PORM0220
C                 IJ=NZ(IZMD-1+J)+J-I : RELATIVE ADDRESS IF ELEMENT (I,J)PORM0230
C                 B(I,J)=Z(IZSTIP-1 + IJ)                               PORM0240
C          ELSE                                                         PORM0250
C                 B(I,J)=0              : (I,J) IS ABOVE THE SKYLINE    PORM0260
C                                                                       PORM0270
C   EXAMPLE: LET B BE THE 5X5 FULL SYMETRIC MATRIX BELOW...             PORM0280
C       | 1 | 3 |   |  12 |                                             PORM0290
C       |   | 2 | 5 |  11 |  NOTE: B(1,3),B(1,4) AND B(2,4) ARE ZERO. PORM0300
C   B=  |   |   | 4 | 7 | 10 |                                        PORM0310
C       |  SYMM |   | 6 |  9 |  MD= 1, 2, 4, 6, 8, 13                  PORM0320
C       |       |   |   |  8 |                                         PORM0330
C   SAY IZMD-121,IZSTIP=200                                             PORM0340
C   NZ(121)=1          Z(200)=1                                         PORM0350
C   NZ(122)=2          Z(201)=2                                         PORM0360
C   NZ(123)=4          Z(202)=3                                         PORM0370
C   NZ(124)=6          Z(203)=4                                         PORM0380
C   NZ(125)=8          Z(204)=5                                         PORM0390
C   NZ(126)=13         Z(205)=6                                         PORM0400
C                      Z(206)=7                                         PORM0410
C                      Z(206)=8                                         PORM0420
C                      Z(206)=9                                         PORM0430
C                      Z(206)=10                                        PORM0440
C                      Z(206)=11                                        PORM0450
C                      Z(206)=12                                        PORM0460
C   THUS, FOR B(3,4)  I=3, J=4                                          PORM0470
C             IJ=NZ(IZMD-1+J)+J-I = NZ(121-1+4) +4-3 = 6+4-3 =7         PORM0480
C             B(I,J)=Z(IZSTIP ) + IJ) = Z(200-1+7) = Z(206) = 7  OK     PORM0490
C   THUS, FOR B(2,4)   I=2, J=4                                         PORM0500
C             IJ=NZ(IZMD-1+J)+J-I = NZ(121-1+4) +4-2 = 6+4-2 =8         PORM0510
C             NZ(IZMD -1+J+1)-NZ(121-1+4+1)=8                           PORM0520
C             SINCE IJ GE NZ(IZMD-1+J+1)                                PORM0530
C             B(I,J)=0  (I,J) IS ABOVE THE SKYLINE..                    PORM0540
C ------------------------------------------------------------------    PORM0550
C                                                                       PORM0560
      BUG5 = BTEST(IBUG,5)                                              PORM0570
      BUG5?= BTEST(IBUG,7) .AND. BUG5                                   PORM0580
      LSTTYP=0                                                          PORM0590
      PRINT=.FALSE                                                      PORM0600
      HEAD =.FALSE                                                      PORM0610
C                                                                       PORM0620
C------- ----- ---------- -------- ---------- ------------------------ PORM0630
C IOPT=1 FORM STIFFNESS MATRIX                                          PORM0640
C IOPT=2 FORM GEOMETRIC STIFFNESS MATRIX                                PORM0650
C---------------------- ---  ------------------------------ ----------PORM0660
C                                                                       PORM0670
       IF (BUG5)  WRITE (6,*) ' IN ROUTINE FORM IOPT=',IOPT             PORM0680
C                                                                       PORM0690
   49 FORMAT (1X,A/('  I =',I3,'  MDMASS(I) =',I11))                   PORM0700
C                                                                       PORM0710
      GO TO (100,200) IOPT                                              PORM0720
C                                                                       PORM0730
C-------------------------                                              PORM0740
C- ASSEMBLE STIFFNESS  -----                                            PORM0750
C-------------------------                                              PORM0760
  100 CONTINUE                                                          PORM0770
          IST =IZSTIP                                                   PORM0780
          IEND=IZSTIP +NZ(IZMD+NDOF) -1                                 PORM0790
C                                                                       PORM0800
C----------  ZERO STIFFNESS                                             PORM0810
      DO 110 I=IST,IEND                                                 PORM0820
  110   Z(I)=0                                                          PORM0830
C                                                                       PORM0840
C--- -------- LOOP FOR EACH ELEMENT                                     PORM0850
      DO 120 IELNO=1,NELMT                                              PORM0860
C                                                                       PORM0870
C----------- GET STIFFNESS OF EACH ELEMENT                              PORM0880
        FALSE=.FALSE                                                    PORM0890
        CALL ELELIB                                                     PORM0900
     &  (3,LSTTYP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DAMAGE,DUCFAG,        PORM0910
     &  IELNO ,IELDOF,KGDOF,PGEOM,2  ,AXIAL,IZDISP,IZLOAD,MSTOR)       PORM0920
C                                                                       PORM0930
       IF (BUG5?) THEN                                                  PORM0940
         WRITE (ELNO,15)  IELNO,(NZ(I-IZLM 1),I=1,MIN(IELDOF,18))      PORM0950
   15    FORMAT ('ELEMENT #',I6,' LM =',18I6)                          PORM0960
         CALL LMATRX(Z(IZKE),MD,IELDOF,MD(IELDOF+1),ELNO)              PORM0970
       ENDIF                                                            PORM0980
C                                                                       PORM0990
      CALL FORML(Z(IZSTIF),NZ(IZMD),NDOF,NZ(IZMD+NDOF),                 PORM1000
     &           Z(IZKE),MD,IELDOF,MD(IELDOF+1),NZ(IZLM),1 )            PORM1010
C                                                                       PORM1020
C---------- SUBTRACT GEOMETRIC STIFFNESS                                PORM1030
      NZ(IZKGDT+3)=NZ(IZKGDT+3)                                         PORM1040
C                                                                       PORM1050
      IF (NZ(IZKGDT+3).EQ.1) THEN                                       PORM1060
        FALSE=.FALSE                                                    PORM1070
        CALL ELELIB                                                     PORM1080
     &  (2,LSTTYP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DAMAGE,DUCFAG,        PORM1090
     &  IELNO ,IELDOF,KGDOF,PGEOM,2  ,AXIAL,IZDISP,IZLOAD,MSTOR)       PORM1100
C                                                                       PORM1110
        KGDOF=KGDOF                                                     PORM1120
C                                                                       PORM1130
        IF (BUG5?) THEN                                                 PORM1140
          WRITE (ELNO,15)  IELNO,(NZ(I+IZLMXG-1),I=1,MIN(KGDOF,18))    PORM1150
          CALL LMATRX(Z(IZKEKG),MD,KGDOF,MD(KGDOF+1),ELNO)              PORM1160
        ENDIF                                                           PORM1170
C                                                                       PORM1180
        IF ( NOT. AXIAL) GO TO 120                                      PORM1190
C            DETERMINE THE FACTOR FOR KG                                PORM1200
             IF (NZ(IZKGDT+1).EQ.1) THEN                                PORM1210
                  ACCELZ= Z(IZKGDT)                                     PORM1220
                  PGEOM=PGEOM*(1+ACCELZ)                                PORM1230
             ENDIF                                                      PORM1240
C     SWAP THE SIGN ON PGEOM FOR SUBTRACTION.....                       PORM1250
      PGEOM=-PGEOM                                                      PORM1260
C                                                                       PORM1270
      CALL FORML(Z(IZSTIF),NZ(IZMD),NDOF,NZ(IZMD+NDOF),                 PORM1280
     &      Z(IZKEKG),MD,KGDOF,MD(KGDOF+1),NZ(IZLMXG),PGEOM)            PORM1290
      ENDIF                                                             PORM1300
C                                                                       PORM1310
  120   CONTINUE                                                        PORM1320
C                                                                       PORM1330
      RETURN                                                            PORM1340
C                                                                       PORM1350
C---------- ASSEMBLE GEOMETRIC STIFNESS                                 PORM1360
  200 CONTINUE                                                          PORM1370
      IF (NZ(IZKGDT+3).NE.2 ) RETURN                                    PORM1380
          IST =IZKG                                                     PORM1390
          IEND=IZKG + NZ(IZMDKG+NDOF) - 1                               PORM1400
C                                                                       PORM1410
C----------- ZERO STIFFNESS                                             PORM1420
      DO 210 I=IST,IEND                                                 PORM1430
  210   Z(I)=0                                                          PORM1440
C                                                                       PORM1450
C----------- LOOP FOR EACH ELEMENT                                      PORM1460
      DO 220 IELNO=1,NELMT                                              PORM1470
C                                                                       PORM1480
C----------- GET STIFFNESS OF EACH ELEMENT                              PORM1490
        FALSE=.FALSE                                                    PORM1500
        CALL ELELIB                                                     PORM1510
     &  (2,LSTTYP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DAMAGE,DUCFAG,        PORM1520
     &  IELNO ,IELDOF,KGDOF,PGEOM,2  ,AXIAL,IZDISP,IZLOAD,MSTOR)       PORM1530
C                                                                       PORM1540
      IF (.NOT. AXIAL) GO TO 220                                        PORM1550
C                                                                       PORM1560
        IF (BUG5?) THEN                                                 PORM1570
          WRITE (6,*)  'PGEOM=',PGEOM                                   PORM1580
          WRITE (ELNO,15)  IELNO,(NZ(I+IZLMXG-1),I=1,MIN(KGDOF,18))    PORM1590
          CALL LMATRX(Z(IZKEKG),MD,KGDOF,MD(KGDOF+1),ELNO)              PORM1600
        ENDIF                                                           PORM1610
C                                                                       PORM1620
      CALL FORML(Z(IZKG),NZ(IZMDKG),NDOF,NZ(IZMDKG+NDOF),               PORM1630
     &      Z(IZKEKG),MD,KGDOF,MD(KGDOF+1),NZ(IZLMXG),PGEOM)            PORM1640
  220   CONTINUE                                                        PORM1650
C                                                                       PORM1660
      RETURN                                                            PORM1670
      END                                                               PORM1680
C*********************************************************************  CPEM50010
C                                                                       CPEM50020
C   PROGRAM PEM - ANALYSIS OF STRUCTURES                                CPEM50030
C                                                                       CPEM50040
C*********************************************************************  CPEM50050
C                                                                       CPEM50060
C   NOTE. USE IBM FORTVS2 COMPILER OPTION AUTODBL(DBLPAD4)              CPEM50070
C         TO COMPILE THIS PROGRAM IN DOUBLE PRECISION                   CPEM50080
C                                                                       CPEM50090
C   NOTE: USE IBM FORTVS2 COMPILER OPTION OPT(2)                        CPEM50100
C         TO OPTIMIZE THE CODE FOR FASTER EXECUTION                     CPEM50110
C                                                                       CPEM50120
C---------------------------------------------------------------------  CPEM50130
      CHARACTER*80 OPTION,INPUT,BLANK                                   PEM50140
      CHARACTER*8  TIME1,DATE1                                          PEM50150
      CHARACTER*1  CHR(0:25)                                            PEM50160
      LOGICAL TEST,HEAD,AXIAL                                           PEM50170
$INCLUDE:'ZCOMN'                                                        PEM50180
      DATA CHR/'A','B','C','D','E','F','G','H','I','J','K','L','M',    PEM50190
     &         'N','O','P','Q','R','S','T','U','V','W','Z','Y','Z'/    PEM50200
C                                                                       PEM50210
C*************************                                              PEM50220
C** ECHO THE INPUT.....    **                                           PEM50230
C*************************                                              PEM50240
      I=55                                                              PEM50250
      L=0                                                               PEM50260
9995  READ(105,('(A)'),END=9996) INPUT                                  PEM50270
      IF (INDEX(INPUT,'NOECHO').NE.0) GO TO 9996                        PEM50280
      I=I+1                                                             PEM50290
      L=L+1                                                             PEM50300
      IF (  I.GT.55 ) THEN                                              PEM50310
        WRITE(108,9994) (XX,XX=1,8)                                     PEM50320
        I=0                                                             PEM50330
      ENDIF                                                             PEM50340
      WRITE(108,9997) L,INPUT                                           PEM50350
      IF (INDEX(INPUT,'STOP'  ).EQ.0) GO TO 9995                        PEM50360
9994  FORMAT ('1    ECHO OF INPUT DATA '//                              PEM50370
     &       1X,3X,'LINE  ',8('.    :  .',I1,'0'))                     PEM50380
9997  FORMAT (1X,I7,' ',A)                                              PEM50390
9996  REWIND(105)                                                       PEM50400
C                                                                       PEM50410
C*************************                                              PEM50420
C** INITIALIZE DATA.....  **                                           PEM50430
C*************************                                              PEM50440
      TITLE(1)=' '                                                      PEM50450
      TITLE(2)=' '                                                      PEM50460
      BLANK  =' '                                                       PEM50470
      BUG =.FALSE.                                                      PEM50480
      DOUBLE =.FALSE                                                    PEM50490
      IBUG=0                                                            PEM50500
      MAXEDF=24                                                         PEM50510
      MD(1)=1                                                           PEM50520
      DO 1 I=2,(MAXEDF+1)                                               PEM50530
    1   MD(I)=MD(I-1)+(I-1)                                             PEM50540
      IZ=0                                                              PEM50550
C                                                                       PEM50560
   20 READ(105,*,END=1000) OPTION                                       PEM50570
      RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                    PEM50580
C                                                                       PEM50590
C*************************                                              PEM50600
C** DEFINE THE STRUCTURE ***                                           PEM50610
C*************************                                              PEM50620
      IF (INDEX(OPTION,'STR').NE.0)  THEN                               PEM50630
C -----  -----  - ---  --------------                                    PEM50640
            IZ=1                                                        PEM50650
            TITLE(1)=' '                                                PEM50660
            TITLE(2)=' '                                                PEM50670
C ---------------------------------                                     PEM50680
            GRAV=386.4                                                  PEM50690
            READ(105,*,END=1000) TITLE(1)                               PEM50700
            CALL TIDAE(TIME1,DATE1)                                     PEM50710
            WRITE(TITLE(1)(80:),45) TIME1,DATE1                         PEM50720
            RATIZ=100 *REAL(IZ)/REAL(MAXZ)                              PEM50730
            WRITE(*,52)' BUILDING STRUCTURAL MODEL ----- -- --  ---- *', PEM50740
     &                  IZ,RATIZ                                        PEM50750
```

```
          WRITE(*,51) TITLE(1)(1:70)                                    PEM50760
          CALL STRUCT                                                   PEM50770
          IZREL-IZ                                                      PEM50780
          RATIZ-100.*REAL(IZ)/REAL(MAXZ)                                PEM50790
C                                                                       PEM50800
C---------------------------------                                      PEM50810
C-- DEFINE THE LOADING AND SOLVE --                                     PEM50820
C---------------------------------                                      PEM50830
      ELSE IF (INDEX(OPTION,'SOL') NE.0)  THEN                          PEM50840
          READ(105,*,END=1000) TITLE(2)                                 PEM50850
          CALL TIDAE(TIME1,DATE1)                                       PEM50860
          WRITE(TITLE(2)(80:),45) TIME1,DATE1                           PEM50870
          WRITE(*,52)' SOLVING STRUCTURAL MODEL - ' ,PEM50880
     &                  IZ,RATIZ                                        PEM50890
          WRITE(*,51) TITLE(1)(1:70)                                    PEM50900
          WRITE(*,51) TITLE(2)(1:70)                                    PEM50910
          CALL SOLN(OPTION)                                             PEM50920
          RATIZ-100.*REAL(IZ)/REAL(MAXZ)                                PEM50930
C                                                                       PEM50940
C---------------------------------                                      PEM50950
C-- STOP... END OF PROGRAM       --                                     PEM50960
C---------------------------------                                      PEM50970
      ELSE IF (INDEX(OPTION,'STOP').NE.0)  THEN                         PEM50980
          WRITE(*,52)' NORMAL STOP ' ,PEM50990
     &                  IZ,RATIZ                                        PEM51000
          STOP                                                          PEM51010
C                                                                       PEM51020
C---------------------------------                                      PEM51030
C-- READ RESULTS FROM INPUT FILE ---                                    PEM51040
C---------------------------------                                      PEM51050
      ELSE IF (INDEX(OPTION,'READ').NE.0)  THEN                         PEM51060
  140     CALL GETINT(OPTION,'UNIT-',IUNIT,0,.TRUE.,.TRUE. )            PEM51070
          CALL GETINT(OPTION, 'INC-',INC  ,1,.TRUE., FALSE.)            PEM51080
          IF (IUNIT.GT.0)  THEN                                         PEM51090
            WRITE(*,52) ' READING DATA FROM FILE ' ,PEM51100
     &                  IZ,RATIZ                                        PEM51110
            IOPT=4                                                      PEM51120
            CALL DMPDAT(IOPT,IWRITE,TO,DT)                              PEM51130
            GO TO 140                                                   PEM51140
          ENDIF                                                         PEM51150
          RATIZ-100.*REAL(IZ)/REAL(MAXZ)                                PEM51160
C                                                                       PEM51170
C-----------------------                                                PEM51180
C-- SET BUG OPTION     --                                               PEM51190
C-----------------------                                                PEM51200
      ELSE IF (TEST(OPTION,'BUG-',.FALSE.))  THEN                       PEM51210
          CALL GETCHR(OPTION,'BUG-',INPUT,.TRUE.,.FALSE )               PEM51220
          IBUG=0                                                        PEM51230
          WRITE(OPTION,160) INPUT(1:70)                                 PEM51240
          DO 150 I=0,23                                                 PEM51250
            IF (TEST(INPUT,CHR(I),.FALSE.)) IBUG=IBSET(IBUG,I)          PEM51260
  150     CONTINUE                                                      PEM51270
  160     FORMAT ('BUG-',A)                                             PEM51280
          WRITE(*,52)' SET BUG OPTION: ' ,PEM51290
     &                  IZ,RATIZ                                        PEM51300
          WRITE(*,51)  OPTION(1:70)                                     PEM51310
          WRITE(108,51)  'SET BUG OPTION  '                             PEM51320
          WRITE(108,51)  OPTION(1:70)                                   PEM51330
C                                                                       PEM51340
C-----------------------                                                PEM51350
C-- SET BUG OPTION     --                                               PEM51360
C-----------------------                                                PEM51370
      ELSE IF (INDEX(OPTION,'BUG') NE.0)  THEN                          PEM51380
          WRITE(*,51) ('SET BUG OPTION. '//OPTION(1:54) )               PEM51390
          BUG=.TRUE                                                     PEM51400
          IF (INDEX(OPTION,'NOBUG').NE.0) BUG=.FALSE.                   PEM51410
C                                                                       PEM51420
C-------------------------------------                                  PEM51430
C-- RELEASE MEMORY FROM LAST SOLUTION --                                PEM51440
C-------------------------------------                                  PEM51450
      ELSE IF (INDEX(OPTION,'RELEASE').NE.0)  THEN                      PEM51460
          WRITE(108,52)' RELEASING MEMORY ' ,PEM51470
          WRITE(*,52)' RELEASING MEMORY ' ,PEM51480
          WRITE(108,52)' RELEASING MEMORY ' ,PEM51490
          IF (INDEX(OPTION,'ELEMENT').NE.0)  THEN                       PEM51500
          WRITE(108,52)' RESETING ELEMENT FORCES ' ,PEM51510
          WRITE(108,52)' ' ,PEM51520
            DO 170 IELNO=1,NELMT                                        PEM51530
              LSTTYP=0                                                  PEM51540
              HEAD =.FALSE.                                             PEM51550
  170         CALL ELELIB                                               PEM51560
     &      (13,LSTTYP,.FALSE.,IREL,HEAD,INPUT,EESE,EPSE,DAMAGE,DUCFAG,  PEM51570
     &       IELNO,IELDOF,XGDOF,PGEOM,2 ,AXIAL,IZDDSP,IZDLOA,MSTOR)      PEM51580
          ELSE                                                          PEM51590
          WRITE(108,52)' ' ,PEM51600
          ENDIF                                                         PEM51610
          IZ-IZREL                                                      PEM51620
            IZACC =0                                                    PEM51630
            IZDACC=0                                                    PEM51640
            IZDDSP=0                                                    PEM51650
            IZDISP=0                                                    PEM51660
            IZDLOA=0                                                    PEM51670
            IZDVEL=0                                                    PEM51680
            IZFUE =0                                                    PEM51690
            IZLOAD=0                                                    PEM51700
            IZVEL =0                                                    PEM51710
          RATIZ=100.*REAL(IZ)/REAL(MAXZ)                                PEM51720
C                                                                       PEM51730
C-------------------------------------                                  PEM51740
C-- SAVE MEMORY FOR A RESTART       --                                  PEM51750
C-------------------------------------                                  PEM51760
      ELSE IF (INDEX(OPTION,'SAVE') NE.0)  THEN                         PEM51770
          WRITE(*,52)' SAVING DATA ' ,PEM51780
     &                  IZ,RATIZ                                        PEM51790
          WRITE(*,51) TITLE(1)(1:70)                                    PEM51800
          WRITE(*,51) TITLE(2)(1:70)                                    PEM51810
          WRITE(108,200)                                                PEM51820
  200     FORMAT(5X,'******** WRITING DATA TO UNIT FT09 ********')      PEM51830
          OPEN(UNIT=9,ACCESS='SEQUENTIAL',STATUS='UNKNOWN',             PEM51840
     &         FORM='UNFORMATTED')                                      PEM51850
          WRITE(9)   TITLE                                              PEM51860
          WRITE(9)   IZ    ,IZREL ,                                     PEM51870
     &      NDOF  ,NCORD ,NFREE ,NREST ,MAXNOD,NMAT  ,NELMT ,NNODE ,    PEM51880
     &      NLOAD ,NSOLN ,MAXEDF,MD,NCOS,MAXELD,NELD,                   PEM51890
     &      IZID  ,IZIDOF,IZCORD,IZCOS ,IZCNST,IZMAT ,IZELE ,           PEM51900
     &      IZJFLG,IZJCOS,                                              PEM51910
     &      IZKE  ,IZLM  ,IZSTIF,IZMD  ,IZLOAD,IZDISP,IZIELD,IZELD      PEM51920
          DO 210 J=1,IZ,256                                             PEM51930
             X-J+255                                                    PEM51940
             K-MIN(X,IZ)                                                PEM51950
  210        WRITE(9)  (Z(I),I-J,K)                                     PEM51960
          CLOSE (UNIT=9)                                                PEM51970
          RATIZ-100 *REAL(IZ)/REAL(MAXZ)                                PEM51980
C                                                                       PEM51990
C-------------------------------------                                  PEM52000
C-- RESTART PROGRAM                 --                                  PEM52010
C-------------------------------------                                  PEM52020
          WRITE(*,52)' RETREVING DATA ' ,PEM52030
     &                  IZ,RATIZ                                        PEM52040
      ELSE IF (INDEX(OPTION,'RESTART').NE.0)  THEN                      PEM52050
          WRITE(108,300)                                                PEM52060
  300     FORMAT(5X,'******** READING DATA FROM UNIT FT09 ********')    PEM52070
          OPEN(UNIT=9,ACCESS='SEQUENTIAL',STATUS='OLD',                 PEM52080
     &         FORM='UNFORMATTED')                                      PEM52090
          READ  (9)   TITLE                                             PEM52100
          READ  (9)   IZ    ,IZREL ,                                    PEM52110
     &      NDOF  ,NCORD ,NFREE ,NREST ,MAXNOD,NMAT  ,NELMT ,NNODE ,    PEM52120
     &      NLOAD ,NSOLN ,MAXEDF,MD,NCOS,MAXELD,NELD,                   PEM52130
     &      IZID  ,IZIDOF,IZCORD,IZCOS ,IZCNST,IZMAT ,IZELE ,           PEM52140
     &      IZJFLG,IZJCOS,                                              PEM52150
     &      IZKE  ,IZLM  ,IZSTIF,IZMD  ,IZLOAD,IZDISP,IZIELD,IZELD      PEM52160
          DO 310 J=1,IZ,256                                             PEM52170
             X-J+255                                                    PEM52180
             K-MIN(X,IZ)                                                PEM52190
  310        READ  (9)  (Z(I),I-J,K)                                    PEM52200
          CLOSE (UNIT=9)                                                PEM52210
          WRITE(*,52)' DATA RETREIVED ' ,PEM52220
     &                  IZ,RATIZ                                        PEM52230
          WRITE(*,51) TITLE(1)(1:70)                                    PEM52240
          WRITE(*,51) TITLE(2)(1:70)                                    PEM52250
          RATIZ-100.*REAL(IZ)/REAL(MAXZ)                                PEM52260
C                                                                       PEM52270
C---------------------------------                                      PEM52280
C-- INVALID OPTION               --                                     PEM52290
C---------------------------------                                      PEM52300
      ELSE                                                              PEM52310
          IF (TEST(OPTION,'NOECHO',.FALSE.) ) GO TO 20                  PEM52320
          IF (OPTION.EQ.BLANK) GO TO 20                                 PEM52330
          WRITE(108,30) OPTION                                          PEM52340
          WRITE(*,51) ('INVALID OPTION- '//OPTION(1:54))                PEM52350
      ENDIF                                                             PEM52360
      GO TO  20                                                         PEM52370
C                                                                       PEM52380
   45 FORMAT(' TIME: ',A,' DATE: ',A)                                   PEM52390
   50 FORMAT (A1,' STRUCTURE   : ',A,/1X,' SOLUTION    : ',A,/)        PEM52400
   51 FORMAT (1X,A)                                                     PEM52410
   52 FORMAT (1X,'----',A45,'--',1,' IZ=',I7,' MEM=',F6.3,'M')          PEM52420
   30 FORMAT ('1 INVALID OPTION- ',A)                                   PEM52430
  110 FORMAT (' ZMATRX(',I6,'):',I15,1P,G20.10,:,G20.10)                PEM52440
 1000 STOP                                                              PEM52450
      END                                                               PEM52460
C                                                                       TIDA0010
C ----- TAKE PRESENT TIME AND DATE ----                                 TIDA0020
C                                                                       TIDA0030
      SUBROUTINE TIDAE(TIME1,DATE1)                                     TIDA0040
       INTEGER*2 IYR,IMON,IDAY,IHR,IMIN,ISEC,I100TH                     TIDA0050
       CHARACTER*8 TIME1,DATE1                                          TIDA0060
       CALL GETDAT(IYR,IMON,IDAY)                                       TIDA0070
       IYR=IYR-1900                                                     TIDA0080
       CALL GETTIM(IHR,IMIN,ISEC,I100TH)                                TIDA0090
       WRITE(DATE1,'(i2,1h/,i2,1h/,i2)') IMON,IDAY,IYR                  TIDA0100
       WRITE(TIME1,'(i2,1h:,i2,1h:,i2)') IHR,IMIN,ISEC                  TIDA0110
      RETURN                                                            TIDA0120
      END                                                               TIDA0130
C                                                                       SOLN0010
C-----------------------------------------------------------------------SOLN0020
C                                                                       SOLN0030
      SUBROUTINE SOLN(OPTION)                                           SOLN0040
      CHARACTER*80 OPTION                                               SOLN0050
$INCLUDE:'ZCOMN'                                                        SOLN0060
      STEPID=' '                                                        SOLN0070
C                                                                       SOLN0080
C---------------------------                                            SOLN0090
C- DETERMINE SOLUTION NO. --                                            SOLN0100
C---------------------------                                            SOLN0110
      CALL GETINT(OPTION,'SOL',NSOLN,0,.TRUE.,.FALSE.)                  SOLN0120
C                                                                       SOLN0130
C-----------------------                                                SOLN0140
C- CHOOSE THE SOLUTION --                                               SOLN0150
C-----------------------                                                SOLN0160
C     IF (NSOLN.EQ.1) THEN                                              SOLN0170
C          CALL SOL01                                                   SOLN0180
C     ELSE IF (NSOLN.EQ.2) THEN                                         SOLN0190
C          NLOAD=1                                                      SOLN0200
C          MAXELD=0                                                     SOLN0210
C          CALL SOL02                                                   SOLN0220
C     ELSE IF (NSOLN.EQ.3) THEN                                         SOLN0230
C          NLOAD=1                                                      SOLN0240
C          MAXELD=0                                                     SOLN0250
C          CALL SOL03                                                   SOLN0260
C     ELSE IF (NSOLN.EQ.4) THEN                                         SOLN0270
C          NLOAD=1                                                      SOLN0280
C          CALL SOL04                                                   SOLN0290
C     ELSE IF (NSOLN.EQ.5) THEN                                         SOLN0300
      IF (NSOLN.EQ.5) THEN                                              SOLN0310
          NLOAD=1                                                       SOLN0320
          MAXELD=0                                                      SOLN0330
          CALL SOL05                                                    SOLN0340
      ELSE                                                              SOLN0350
          WRITE(108,*) 'INVALID SOLN#',NSOLN                            SOLN0360
          STOP 'INVALID SOLUTION NUMBER'                                SOLN0370
      ENDIF                                                             SOLN0380
   30 RETURN                                                            SOLN0390
      END                                                               SOLN0400
C                                                                       MSOL0010
C-----------------------------------------------------------------------MSOL0020
C  MATRIX SOLUTION OF EQUATION AX-P, WHERE THE UPPER TRIANGULAR MATRIX  MSOL0030
C  OF THE SYMMETRIC MATRIX A IS STORED BY SKYLINES. P IS STORED AS A    MSOL0040
C  FULL MATRIX.  THE MATRIX P IS REPLACED BY X AFTER BACK SUBSTITUTION  MSOL0050
C  IS COMPLEATE.  BOTH A AND P ARE ALTERED.....                         MSOL0060
C  GUYAN OPTION. IF GUYAN IS  TRUE. MASS MATRIX IS ALSO CONDENSED       MSOL0070
C-----------------------------------------------------------------------MSOL0080
      SUBROUTINE MSOLL(IOPT,PRINT,N,IMD,LROW,NLC,L1,L2,MD,A,P,FACTOR,   MSOL0090
     &                 GUYAN,MASS,MDMASS,IMDMAS,KGCOND,XG,MDKG,IMDKG)   MSOL0100
      REAL   MASS(IMDMAS),XG(IMDKG)                                     MSOL0110
      LOGICAL GUYAN,PRINT,KGCOND                                        MSOL0120
      DIMENSION A(IMD),FACTOR(N),P(LROW,NLC),MD(N+1)                    MSOL0130
      DIMENSION MDMASS(N+1),MDKG(N+1)                                   MSOL0140
C                                                                       MSOL0150
      IJ(I,J)=MD(J)+J-I                                                 MSOL0160
      IJM(I,J)=MDMASS(J)+J-I                                            MSOL0170
      IJXG(I,J)=MDKG(J)+J-I                                             MSOL0180
C- ------------------------------------------------------------ --------MSOL0190
C   MATRIX STORAGE SCHEME.                                              MSOL0200
C  THE MATRIX A IS BANDED AND SYMMETRIC. THUS ONLY THE SKYLINE ABOVE     MSOL0210
C   AND THE MAIN DIAGONAL NEEDS TO BE STORED.                           MSOL0220
C   THE MATRIX A IS STORED IN A LINEAR ARRAY BY COLUMNS.  THE VALUE ON  MSOL0230
C   THE MAIN DIAGONAL IS STORED IN THE LINEAR ARRAY FIRST.              MSOL0240
C   ADDRESSES OF THE LINEAR ARRAY ELEMENTS CONTAINING THE MAIN DIAGONAL MSOL0250
C   TERMS ARE STORED IN MD.                                             MSOL0260
C   EXAMPLE: LET B BE THE 5X5 FULL SYMETRIC MATRIX BELOW. .             MSOL0270
C        | 1 | 3 |   |   | 12 |                                        MSOL0280
C        |   | 2 | 5 |   | 11 |   NOTE. B(1,3),B(1,4) AND B(2,4) ARE ZERO. MSOL0290
C   B=   |   |   | 4 | 7 | 10 |                                        MSOL0300
C        |   | SYMM. | 6 | 9 |  MD= 1, 2, 4, 6, 8, 13                 MSOL0310
C        |   |       |   | 8 |                                        MSOL0320
C   THUS, B(3,3)=A( MD(3) )= A(4)                                       MSOL0330
C         B(I,J)=A( MD(J)+J-I ) IF J>I AND (MD(J)+J-I)<MD(J-1)          MSOL0340
C               =0              IF J>I AND (MD(J)+J-I)>MD(J-1)        MSOL0350
C   MATRIX P IS NOT SYMMETRIC, AND IS STORED AS A FULL MATRIX           MSOL0360
C- ------------------------------------------------------------ --------MSOL0370
C   VARIABLE TABLE.                                                     MSOL0380
C   IOPT  - OPTION NUMBER                                               MSOL0390
C   N     - NUMBER OF DEGREES OF FREEDOM OF A MATRIX                    MSOL0400
C   NLC   - NUMBER OF LOAD CASES FOR THE P MATRIX                       MSOL0410
C   L1    - FIRST ROW TO BE WORKED WITH                                 MSOL0420
C   L2    - LAST ROW TO BE WORKED WITH                                  MSOL0430
C   MD    - ADDRESS OF THE MAIN DIAGONAL TERMS OF A                     MSOL0440
C   P     - LAOD MATRIX ON INPUT, SOLN (X) OF AX-P ON COMPLEATION OF    MSOL0450
C           IOPT=4                                                      MSOL0460
C   FACTOR- TEMPORARY STORAGE MATRIX                                    MSOL0470
C   GUYAN - GUYAN REDUCTION FLAG FOR MASS MATRIX                        MSOL0480
C   MASS  - MASS MATRIX                                                 MSOL0490
C   MDMASS- MASS MATRIX ADDRESSES OF MAIN DIAGONAL ELEMENTS             MSOL0500
C   KGCOND- FLAG FOR CONDENSING OUT KG                                  MSOL0510
C   XG    - GEOMETRIC STIFFNESS MATRIX                                  MSOL0520
C   MDKG  - KG MATRIX ADDRESSES OF MAIN DIAGONAL ELEMENTS               MSOL0530
C   L     - ROW NUMBER FOR ELIMINATION                                  MSOL0540
C   I     - ROW NUMBER                                                  MSOL0550
C   J     - COLUMN NUMBER                                               MSOL0560
C- ------------------------------------------------------------ --------MSOL0570
C                                                                       MSOL0580
      L1=MAX(L1,1)                                                      MSOL0590
      L2=MIN(L2,N)                                                      MSOL0600
C                                                                       MSOL0610
      GO TO (10,110,210,310) IOPT                                       MSOL0620
C                                                                       MSOL0630
   10 CONTINUE                                                          MSOL0640
C--- ----- IOPT-1, REDUCE A AND P -- ----- ------------ ----------------MSOL0650
C         GAUSSIAN ELIMINIATION IS USED TO REDUCE THE A AND P MATRICIESMSOL0660
C         FROM ROW L1 TO ROW L2   IF L1 IS NOT 1, IT IS ASSUMED THAT   MSOL0670
C         A AND P ARE ALLREADY REDUCED FROM 1 TO L1 ..                  MSOL0680
```

```
C-----------------------------------------------------------------------MSOL0690
      IF (PRINT) THEN                                                   MSOL0700
        WRITE(108,500) 'IOPT=1, REDUCE STIFFNESS AND LOADS',L1,L2,N     MSOL0710
        CALL LMATRX(A,MD,N,IMD ,'INPUT STIFFNESS')                      MSOL0720
        CALL WMATRX(P,N ,NLC,'INPUT LOAD')                              MSOL0730
        IF (GUYAN ) CALL LMATRX(MASS,MDMASS,N,IMDMAS,'INPUT MASS')      MSOL0740
        IF (KGCOND) CALL LMATRX(KG  ,MDKG  ,N,IMDKG ,'INPUT KG  ')     MSOL0750
      ENDIF                                                             MSOL0760
C                                                                       MSOL0770
      DO 100 L=L1,L2                                                    MSOL0780
        IF (L.EQ.N) GO TO 100                                           MSOL0790
C                                                                       MSOL0800
C------- TEST FOR ZERO PIVOT. . . . . .                                 MSOL0810
        IF (A(MD(L)).EQ.0) THEN                                         MSOL0820
          WRITE(108,2000) L,MD(L),A(MD(L))                              MSOL0830
          STOP                                                          MSOL0840
        ENDIF                                                           MSOL0850
C                                                                       MSOL0860
C---- -- CALCULATE THE FACTORS EACH ROW IS MULTIPLIED BY                MSOL0870
        DO 30 I=(L+1),N                                                 MSOL0880
          LI=IJI(L,I)                                                   MSOL0890
          IF ( LI.LT.MD(I+1) ) THEN                                     MSOL0900
           FACTOR(I)=A(LI) /A( MD(L) )                                  MSOL0910
          ELSE                                                          MSOL0920
           FACTOR(I)=0                                                  MSOL0930
          ENDIF                                                         MSOL0940
   30     CONTINUE                                                      MSOL0950
C                                                                       MSOL0960
C------- REDUCE A BY COLUMNS (J) EACH ROW (I). .                        MSOL0970
        DO 40 J=(L+1),N                                                 MSOL0980
          LJ=IJI(L,J)                                                   MSOL0990
          IF (LJ.GE.MD(J+1)) GO TO 40                                   MSOL1000
          DO 35 I=(L+1),J                                               MSOL1010
            IJ=IJI(I,J)                                                 MSOL1020
            A(IJ)=A(IJ)-FACTOR(I)*A(LJ)                                 MSOL1030
   35     CONTINUE                                                      MSOL1040
   40   CONTINUE                                                       MSOL1050
C                                                                       MSOL1060
C------- REDUCE P BY COLUMNS (J) EACH ROW (I). .                        MSOL1070
        DO 50 J=1,NLC                                                   MSOL1080
          IF (P(L,J).EQ.0) GO TO 50                                     MSOL1090
          DO 45 I=(L+1),N                                               MSOL1100
   45       P(I,J)=P(I,J)-FACTOR(I)*P(L,J)                             MSOL1110
   50   CONTINUE                                                       MSOL1120
C                                                                       MSOL1130
C------- REDUCE MASS MATRIX BY GUYAN REDUCTION                          MSOL1140
      IF (GUYAN) THEN                                                   MSOL1150
C                                                                       MSOL1160
C ........ M22=M22 - K21/K11 * M11 * 1/K11 * K12                       MSOL1170
        RM=MASS(MDMASS(L))                                              MSOL1180
        IF (RM.NE.0) THEN                                               MSOL1190
             DO 60 J=L,N                                                MSOL1200
                  IF (FACTOR(J).EQ.0) GO TO 60                          MSOL1210
                  I1=MAX(L+1,J-(MDMASS(J+1)-MDMASS(J)) )                MSOL1220
                  DO 55 I=I1,J                                          MSOL1230
                    IJ=IJM(I,J)                                         MSOL1240
   55               MASS(IJ)=MASS(IJ)+FACTOR(I)*RM*FACTOR(J)           MSOL1250
   60           CONTINUE                                               MSOL1260
        ENDIF                                                           MSOL1270
C                                                                       MSOL1280
C. ........ M22=M22 - K21/K11 * M12                                     MSOL1290
        DO 66 J=L+1,N                                                   MSOL1300
             LJ=IJM(L,J)                                                MSOL1310
             IF (LJ.GE.MDMASS(J+1)) GO TO 66                            MSOL1320
             IF (MASS(LJ).EQ.0) GO TO 66                                MSOL1330
             DO 65 I=L+1,J                                              MSOL1340
               IJ=IJM(I,J)                                              MSOL1350
               IF (IJ.LT.MDMASS(J+1))                                   MSOL1360
     &           MASS(IJ)=MASS(IJ)-FACTOR(I)*MASS(LJ)                   MSOL1370
   65        CONTINUE                                                   MSOL1380
   66   CONTINUE                                                       MSOL1390
C                                                                       MSOL1400
C  ..  ... M22=M22   M21 * 1/K11 * K12                                  MSOL1410
        DO 70 J=L+1,N                                                   MSOL1420
             IF (FACTOR(J).EQ.0) GO TO 70                               MSOL1430
             DO 69 I=L+1,J                                              MSOL1440
               LI=IJM(L,I)                                              MSOL1450
               IF (LI.GE.MDMASS(I+1)) GO TO 69                          MSOL1460
               IJ=IJM(I,J)                                              MSOL1470
               IF (IJ.LT.MDMASS(J+1))                                   MSOL1480
     &           MASS(IJ)=MASS(IJ)-FACTOR(J)*MASS(LI)                   MSOL1490
   69        CONTINUE                                                   MSOL1500
   70   CONTINUE                                                       MSOL1510
      ENDIF                                                             MSOL1520
C                                                                       MSOL1530
C------- REDUCE GEOMETRIC STIFFNESS MATRIX                              MSOL1540
      IF (KGCOND) THEN                                                  MSOL1550
C                                                                       MSOL1560
C  ....   KG22=KG22 - K21/K11 * KG11 * 1/K11 * KG12                     MSOL1570
        RKG=KG(MDKG(L))                                                 MSOL1580
        IF (RKG.NE.0) THEN                                              MSOL1590
             DO 80 J=L+1,N                                              MSOL1600
                  IF (FACTOR(J).EQ.0) GO TO 80                          MSOL1610
                  I1=MAX(L+1,J-(MDKG(J+1)-MDKG(J)) )                    MSOL1620
                  DO 75 I=I1,J                                          MSOL1630
                    IJ=IJKG(I,J)                                        MSOL1640
   75               KG(IJ)=KG(IJ)+FACTOR(I)*RKG*FACTOR(J)              MSOL1650
   80           CONTINUE                                               MSOL1660
        ENDIF                                                           MSOL1670
C                                                                       MSOL1680
C ....... . KG22=KG22 - K21/K11 * KG12                                  MSOL1690
        DO 86 J=L+1,N                                                   MSOL1700
             LJ=IJKG(L,J)                                               MSOL1710
             IF (LJ.GE.MDKG(J+1)) GO TO 86                              MSOL1720
             IF (KG(LJ).EQ.0) GO TO 86                                  MSOL1730
             DO 85 I=L+1,J                                              MSOL1740
                  IJ=IJKG(I,J)                                          MSOL1750
                  IF (IJ.LT.MDKG(J+1))                                  MSOL1760
     &              KG(IJ)=KG(IJ)-FACTOR(I)*KG(LJ)                      MSOL1770
   85        CONTINUE                                                   MSOL1780
   86   CONTINUE                                                       MSOL1790
C                                                                       MSOL1800
C.........  KG22=KG22 - KG21 * 1/K11 * K12                             MSOL1810
        DO 90 J=L+1,N                                                   MSOL1820
             IF (FACTOR(J) EQ.0) GO TO 90                               MSOL1830
             DO 89 I=L+1,J                                              MSOL1840
                  LI=IJKG(L,I)                                          MSOL1850
                  IF (LI GE MDKG(I+1)) GO TO 89                         MSOL1860
                  IJ=IJKG(I,J)                                          MSOL1870
                  IF (IJ.LT.MDKG(J+1))                                  MSOL1880
     &              KG(IJ)=KG(IJ)-FACTOR(J)*KG(LI)                      MSOL1890
   89        CONTINUE                                                   MSOL1900
   90   CONTINUE                                                       MSOL1910
C                                                                       MSOL1920
      ENDIF                                                             MSOL1930
C                                                                       MSOL1940
  100 CONTINUE                                                          MSOL1950
      IF (PRINT) THEN                                                   MSOL1960
        CALL LMATRX(A,MD,N,IMD,'REDUCED STIFFNESS')                     MSOL1970
        CALL WMATRX(P,N ,NLC,'REDUCED LOAD')                            MSOL1980
        IF (GUYAN)  CALL LMATRX(MASS,MDMASS,N,IMDMAS,'REDUCED MASS')    MSOL1990
        IF (KGCOND) CALL LMATRX(KG  ,MDKG  ,N,IMDKG ,'REDUCED KG')     MSOL2000
      ENDIF                                                             MSOL2010
      RETURN                                                            MSOL2020
C                                                                       MSOL2030
  110 CONTINUE                                                          MSOL2040
C-------- IOPT=2, REDUCE P ONLY ------------------------------------------MSOL2050
C          GAUSSIAN ELIMINIATION IS USED TO REDUCE THE P MATRIX         MSOL2060
C          FROM ROW L1 TO ROW L2   IF L1 IS NOT 1, IT IS ASSUMED THAT   MSOL2070
C          A AND P ARE ALLREADY REDUCED FROM 1 TO L1.                   MSOL2080
C-----------------------------------------------------------------------MSOL2090
      IF (PRINT) THEN                                                   MSOL2100
        WRITE(108,500) 'IOPT=2, REDUCE LOADS ONLY',L1,L2,N              MSOL2110
        CALL LMATRX(A,MD,N,MD(N+1),'REDUCED STIFFNESS')                 MSOL2120
        CALL WMATRX(P,N ,NLC,'NEW LOAD MATRIX ')                        MSOL2130
      ENDIF                                                             MSOL2140
C                                                                       MSOL2150
C--- --- LOOP FOR EACH ROW (L) TO REDUCE....                            MSOL2160
      DO 160 L=L1,L2                                                    MSOL2170
C                                                                       MSOL2180
        IF (A(MD(L)).EQ.0) THEN                                         MSOL2190
          WRITE(108,2000) L,MD(L),A(MD(L))                              MSOL2200
          STOP                                                          MSOL2210
        ENDIF                                                           MSOL2220
C                                                                       MSOL2230
C       LOOP FOR EACH COLUMN (J)                                        MSOL2240
        DO 150 J=1,NLC                                                  MSOL2250
          IF (P(L,J).EQ.0) GO TO 150                                    MSOL2260
          DO 120 I=(L+1),N                                              MSOL2270
            LI=IJI(L,I)                                                 MSOL2280
            IF (LI.LT.MD(I+1)) P(I,J)=P(I,J)-A(LI)*P(L,J)/A(MD(L))      MSOL2290
  120       CONTINUE                                                    MSOL2300
  150   CONTINUE                                                        MSOL2310
  160   CONTINUE                                                        MSOL2320
      RETURN                                                            MSOL2330
C                                                                       MSOL2340
  210 CONTINUE                                                          MSOL2350
C------- IOPT=3, BACK SUBSTUTION ------------------------------------------MSOL2360
C          BACK SUBSTUTION IS USED TO SOLVE FOR X IN SX=V, WHERE S IS   MSOL2370
C          THE REDUCE UPPER TRIANGULAR FACTOR OF THE A MATRIX, AND V    MSOL2380
C          IS THE REDUCED FORRM OF THE P MATRIX.  THE RESULTS (X) ARE   MSOL2390
C          STORED IN P                                                  MSOL2400
C          BACK SUBSTUTION ONLY TAKES PLACE BETWEEN ROWS L1 AND L2.     MSOL2410
C-----------------------------------------------------------------------MSOL2420
      IF (PRINT) THEN                                                   MSOL2430
        WRITE(108,500) 'IOPT=3, BACK SUBSTUTION',L1,L2,N                MSOL2440
        CALL LMATRX(A,MD,N,MD(N+1),'REDUCED STIFFNESS')                 MSOL2450
        CALL WMATRX(P,N ,NLC,'REDUCED LOAD')                            MSOL2460
      ENDIF                                                             MSOL2470
C                                                                       MSOL2480
C------ LOOP FOR EACH ROW L                                             MSOL2490
C       CALCULAGE X FOR EACH LOAD CASE, STORE IN P                      MSOL2500
        DO 250 J=L2,L1,-1                                               MSOL2510
        IF (A(MD(J)).EQ.0) THEN                                         MSOL2520
          WRITE(108,2000) J,MD(J),A(MD(J))                              MSOL2530
          STOP                                                          MSOL2540
        ENDIF                                                           MSOL2550
        DO 215 LC=1,NLC                                                 MSOL2560
  215     P(J,LC)=P(J,LC)/A(MD(J))                                      MSOL2570
        IF (J.GT.1) THEN                                                MSOL2580
          IJ=MD(J+1)                                                    MSOL2590
          NI=J-(MD(J+1)-MD(J)) +1                                       MSOL2600
          DO 220 I=NI,J-1,+1                                            MSOL2610
            IJ=IJ+1                                                     MSOL2620
            DO 220 LC=1,NLC                                             MSOL2630
              P(I,LC)=P(I,LC)-P(J,LC)*A(IJ)                             MSOL2640
  220     CONTINUE                                                      MSOL2650
        ENDIF                                                           MSOL2660
  250   CONTINUE                                                        MSOL2670
      IF (PRINT) CALL WMATRX(P,N ,NLC,'DISPLACEMENT')                   MSOL2680
C                                                                       MSOL2690
      RETURN                                                            MSOL2700
C                                                                       MSOL2710
  310 CONTINUE                                                          MSOL2720
C------- IOPT=4, REDUCED MULTIPLICATION ------------------------------------MSOL2730
C          SOLVE FOR V IN S*D=P WHERE S HAS BEEN REDUCED BY GAUSSIAN    MSOL2740
C          ELIMINATION                                                  MSOL2750
C          THE RESULTING LOAD IS STORED IN P.                           MSOL2760
C          THE DISPLACEMENTS ARE INPUT IN D.                            MSOL2770
C          MULTIPLACATION ONLY TAKES PLACE BETWEEN ROWS L1 AND L2.      MSOL2780
C-----------------------------------------------------------------------MSOL2790
      IF (PRINT) THEN                                                   MSOL2800
        WRITE(108,500) 'IOPT=4, REDUCED MULTIPLACATION',L1,L2,N         MSOL2810
        CALL LMATRX(A,MD,N,MD(N+1),'REDUCED STIFFNESS')                 MSOL2820
        CALL WMATRX(FACTOR,N ,1,'DISPLACEMENT')                         MSOL2830
      ENDIF                                                             MSOL2840
C                                                                       MSOL2850
C-----  ONLY ONE LOAD CASE....                                          MSOL2860
      LC=1                                                              MSOL2870
C       ---- ZERO P ----                                                MSOL2880
        DO 320 J=L1,L2                                                  MSOL2890
          IF (A(MD(J)).EQ.0) THEN                                       MSOL2900
            WRITE(108,2000) J,MD(J),A(MD(J))                            MSOL2910
            STOP                                                        MSOL2920
          ENDIF                                                         MSOL2930
  320     P(J,LC)=0                                                     MSOL2940
C                                                                       MSOL2950
C------------ MULTIPLY S_RED * DISP = P                                 MSOL2960
      DO 340 J=L1,N                                                     MSOL2970
        IC1=J-(MD(J+1)-MD(J)) +1                                        MSOL2980
        IC2=MAX(IC1,L1)                                                 MSOL2990
        DO 340 I=J,IC2,-1                                               MSOL3000
          IJ=IJI(I,J)                                                   MSOL3010
  340     P(I,LC)=P(I,LC)+A(IJ)*FACTOR(J)                               MSOL3020
      IF (PRINT) CALL WMATRX(P,N ,1,'REDUCED LOAD')                     MSOL3030
C                                                                       MSOL3040
C------- UNREDUCE THE LOAD MATRIX  ..                                   MSOL3050
      DO 360 L=L2,L1,-1                                                 MSOL3060
C                                                                       MSOL3070
C       LOOP FOR EACH COLUMN (J)                                        MSOL3080
        DO 350 J=1,NLC                                                  MSOL3090
          IF (P(L,LC).EQ.0) GO TO 350                                   MSOL3100
          DO 345 I=(L+1),N                                              MSOL3110
            LI=IJI(L,I)                                                 MSOL3120
            IF (LI.LT.MD(I+1)) P(I,LC)=P(I,LC)+A(LI)*P(L,LC)/A(MD(L))   MSOL3130
  345       CONTINUE                                                    MSOL3140
  350   CONTINUE                                                        MSOL3150
  360   CONTINUE                                                        MSOL3160
  400   CONTINUE                                                        MSOL3170
C                                                                       MSOL3180
      IF (PRINT) CALL WMATRX(P,N ,1,'LOAD')                             MSOL3190
  500 FORMAT (/2X,A,2X,'L1:',I5,'   L2 ',I5,'   N:',I5)                MSOL3200
 2000 FORMAT (///1X,50('*')/                                            MSOL3210
     &        ' *  ER','ROR IN SUBROUTINE MSOLL',       T51,'*'/       MSOL3220
     &        ' *  PIVOT IS LE 0 -- ---- ROW ',I5,      T51,'*'/       MSOL3230
     &        ' *                  STORAGE LCN ',I5,    T51,'*'/       MSOL3240
     &        ' *                        PIVOT ',1P,G15.6,T51,'*'/       MSOL3250
     &        ' *  SOLUTION IS ABORTED ',               T51,'*'/       MSOL3260
     &        1X,50('*')/)                                            MSOL3270
C                                                                       MSOL3280
      RETURN                                                            MSOL3290
      END                                                               MSOL3300
C                                                                       MSOL3310
C-----------------------------------------------------------------------REAC0020
C                                                                       REAC0030
      SUBROUTINE REACTN(IOPT,PRINT,NNODE,L1,L2,L3,L4,FORCE,DISPL,       REAC0040
     &                  MD,A,KM,NDOF,NLOAD,FACTOR,                      REAC0050
     &                  MAXNOD,IDOF,COORD,ID,TITLE,STEPID,HEAD,         REAC0060
     &                  NCOS,COSINE,JTCOS,JTFLG,SUM)                    REAC0070
      CHARACTER*(*) TITLE(2),STEPID                                     REAC0080
      CHARACTER*15 CD(6)                                                REAC0090
      LOGICAL PRINT,HEAD,FLAG,BTEST                                     REAC0100
      DIMENSION A(KM),MD(NDOF+1),IDOF(MAXNOD,6),G(6)                    REAC0110
      DIMENSION FORCE(NDOF,NLOAD),DISPL(NDOF,NLOAD)                     REAC0120
      DIMENSION ID(MAXNOD),COORD(MAXNOD,3),JTFLG(MAXNOD)                REAC0130
      DIMENSION P(6),SUM(6),COSINE(3,3,NCOS),JTCOS(MAXNOD)              REAC0140
C                                                                       REAC0150
      IF (IOPT.EQ.1) THEN                                               REAC0160
C                                                                       REAC0170
C-----------------------------------------                              REAC0180
C- MODIFY LOADS FOR RESTRAINT DISPLACEMENTS ---                         REAC0190
C-----------------------------------------                              REAC0200
C CONSIDER THE PARTITIONED STIFFNESS WHERE K2 ARE THE RESTRAINT DISPL.  REAC0210
C                                                                       REAC0220
C   |P1|   | K11 | K12 |  |X1|                                         REAC0230
C   |--| = |-----+-----| *|--|                                         REAC0240
C   |P2|   | K21 | K22 |  |X2|                                         REAC0250
C                                                                       REAC0260
C  THE MODIFIED FREE LOAD IS {P1 K12*X2} = K11*X1                       REAC0270
C-------------------------------------------------------------C         REAC0280
C     DO 10 I=L1,L2                                  C  FULL STORAGE    REAC0290
C     DO 10 J=1,NLOAD                                C  EQUIVALENT      REAC0300
```

```
C      DO 10 K=L3,L4                                                    C
C  10    FORCE(I,J)=FORCE(I,J)-K(I,K)*DISPL(K,J)                        C
C-------------------------------------------------------------------C
       DO 20 K=L3,L4
       MDK=MD(K+1)
         DO 20 J=1,NLOAD
         DO 10 I=L2,L1,-1
           IK=MD(K)+K-I
           IF (IK.GE.MDK) GO TO 20
           FIJ=FORCE(I,J)
           AIK=A(IK)
           DKJ=DISPL(K,J)
    10     FORCE(I,J)=FORCE(I,J)-A(IK)*DISPL(K,J)
    20   CONTINUE
         IF (PRINT)
     &   CALL WMATRX(FORCE,NDOF,NLOAD,'MODIFIED LOADS')
C
       ELSE IF (IOPT.EQ.2) THEN
C
C-------------------------
C- SOLVE FOR REACTIONS ---
C-------------------------
C CONSIDER THE PARTITIONED STIFFNESS WHERE X2 ARE THE RESTRAINT DISPL.
C
C   |P1|   | K11 | K12 | |X1|
C   |--| = |-----+-----| |--| * FACTOR
C   |P2|   | K21 | K22 | |X2|
C
C   THE REACTIONS ARE  P2 = P2 + ( K21*X1 + K22*X2 ) * FACTOR
C   NOTE: BOTH P1 AND P2 CONTAIN FIXED END FORCES FROM MEMBER LOADS.
C         THUS, K21*X1 + K22*X2 IS ADDED TO P2 TO GET THE REACTIONS
C-------------------------------------------------------------------C
C      DO 120 I=L3,L4                                               C    FULL STORAGE
C      DO 120 J=1,NLOAD                                             C    MODE EQUIVALENT
CC      FORCE(I,J)=0    * P2 CONTAINS FEF'S!                        C
C      DO 110 K=L1,L2                                               C
C 110   FORCE(I,J)=FORCE(I,J)+K(I,K)*DISPL(K,J)                     C
C      DO 120 K=L3,L4                                               C
C 120   FORCE(I,J)=FORCE(I,J)+K(I,K)*DISPL(K,J)                     C
C-------------------------------------------------------------------C
C
       DO 120 I=L3,L4
       DO 120 J=1,NLOAD
       DO 110 K=L2,L1,-1
         KI=MD(I)+I-K
         IF (KI.GE.MD(I+1)) GO TO 120
         AKI=A(KI)
    110  FORCE(I,J)=FORCE(I,J)+A(KI)*DISPL(K,J)*FACTOR
    120 CONTINUE
C
        DO 140 I=L3,L4
        DO 140 K=L3,L4
          IF (I.LE.K) THEN
             IK=MD(K)+K-I
             MDK=MD(K+1)
          ELSE
             IK=MD(I)+I-K
             MDK=MD(I+1)
          ENDIF
          IF (IK.GE.MDK)  GO TO 140
          AIK=A(IK)
          DO 130 J=1,NLOAD
    130     FORCE(I,J)=FORCE(I,J)+A(IK)*DISPL(K,J)*FACTOR
    140   CONTINUE
C
        IF (PRINT)
     &   CALL WMATRX(FORCE,NDOF,NLOAD,'FINAL LOADS AND REACTIONS')
C
       ELSE IF (IOPT.EQ.3 .OR. IOPT.EQ 4) THEN
          NF=L2
C----------------------------
C- LOOP FOR EACH LOAD CASE  ---
C----------------------------
       DO 300 L=1,NLOAD
C
C-----------------------------
C- PRINT GLOBAL REACTIONS    ---
C-----------------------------
C
       IF (PRINT) THEN
           IF (HEAD) WRITE(108,320) TITLE(1),TITLE(2)
             NREACT=0
             IF (IOPT.NE.4) WRITE(108,321) L,STEPID
             IF (IOPT.EQ 4) WRITE(108,421)
       ENDIF
         IF (IOPT.EQ.3) THEN
           DO 190 I=1,6
    190    SUM(I)=0
         ENDIF
C
       IF ( .NOT.(NCOS.EQ.1 .AND. COSINE(1,1,1).EQ.1 .AND COSINE(2,2,1)
     &      .EQ.1 .AND. COSINE(3,3,1).EQ 1) .AND  IOPT.EQ.4  ) GO TO 279
C
         DO 200 I=1,NNODE
C
C .... .. ..   GET REACTNS IN JOINT COORDINATE SYSTEM
         FLAG=.FALSE.
         DO 209 J=1,6
           IJ=IDOF(I,J)
           IF (IJ.LE.NF .OR. BTEST(JTFLG(I),J+11)) THEN
             F(J)=0
           ELSE
             P(J)=FORCE(IJ,L)
             FLAG= TRUE
           ENDIF
    209    CONTINUE
           IF (.NOT.FLAG) GO TO 200
C
C. ..        .. TRANSFER REACTNS TO GLOBAL COORDINATE SYSTEM
         DO 229 I1=1,3
           I2=I+3
           G(I1)=0
           G(I2)=0
           DO 219 J1=1,3
             J2=J1+3
             G(I1)=G(I1) + COSINE(J1,I1,JTCOS(I))*F(J1)
    219      G(I2)=G(I2) + COSINE(J1,I1,JTCOS(I))*F(J2)
           IF (PRINT) THEN
               WRITE(CD(I1),210) G(I1)
               WRITE(CD(I2),210) G(I2)
           ENDIF
    229    CONTINUE
C
C  . . .     .. SUM REACTIONS  ... .  ..
         IF (IOPT.EQ 3) THEN
           DO 240 I1=1,6
    240      SUM(I1)=SUM(I1) + G(I1)
           SUM(4)= SUM(4) - G(2)*COORD(I,3) + G(3)*COORD(I,2)
           SUM(5)= SUM(5) + G(1)*COORD(I,3)   G(3)*COORD(I,1)
           SUM(6)= SUM(6)   G(1)*COORD(I,2) + G(2)*COORD(I,1)
         ENDIF
C
C     . . . . . PRINT GLOBAL REACTIONS .......
         NREACT=NREACT+1
         IF (PRINT) WRITE(108,220) ID(I),(CD(K),K=1,6)
    200  CONTINUE
         IF (IOPT EQ 3  AND  PRINT ) WRITE(108,230) SUM
    279  IF (IOPT.EQ.4 .AND. PRINT ) WRITE(108,231) SUM
C
C-------------------------------------------------
C- PRINT LOCAL JOINT COORDINATE SYSTEMS REACTIONS  ---
C-------------------------------------------------
C
       IF (.NOT. PRINT ) GO TO 300
       IF (NCOS.EQ 1 .AND  COSINE(1,1,1).EQ.1  AND  COSINE(2,2,1) EQ 1
     &              .AND  COSINE(3,3,1).EQ 1 ) GO TO 300
       IF (HEAD.AND. NREACT.GT.20) WRITE(108,320) TITLE(1),TITLE(2)
         IF (IOPT.NE.4) WRITE(108,322) L,STEPID
         IF (IOPT.EQ.4) WRITE(108,422)
         DO 290 I=1,NNODE
C
C.... ..... GET REACTNS IN JOINT COORDINATE SYSTEM
         FLAG=.FALSE.
         DO 280 J=1,6
           IJ=IDOF(I,J)
           IF (IJ.LE.NF .OR. BTEST(JTFLG(I),J+11)) THEN
             CD(J)=' '
           ELSE
             WRITE(CD(J),210) FORCE(IJ,L)
             FLAG=.TRUE.
           ENDIF
  280    CONTINUE
         IF (.NOT.FLAG) GO TO 290
C
C.......... PRINT REACTNS IN LOCAL COORDINATE SYSTEM . . .
           WRITE(108,221) ID(I),(CD(K),K=1,6),JTCOS(I)
  290   CONTINUE
  300 CONTINUE
C
  210 FORMAT (1P,G15.7)
  220 FORMAT (5X,I5,6A)
  221 FORMAT (5X,I5,6A,I10)
  230 FORMAT (' ',90('-'),/' SUMMATION',6G15.7)
  231 FORMAT (' MAX OF ALL',69('-'),/' GCS SUMM.',6G15.7 )
  320 FORMAT ('1 STRUCTURE... : ',A,/'  SOLUTION..... : ',A)
  321 FORMAT (//'  GCS RESTRAINT REACTIONS,    LOADING #',I5,5X,A,/
     &          ' -------------------------'//6X,'NODE',7X,'FX',
     &          13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ'/)
  322 FORMAT (//'  JCS RESTRAINT REACTIONS,   LOADING #',I5,5X,A,/
     &          ' -------------------------'//6X,'NODE',7X,'FX',
     &          13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ',
     &          6X,'  COSINE #'/)
  421 FORMAT (//'  MAXIMUM GCS RESTRAINT REACTIONS',
     &       5X,'NOTE: MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'/
     &          ' -------------------------------'//6X,'NODE',7X,'FX',
     &         13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ'/)
  422 FORMAT (//'  MAXIMUM JCS RESTRAINT REACTIONS ',
     &       5X,'NOTE: MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'/
     &          ' -------------------------------'//6X,'NODE',7X,'FX',
     &       13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ',
     &       6X,'  COSINE #'/)
C
        ELSE
         WRITE(108,*) 'INVALID OPTION IN SUBROUTINE  REA_CTN, IOPT=',IOPT
        ENDIF
        RETURN
        END
C                                                                      FORM0010
C----------------------------------------------------------------------FORM0020
C                                                                      FORM0030
       SUBROUTINE FORM(IOPT)                                           FORM0040
       CHARACTER  NAME*80,ELNO*130                                     FORM0050
       LOGICAL PRINT,HEAD,FALSE,AXIAL,BTEST,BUG5,BUG57                 FORM0060
$INCLUDE.'ZCOMN'                                                       FORM0070
C                                                                      FORM0080
C.....................................................................FORM0090
C   MATRIX STORAGE SCHEME:                                             FORM0100
C   THE STIFFNESS MATRIX (K) IS BANDED AND SYMMETRIC.                  FORM0110
C   THUS ONLY THE SKYLINE ABOVE THE MAIN DIAGONAL NEEDS TO BE STORED   FORM0120
C   THE MATRIX K IS STORED IN A LINEAR ARRAY BY COLUMNS.               FORM0130
C   THE VALUE ON THE MAIN DIAGONAL IS STORED IN THE LINEAR ARRAY FIRST. FORM0140
C   RELATIVE ADDRESSES OF THE LINEAR ARRAY ELEMENTS CONTAINING THE     FORM0150
C   MAIN DIAGONAL TERMS ARE STORED IN MD.                              FORM0160
C   THE ABSOLUTE ADDRESS OF THE FIRST ELEMENT IN THE K MATRIX IS IZSTIF.FORM0170
C   THE STIFFNESS MATRIX K IS STORED IN THE Z ARRAY, Z(IZSTIF)=K(1,1)  FORM0180
C   THE ABSOLUTE ADDRESS OF THE FIRST ELEMENT IS THE MD MATRIX IS IZMD. FORM0190
C   THE STIFFNESS MATRIX K IS STORED IN THE NZ ARRAY, NZ(IZMD)=MD(1)   FORM0200
C   THUS: IF I LE J (UPPER TRIANGULAR MATRIX)                          FORM0210
C         IF IZMD-1+J < IZMD-1+J+1                                     FORM0220
C              IJ=NZ(IZMD-1+J)+J-I : RELATIVE ADDRESS IF ELEMENT (I,J)FORM0230
C              B(I,J)=Z(IZSTIF-1 + IJ)                                 FORM0240
C         ELSE                                                         FORM0250
C              B(I,J)=0                . (I,J) IS ABOVE THE SKYLINE    FORM0260
C                                                                      FORM0270
C   EXAMPLE: LET B BE THE 5X5 FULL SYMETRIC MATRIX BELOW...            FORM0280
C      | 1 | 3 |   |   | 12 |                                         FORM0290
C      |   | 2 | 5 |   | 11 |  NOTE. B(1,3),B(1,4) AND B(2,4) ARE ZERO. FORM0300
C   B= |   |   | 4 | 7 | 10 |                                         FORM0310
C      |   | SYMM. | 6 |  9 |  MD= 1, 2, 4, 6, 8, 13                  FORM0320
C      |   |   |   |   |  8 |                                         FORM0330
C   SAY IZMD=121,IZSTIF=200                                            FORM0340
C   NZ(121)=1          Z(200)=1                                        FORM0350
C   NZ(122)=2          Z(201)=2                                        FORM0360
C   NZ(123)=4          Z(202)=3                                        FORM0370
C   NZ(124)=6          Z(203)=4                                        FORM0380
C   NZ(125)=8          Z(204)=5                                        FORM0390
C   NZ(126)=13         Z(205)=6                                        FORM0400
C                      Z(206)=7                                        FORM0410
C                      Z(206)=8                                        FORM0420
C                      Z(206)=9                                        FORM0430
C                      Z(206)=10                                       FORM0440
C                      Z(206)=11                                       FORM0450
C                      Z(206)=12                                       FORM0460
C   THUS, FOR B(3,4)   I=3, J=4                                        FORM0470
C            IJ=NZ(IZMD-1+J)+J-I = NZ(121-1+4) +4-3 = 6+4-3 =7         FORM0480
C           B(I,J)=Z(IZSTIF-1 + IJ) = Z(200-1+7) = Z(206) = 7  OK      FORM0490
C   THUS, FOR B(2,4)  I=2, J=4                                         FORM0500
C            IJ=NZ(IZMD-1+J)+J-I = NZ(121-1+4) +4-2 = 6+4-2 =8         FORM0510
C            NZ(IZMD -1+J-1)=NZ(121-1+4-1)=8                           FORM0520
C            SINCE IJ GE NZ(IZMD-1+J-1)                                FORM0530
C           B(I,J)=0  (I,J) IS ABOVE THE SKYLINE....                   FORM0540
C.....................................................................FORM0550
C                                                                      FORM0560
       BUG5 = BTEST(IBUG,5)                                            FORM0570
       BUG57= BTEST(IBUG,7) .AND. BUG5                                 FORM0580
       LSTTYP=0                                                        FORM0590
       PRINT= FALSE                                                    FORM0600
       HEAD = FALSE.                                                   FORM0610
C                                                                      FORM0620
C.....................................................................FORM0630
C IOPT=1 FORM STIFFNESS MATRIX                                         FORM0640
C IOPT=2 FORM GEOMETRIC STIFFNESS MATRIX                               FORM0650
C.....................................................................FORM0660
C                                                                      FORM0670
       IF (BUG5)  WRITE (6,*) ' IN ROUTINE FORM IOPT:',IOPT            FORM0680
C                                                                      FORM0690
   69 FORMAT (1X,A/(' I.',I3,'  MDMASS(I):',I11))                      FORM0700
C                                                                      FORM0710
       GO TO (100,200) IOPT                                            FORM0720
C                                                                      FORM0730
C--------------------------                                            FORM0740
C- ASSEMBLE STIFFNESS  -----                                           FORM0750
C--------------------------                                            FORM0760
  100 CONTINUE                                                         FORM0770
         IST =IZSTIF                                                   FORM0780
         IEND=IZSTIF -NZ(IZMD+NDOF) -1                                 FORM0790
C                                                                      FORM0800
C---------- ZERO STIFFNESS                                             FORM0810
       DO 110 I=IST,IEND                                               FORM0820
  110    Z(I)=0                                                        FORM0830
C                                                                      FORM0840
C------- ---- LOOP FOR EACH ELEMENT                                    FORM0850
       DO 120 IELNO=1,NELMT                                            FORM0860
C                                                                      FORM0870
C------- --- GET STIFFNESS OF EACH ELEMENT                            FORM0880
         FALSE= FALSE.                                                 FORM0890
         CALL ELELIB                                                   FORM0900
     &   (3,LSTTYP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DAMAGE,DUCFAG,     FORM0910
     &   IELNO ,IELDOF,KGDOF,PGEOM,2   ,AXIAL,IEDISP,IZLOAD,MSTOR)    FORM0920
C                                                                      FORM0930
         IF (BUG57) THEN                                               FORM0940
```

```
          WRITE (ELNO,15)  IELNO,(NZ(I+IZLM-1),I=1,MIN(IELDOF,18))      FORM0950
   15     FORMAT ('ELEMENT #',I6,' LM:',18I6)                           FORM0960
          CALL LMATRX(Z(IZKE),MD,IELDOF,MD(IELDOF+1),ELNO)              FORM0970
        ENDIF                                                           FORM0980
C                                                                       FORM0990
      CALL FORML(Z(IZSTIF),NZ(IZMD),NDOF,NZ(IZMD+NDOF),                 FORM1000
     &           Z(IZKE),MD,IELDOF,MD(IELDOF+1),NZ(IZLM),1.)            FORM1010
C                                                                       FORM1020
C---------- SUBTRACT GEOMETRIC STIFFNESS                                FORM1030
      NZ(IZKGDT+3)=NZ(IZKGDT+3)                                         FORM1040
C                                                                       FORM1050
      IF (NZ(IZKGDT+3).EQ.1) THEN                                       FORM1060
        FALSE=.FALSE.                                                   FORM1070
        CALL ELELIB                                                     FORM1080
     &   (2,LSTTYP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DAMAGE,DUCFAG,      FORM1090
     &   IELNO ,IELDOF,KGDOF,PGEOM,Z   ,AXIAL,IZDISP,IZLOAD,MSTOR)     FORM1100
C                                                                       FORM1110
        KGDOF=KGDOF                                                     FORM1120
C                                                                       FORM1130
        IF (BUG57) THEN                                                 FORM1140
          WRITE (ELNO,15)  IELNO,(NZ(I+IZLMKG-1),I=1,MIN(KGDOF,18))    FORM1150
          CALL LMATRX(Z(IZKEKG),MD,KGDOF,MD(KGDOF+1),ELNO)              FORM1160
        ENDIF                                                           FORM1170
C                                                                       FORM1180
        IF (.NOT. AXIAL) GO TO 120                                      FORM1190
C           DETERMINE THE FACTOR FOR KG                                 FORM1200
              IF (NZ(IZKGDT+1).EQ.1) THEN                               FORM1210
                    ACCELZ= Z(IZKGDT)                                   FORM1220
                    PGEOM=PGEOM*(1+ACCELZ)                              FORM1230
              ENDIF                                                     FORM1240
C     SWAP THE SIGN ON PGEOM FOR SUBTRACTION .  .                       FORM1250
      PGEOM=-PGEOM                                                      FORM1260
C                                                                       FORM1270
      CALL FORML(Z(IZSTIF),NZ(IZMD),NDOF,NZ(IZMD+NDOF),                 FORM1280
     &     Z(IZKEKG),MD,KGDOF,MD(KGDOF+1),NZ(IZLMKG),PGEOM)             FORM1290
      ENDIF                                                             FORM1300
C                                                                       FORM1310
  120   CONTINUE                                                        FORM1320
C                                                                       FORM1330
        RETURN                                                          FORM1340
C                                                                       FORM1350
C---------- ASSEMBLE GEOMETRIC STIFFNESS                                FORM1360
  200 CONTINUE                                                          FORM1370
      IF (NZ(IZKGDT+3) NE.2 ) RETURN                                    FORM1380
          IST =IZKG                                                     FORM1390
          IEND=IZKG + NZ(IZMD+KG+NDOF) - 1                              FORM1400
C                                                                       FORM1410
C--- ------- ZERO STIFFNESS                                            FORM1420
        DO 210 I=IST,IEND                                               FORM1430
  210     Z(I)=0                                                        FORM1440
C                                                                       FORM1450
C-------- --- LOOP FOR EACH ELEMENT                                    FORM1460
      DO 220 IELNO=1,NELMT                                              FORM1470
C                                                                       FORM1480
C---------- GET STIFFNESS OF EACH ELEMENT                               FORM1490
        FALSE=.FALSE.                                                   FORM1500
        CALL ELELIB                                                     FORM1510
     &   (2,LSTTYP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DAMAGE,DUCFAG,      FORM1520
     &   IELNO ,IELDOF,KGDOF,PGEOM,Z   ,AXIAL,IZDISP,IZLOAD,MSTOR)     FORM1530
C                                                                       FORM1540
        IF (.NOT. AXIAL) GO TO 220                                      FORM1550
C                                                                       FORM1560
        IF (BUG57) THEN                                                 FORM1570
          WRITE (6,*)  'PGEOM=',PGEOM                                   FORM1580
          WRITE (ELNO,15)  IELNO,(NZ(I-IZLMKG-1),I=1,MIN(KGDOF,18))    FORM1590
          CALL LMATRX(Z(IZKEKG),MD,KGDOF,MD(KGDOF+1),ELNO)              FORM1600
        ENDIF                                                           FORM1610
C                                                                       FORM1620
      CALL FORML(Z(IZKG),NZ(IZMDKG),NDOF,NZ(IZMDKG+NDOF),              FORM1630
     &     Z(IZKEKG),MD,KGDOF,MD(KGDOF+1),NZ(IZLMKG),PGEOM)             FORM1640
  220   CONTINUE                                                        FORM1650
C                                                                       FORM1660
        RETURN                                                          FORM1670
        END                                                             FORM1680
C-----------------------------------------------------------------------ELE00010
      SUBROUTINE ELE01                                                  ELE00020
     &  (IOPT       ,FIRST      ,PRINT      ,ID         ,IDOF       ,  ELE00030
     &   COORD      ,COSINE     ,JTCOS      ,CONST      ,DISPL      ,  ELE00040
     &   FORCE      ,IREL       ,XGDATA     ,PGEOM      ,IELD       ,  ELE00050
     &   ELD        ,NAME       ,AXIAL      ,IELNO      ,RINPUT     ,  ELE00060
     &   FUB        ,EESE       , EPSE      ,DAMAGE     ,IELTYP     ,  ELE00070
     &   IELDOF     ,TM         , R         ,RT         ,P          ,  ELE00080
     &   FT         ,LM         ,CTS        ,CTE        ,BS         ,  ELE00090
     &   BE         ,D          ,NODE1      ,NODEJ      ,JOINTI     ,  ELE00100
     &   JOINTJ     ,REL        ,LENGTH     ,KGDOF      ,LKG        ,  ELE00110
     &   LMKG       ,ISE        ,ENGDAT     ,FMAX       ,MAT        ,  ELE00120
     &   SE         ,STIFF      ,FEM        )                          ELE00130
C                                                                       ELE00140
      IMPLICIT REAL(A-H,O-Z)                                            ELE00150
C                                                                       ELE00160
      COMMON /RSA/ICOMN,RSAF(12)                                        ELE00170
      REAL KIIZ,KJJZ,KIJZ,KIIY,KJJY,KIJY,KTI,LENGTH,ICOMN              ELE00180
      INTEGER REL,RELT                                                  ELE00190
      LOGICAL FIRST,PRINT,FEMPLG,FALSE,AXIAL,BTEST                      ELE00200
      CHARACTER NAME*80,IRELC*6                                         ELE00210
$INCLUDE:'SCOMMN'                                                       ELE00220
      DIMENSION IDOF(MAXNOD,6),COORD(MAXNOD,6),COSINE(3,3,NCOS)        ELE00230
      DIMENSION CONST(MAXNOD,6),ID(MAXNOD),JTCOS(MAXNOD)                ELE00240
      DIMENSION ASAT(12,12),SAT(12,12),STIFF(156),XGDATA(3)             ELE00250
      DIMENSION A(12,12),S(12,12),LM(12),LMT(12),LMKG(12),FMAX(144)     ELE00260
      DIMENSION CTS(3,3),CTE(3,3),VX(3),VY(3),FUB(NDOF)                 ELE00270
      DIMENSION BS(3,3),BE(3,3),SE(156),ENGDAT(2),TM(14)                ELE00280
      DIMENSION R(12),DISPL(NDOF,NLOAD),F(12),RT(12),FT(12)             ELE00290
      DIMENSION FORCE(NDOF,NLOAD),IELD(6,MAXELD),ELD(12,MAXELD)         ELE00300
      DIMENSION RINPUT(100),FEM(12,NLOAD),GFEM(12),DUCT(3),EXCR(6)      ELE00310
C                                                                       ELE00320
C-----------------------------------------------------------------------ELE00330
C VARIABLES.                                                            ELE00340
C-------- ------------ ---  ------------------------------------------ ELE00350
C------ --- GLOBAL VARIABLES                                           ELE00360
C IOPT   - 1, INITIALIZE BEAM COLUMN ELEMENT                            ELE00370
C        - 2, CALCULATE GEOMETRIC STIFFNESS                             ELE00380
C        - 3, FORM STIFFNESS                                            ELE00390
C        - 4, CALCULATE INCREMENTAL FORCES                              ELE00400
C        - 5, CALCULATE TOTAL FORCES AND ENERGIES                       ELE00410
C IUNIT  - OUTPUT FILE UNIT # FOR PRINTING OUTPUT                       ELE00420
C FIRST  - FLAG, FIRST=.TRUE., PRINT HEADERS                            ELE00430
C PRINT  - FLAG, PRINT=.TRUE , PRINT DATA                               ELE00440
C NDOF   - # OF GLOBAL DOF                                              ELE00450
C NLOAD  - # OF LOAD COMBINATIONS                                       ELE00460
C MAXNOD - # ROW DIMENSION OF NODE ARRAY                                ELE00470
C NNODE  - # OF NODES                                                   ELE00480
C ID     - ARRAY OF EXTERNAL NODE NUMBERS                               ELE00490
C IDOF   - ARRAY OF DEGREES OF FREEDOM                                  ELE00500
C COORD  - ARRAY OF COORDINATES                                         ELE00510
C COSINE - ARRAY OF DIRECTION COSINES OF NODES                          ELE00520
C CONST  - ARRAY OF CONSTRAINT TRANSFORMATIONS                          ELE00530
C DISPL  - GLOBAL DISPLACEMENT MATRIX                                   ELE00540
C IZMAT  - LOCATION OF MATERIAL DATA IN GLOBAL STORAGE VECTOR           ELE00550
C NAME   - ELEMENT NAME                                                 ELE00560
C IELNO  - ELEMENT NUMBER                                               ELE00570
C RINPUT - INPUT DATA                                                   ELE00580
C STIFF  - OUTPUT ELEMENT STIFFNESS (UPPER TRIANGULAR FORM)             ELE00590
C LM     - OUTPUT LOCAL TO GLOBAL STIFFNESS MAPPING MATRIX              ELE00600
C---------- ELEMENT VARIABLES                                           ELE00610
C          - TM(2,14),D                                                 ELE00620
C          - INDIVIDUAL TERMS IN THE ELEMENT STIFFNESS (S) MATRIX       ELE00630
C PKG      - TM(1): INPUT LOAD FOR FORMING GEOMETRIC STIFFNESS MATRIX . ELE00640
C R        - INCRIMENTAL DISPLACEMENTS                                  ELE00650
C RT       - TOTAL DISPLACEMENTS                                        ELE00660
C F        - INCRIMENTAL FORCES                                         ELE00670
C FT       - TOTAL FORCES                                               ELE00680
C LM       - LOCAL TO GLOBAL STIFFNESS MAPPING MATRIX                   ELE00690
C CTS      - LOCAL TO GLOBAL ROTATION MATRIX START                      ELE00700
C CTE      - LOCAL TO GLOBAL ROTATION MATRIX END                        ELE00710
C BS       - LOCAL TO GLOBAL ROTATED ECCENTRICITY/CONSTRAINT  MATRIX STARTELE00720
C BE       - LOCAL TO GLOBAL ROTATED ECCENTRICITY/CONSTRAINT  MATRIX END ELE00730
C STIFF  - ELEMENT STIFFNESS                                            ELE00740
C IELTYP - ELEMENT TYPE - 1C                                            ELE00750
C NODEI  - EXTERNAL JOINT NUMBER, END I                                 ELE00760
C NODEJ  - EXTERNAL JOINT NUMBER, END J                                 ELE00770
C JOINTI - INTERNAL JOINT NUMBER, END I                                 ELE00780
C JOINTJ - INTERNAL JOINT NUMBER, END J                                 ELE00790
C---------- TEMPROARY VARIABLES                                         ELE00800
C VX     - VECTOR DEFINING THE ELEMENT X AXIS                           ELE00810
C VY     - VECTOR DEFINING THE ELEMENT Y AXIS                           ELE00820
C COSLOC - LOCAL COSINE MATRIX                                          ELE00830
C A      - GLOBAL TO LOCAL FORCE TRANSFORMATION MATRIX                  ELE00840
C S      - ELEMENT STIFFNESS AT LOCAL COORD, AT ENDS OF FLEXABLE PART   ELE00850
C SAT    - PRODUCT OF S*A'                                              ELE00860
C ASAT   - PRODUCT OF A*SAT                                             ELE00870
C-------------------------------------------------------------- ---------ELE00880
C                                                                       ELE00890
C------- CHOOSE OPTION                                                  ELE00900
        IF (IOPT.LT.1 .OR.IOPT.GT.14) THEN                              ELE00910
          WRITE(108,*) 'INVALID OPTION IN ELE-01, IOPT=',IOPT           ELE00920
          STOP                                                          ELE00930
         ENDIF                                                          ELE00940
C                                                                       ELE00950
        GO TO (100,200,300,400,500,600,700,800,900,1000,1100,           ELE00960
     &                                        1200,1300,1400) IOPT      ELE00970
C                                                                       ELE00980
  100 CONTINUE                                                          ELE00990
C                                                                       ELE01000
C------ ZERO STRAIN ENERGY                                             ELE01010
        ENGDAT(1) = 0                                                   ELE01020
        ENGDAT(2) = 0                                                   ELE01030
C                                                                       ELE01040
C------ INTERPERTATE INPUT DATA                                        ELE01050
        IELTYP= 1                                                       ELE01060
        MAT    = RINPUT( 1)                                             ELE01070
        NODEI  = RINPUT( 2)                                             ELE01080
        NODEJ  = RINPUT( 3)                                             ELE01090
        VY(1)  = RINPUT( 4)                                             ELE01100
        VY(2)  = RINPUT( 5)                                             ELE01110
        VY(3)  = RINPUT( 6)                                             ELE01120
        XS     = RINPUT( 7)                                             ELE01130
        XE     = RINPUT( 8)                                             ELE01140
        TM(1)  = RINPUT( 9)                                             ELE01150
        RELT   = RINPUT(10)                                             ELE01160
C                                                                       ELE01170
C------ GET INTERNAL NODE # ASSOCIATED WITH EXTERNAL NODE #'S          ELE01180
        JOINTI=IQUICK(NODEI,ID,NNODE)                                   ELE01190
        JOINTJ=IQUICK(NODEJ,ID,NNODE)                                   ELE01200
C                                                                       ELE01210
C------ PRINT DATA FOR PLOTTING                                        ELE01220
        IF (BTEST(IBUG,2)) WRITE(7,10) NODEJ,NODEI,(COORD(JOINTJ,I)     ELE01230
     &                                            ,I=1,3),(COORD(JOINTI,J),J=1,3)  ELE01240
   10   FORMAT (I5,1X,I5,1P,6G12.4)                                     ELE01250
C                                                                       ELE01260
C-----------------------------------------------------------------------ELE01270
C  GET GLOBAL TO LOCAL TRANSFORMATION MATRIX                            ELE01280
C-------------------------------------------------------------------- --ELE01290
C------ DEFINE VECTOR X, LENGTH                                        ELE01300
        VX(1)=COORD(JOINTJ,1)-COORD(JOINTI,1)                           ELE01310
        VX(2)=COORD(JOINTJ,2)-COORD(JOINTI,2)                           ELE01320
        VX(3)=COORD(JOINTJ,3)-COORD(JOINTI,3)                           ELE01330
        LENGTH=SQRT(VX(1)**2+VX(2)**2+VX(3)**2)-XS-XE                   ELE01340
        IF (LENGTH.LE.0) THEN                                           ELE01350
          WRITE(108,101) IELNO,JOINTI,JOINTJ,LENGTH                     ELE01360
  101     FORMAT(1X,20('*'),'ER','ROR - LENGTH IS LE 0',/               ELE01370
     &    5X,'IELNO=',I5,5X,'JOINTI=',I5,5X,'JOINTJ=',I5,              ELE01380
     &    5X,'LENGTH',1P,G15.6/                                         ELE01390
     &    5X,'REVISE INPUT.....................'/)                      ELE01400
          LENGTH=1.0                                                    ELE01410
        ENDIF                                                           ELE01420
C------ GET LOCAL TO GLOBAL TRANSFORMATION MATRICIES CT AND B           ELE01430
        ICS=JTCOS(JOINTI)                                               ELE01440
        ICE=JTCOS(JOINTJ)                                               ELE01450
        CALL ROTEYE(VX,VY,COSINE(1,1,ICS),CTS, XS ,0.,0.,BS,            ELE01460
     &             CONST(JOINTI,1),MAXNOD)                              ELE01470
        CALL ROTEYE(VX,VY,COSINE(1,1,ICE),CTE,-XE ,0.,0.,BE,            ELE01480
     &             CONST(JOINTJ,1),MAXNOD)                              ELE01490
C                                                                       ELE01500
C------ GET GLOBAL CONSTRAINT MATRIX BS, BE                            ELE01510
C     BS(1,1)=CONST(JOINTI,1)*CTS(2,1)+CONST(JOINTI,2)*CTS(3,1)+BS(1,1) ELE01520
C     BS(1,2)=CONST(JOINTI,1)*CTS(2,2)+CONST(JOINTI,2)*CTS(3,2)+BS(1,2) ELE01530
C     BS(1,3)=CONST(JOINTI,1)*CTS(2,3)+CONST(JOINTI,2)*CTS(3,3)+BS(1,3) ELE01540
C     BS(2,1)=CONST(JOINTI,3)*CTS(1,1)+CONST(JOINTI,4)*CTS(3,1)+BS(2,1) ELE01550
C     BS(2,2)=CONST(JOINTI,3)*CTS(1,2)+CONST(JOINTI,4)*CTS(3,2)+BS(2,2) ELE01560
C     BS(2,3)=CONST(JOINTI,3)*CTS(1,3)+CONST(JOINTI,4)*CTS(3,3)+BS(2,3) ELE01570
C     BS(3,1)=CONST(JOINTI,5)*CTS(1,1)+CONST(JOINTI,6)*CTS(2,1)+BS(3,1) ELE01580
C     BS(3,2)=CONST(JOINTI,5)*CTS(1,2)+CONST(JOINTI,6)*CTS(2,2)+BS(3,2) ELE01590
C     BS(3,3)=CONST(JOINTI,5)*CTS(1,3)+CONST(JOINTI,6)*CTS(2,3)+BS(3,3) ELE01600
C                                                                       ELE01610
C     BE(1,1)=CONST(JOINTJ,1)*CTE(2,1)+CONST(JOINTJ,2)*CTE(3,1)+BE(1,1) ELE01620
C     BE(1,2)=CONST(JOINTJ,1)*CTE(2,2)+CONST(JOINTJ,2)*CTE(3,2)+BE(1,2) ELE01630
C     BE(1,3)=CONST(JOINTJ,1)*CTE(2,3)+CONST(JOINTJ,2)*CTE(3,3)+BE(1,3) ELE01640
C     BE(2,1)=CONST(JOINTJ,3)*CTE(1,1)+CONST(JOINTJ,4)*CTE(3,1)+BE(2,1) ELE01650
C     BE(2,2)=CONST(JOINTJ,3)*CTE(1,2)+CONST(JOINTJ,4)*CTE(3,2)+BE(2,2) ELE01660
C     BE(2,3)=CONST(JOINTJ,3)*CTE(1,3)+CONST(JOINTJ,4)*CTE(3,3)+BE(2,3) ELE01670
C     BE(3,1)=CONST(JOINTJ,5)*CTE(1,1)+CONST(JOINTJ,6)*CTE(2,1)+BE(3,1) ELE01680
C     BE(3,2)=CONST(JOINTJ,5)*CTE(1,2)+CONST(JOINTJ,6)*CTE(2,2)+BE(3,2) ELE01690
C     BE(3,3)=CONST(JOINTJ,5)*CTE(1,3)+CONST(JOINTJ,6)*CTE(2,3)+BE(3,3) ELE01700
C                                                                       ELE01710
C                                                                       ELE01720
C------ ZERO MATRICIES                                                 ELE01730
        DO 40 I=1,12                                                    ELE01740
          F (I)=0                                                       ELE01750
          FT(I)=0                                                       ELE01760
          R (I)=0                                                       ELE01770
          RT(I)=0                                                       ELE01780
          DO 40 J=1,12                                                  ELE01790
            FMAX((I-1)*12+J)=0.                                         ELE01800
            S(I,J)=0.                                                   ELE01810
   40     A(I,J)=0                                                      ELE01820
C                                                                       ELE01830
C------ ASSEMBLE TRANSFORMATION MATRIX A                               ELE01840
        DO 50 I=1,3                                                     ELE01850
        DO 50 J=1,3                                                     ELE01860
          A(I  ,J  )=CTS(I,J)                                           ELE01870
          A(I+3,J+3)=CTS(I,J)                                           ELE01880
          A(I+6,J+6)=CTE(I,J)                                           ELE01890
          A(I+9,J+9)=CTE(I,J)                                           ELE01900
          A(I+3,J  )=BS(I,J)                                            ELE01910
   50     A(I+9,J+6)=BE(I,J)                                            ELE01920
C                                                                       ELE01930
C------ GET MATERIAL PROPERTIES                                        ELE01940
        FALSE=.FALSE                                                    ELE01950
        LSTTYP=0                                                        ELE01960
        CALL MATLIB(2,LSTTYP,FALSE,EESE,EPSE,DUCT,EXCR,                 ELE01970
     &    FT,F,RT,R,0 ,0.,LELEM,LMAT,MAT,MATTYP,SE,IELNO,DAMAGE)        ELE01980
C                                                                       ELE01990
        IF (MATTYP.NE.1) THEN                                           ELE02000
          WRITE(108,190) IELNO,MATTYP,NODEI,NODEJ                       ELE02010
  190     FORMAT('0ELEMENT ',I6,'  INCOMPATIBLE MATERIAL PROP.#',I5,    ELE02020
     &    '  START JOINT=',I5,'  END JOINT ',I5,/                       ELE02030
     &    ' THE ELEMENT IS REJECTED AND ER','ROR CHECKING CONTINUES']   ELE02040
          GO TO 191                                                     ELE02050
        ENDIF                                                           ELE02060
C                                                                       ELE02070
        EAX   =SE( 1)                                                   ELE02080
        GAY   =SE( 2)                                                   ELE02090
        GAZ   =SE( 3)                                                   ELE02100
        GIX   =SE( 4)                                                   ELE02110
        EIY   =SE( 5)                                                   ELE02120
        EIZ   =SE( 6)                                                   ELE02130
C                                                                       ELE02140
C------ -- ------------------------------  -- ---  --- ---------------ELE02150
C  DETERMINE THE MEMBER RELEASE CODES                                   ELE02160
C-------------------------------------------------------- --------- ---ELE02170
C     RELT  IS AN INPUT MEMBER RELEASE CODE                             ELE02180
```

```
C     REL   IS THE INTERNAL MEMBER RELEASE CODE                          ELE02190
C           BIT 6 AXIAL -          TRUE = AXIAL DEFORMATION CONSIDERED   ELE02200
C           BIT 5 MX AT END I   :                                        ELE02210
C           BIT 4 MY AT END I   :-- ) TRUE-RELEASE                       ELE02220
C           BIT 3 MZ AT END I   :                                        ELE02230
C           BIT 2 MX AT END J   :                                        ELE02240
C           BIT 1 MY AT END J   :                                        ELE02250
C           BIT 0 MZ AT END J   :                                        ELE02260
C                                                                        ELE02270
      REL=0                                                              ELE02280
      IRELC=' '                                                          ELE02290
      IF (MOD(RELT,10).NE.0) THEN                                        ELE02300
           REL=IBSET(REL,0)                                              ELE02310
           IRELC(6:6)='Z'                                                ELE02320
      ENDIF                                                              ELE02330
      RELT=RELT/10                                                       ELE02340
      IF (MOD(RELT,10).NE.0) THEN                                        ELE02350
           REL=IBSET(REL,1)                                              ELE02360
           IRELC(5:5)='Y'                                                ELE02370
      ENDIF                                                              ELE02380
      RELT=RELT/10                                                       ELE02390
      IF (MOD(RELT,10).NE.0) THEN                                        ELE02400
           REL=IBSET(REL,2)                                              ELE02410
           IRELC(4:4)='X'                                                ELE02420
      ENDIF                                                              ELE02430
      RELT=RELT/10                                                       ELE02440
      IF (MOD(RELT,10).NE.0) THEN                                        ELE02450
           REL=IBSET(REL,3)                                              ELE02460
           IRELC(3:3)='Z'                                                ELE02470
      ENDIF                                                              ELE02480
      RELT=RELT/10                                                       ELE02490
      IF (MOD(RELT,10).NE.0) THEN                                        ELE02500
           REL=IBSET(REL,4)                                              ELE02510
           IRELC(2:2)='Y'                                                ELE02520
      ENDIF                                                              ELE02530
      RELT=RELT/10                                                       ELE02540
      IF (MOD(RELT,10).NE.0) THEN                                        ELE02550
           REL=IBSET(REL,5)                                              ELE02560
           IRELC(1:1)='X'                                                ELE02570
      ENDIF                                                              ELE02580
C--------- SET STIFFNESS COIEFFICIENTS...                                ELE02590
C--------- SET CONSISTENT MASS GEOMETRIC STIFFNESS COIEFFICIENTS...      ELE02600
      AZ=0                                                               ELE02610
      BZ=0                                                               ELE02620
      CZ=0                                                               ELE02630
      DZ=0                                                               ELE02640
      EZ=0                                                               ELE02650
      FZ=1.0 / LENGTH                                                    ELE02660
C .. ..... BOTH ENDS RELEASED, Z-AXIS ... ..... .....                   ELE02670
      IF (BTEST(REL,0) .AND. BTEST(REL,3)) THEN                          ELE02680
        KIIZ=0                                                           ELE02690
        KJJZ=0                                                           ELE02700
        KIJZ=0                                                           ELE02710
C... . . ... START END RELEASED, Z-AXIS . .....                         ELE02720
      ELSE IF (BTEST(REL,3)) THEN                                        ELE02730
        KIIZ=0                                                           ELE02740
        KJJZ=3                                                           ELE02750
        KIJZ=0                                                           ELE02760
        BZ=  0.20 * LENGTH                                               ELE02770
        EZ=  0.20                                                        ELE02780
        FZ=  1.20 / LENGTH                                               ELE02790
C ..... ..... END END RELEASED, Z-AXIS . ..... .. .. . . ...            ELE02800
      ELSE IF (BTEST(REL,0)) THEN                                        ELE02810
        KIIZ=3                                                           ELE02820
        KJJZ=0                                                           ELE02830
        KIJZ=0                                                           ELE02840
        AZ=  0.20 * LENGTH                                               ELE02850
        DZ=  0.20                                                        ELE02860
        FZ=  1.20 / LENGTH                                               ELE02870
C ..... .. BOTH ENDS RELEASED, Z-AXIS .... .....                         ELE02880
      ELSE                                                               ELE02890
        KIIZ=4                                                           ELE02900
        KJJZ=4                                                           ELE02910
        KIJZ=2                                                           ELE02920
        AZ=  2 * LENGTH / 15.                                            ELE02930
        BZ=  AZ                                                          ELE02940
        CZ=  LENGTH / 30.                                                ELE02950
        DZ=  0.10                                                        ELE02960
        EZ=  0.10                                                        ELE02970
        FZ=  1.2 / LENGTH                                                ELE02980
      ENDIF                                                              ELE02990
C                                                                        ELE03000
      AY=0                                                               ELE03010
      BY=0                                                               ELE03020
      CY=0                                                               ELE03030
      DY=0                                                               ELE03040
      EY=0                                                               ELE03050
      FY=1.0 / LENGTH                                                    ELE03060
C. . ..... BOTH ENDS RELEASED, Y-AXIS .. ..... .....                    ELE03070
      IF (BTEST(REL,1) .AND. BTEST(REL,4)) THEN                          ELE03080
        KIIY=0                                                           ELE03090
        KJJY=0                                                           ELE03100
        KIJY=0                                                           ELE03110
C . . ... START END RELEASED, Y-AXIS .. ..... .....                     ELE03120
      ELSE IF (BTEST(REL,4)) THEN                                        ELE03130
        KIIY=0                                                           ELE03140
        KJJY=3                                                           ELE03150
        KIJY=0                                                           ELE03160
        BY=  0.20 * LENGTH                                               ELE03170
        EY=  0.20                                                        ELE03180
        FY=  1.20 / LENGTH                                               ELE03190
C. ..... END END RELEASED, Y-AXIS ..... ..... .                         ELE03200
      ELSE IF (BTEST(REL,1)) THEN                                        ELE03210
        KIIY=3                                                           ELE03220
        KJJY=0                                                           ELE03230
        KIJY=0                                                           ELE03240
        AY=  0.20 * LENGTH                                               ELE03250
        DY=  0.20                                                        ELE03260
        FY=  1.20 / LENGTH                                               ELE03270
C ..... BOTH ENDS RELEASED, Z-AXIS .. .. ..... .... ...                 ELE03280
      ELSE                                                               ELE03290
        KIIY=4                                                           ELE03300
        KJJY=4                                                           ELE03310
        KIJY=2                                                           ELE03320
        AY=  2 * LENGTH / 15.                                            ELE03330
        BY=  AY                                                          ELE03340
        CY=  LENGTH / 30.                                                ELE03350
        DY=  0.10                                                        ELE03360
        EY=  0.10                                                        ELE03370
        FY= 1.2 / LENGTH                                                 ELE03380
      ENDIF                                                              ELE03390
C                                                                        ELE03400
      IF (BTEST(REL,2)  AND  BTEST(REL,5)) THEN                          ELE03410
        KTI  =0                                                          ELE03420
        KTJ  =0                                                          ELE03430
      ELSE IF (BTEST(REL,5)) THEN                                        ELE03440
        KTI  =0                                                          ELE03450
        KTJ  =1                                                          ELE03460
      ELSE IF (BTEST(REL,2)) THEN                                        ELE03470
        KTI  =1                                                          ELE03480
        KTJ  =0                                                          ELE03490
      ELSE                                                               ELE03500
        KTI  =1                                                          ELE03510
        KTJ  =1                                                          ELE03520
      ENDIF                                                              ELE03530
C                                                                        ELE03540
C-------------------------------------- ---------- ---  ---------       ELE03550
C  ASSEMBLE LOCAL STIFFNESS MATRIX                                       ELE03560
C------------------------------------------------------------ ------     ELE03570
      TM( 2)=EAX/LENGTH                                                  ELE03580
      TM( 3)=GIX/LENGTH                                                  ELE03590
      TM( 4)=KIIZ*(EIZ/LENGTH)                                           ELE03600
      TM( 5)=KJJZ*(EIZ/LENGTH)                                           ELE03610
      TM( 6)=KIJZ*(EIZ/LENGTH)                                           ELE03620
      TM( 7)=KIIY*(EIY/LENGTH)                                           ELE03630
      TM( 8)=KJJY*(EIY/LENGTH)                                           ELE03640
      D =KIJY*(EIY/LENGTH)                                               ELE03650
      TM( 9) = (TM( 4)-2*TM( 6)+TM( 5))/(LENGTH**2)                     ELE03660
      TM(10) = (TM( 4)-  TM( 6))/(LENGTH    )                           ELE03670
      TM(11)=  (TM( 6)-  TM( 5))/(LENGTH    )                           ELE03680
      TM(12) = (TM(7)-2*D+TM( 8))/(LENGTH**2)                            ELE03690
      TM(13) = (TM(7)-  D    )/(LENGTH    )                              ELE03700
      TM(14)=  (    D-TM( 8))/(LENGTH    )                              ELE03710
      S( 1, 1) = TM( 2)                                                  ELE03720
      S( 1, 7) =-TM( 2)                                                  ELE03730
      S( 2, 2) = TM( 9)                                                  ELE03740
      S( 2, 6) = TM(10)                                                  ELE03750
      S( 2, 8) =-TM( 9)                                                  ELE03760
      S( 2,12) = TM(11)                                                  ELE03770
      S( 3, 3) = TM(12)                                                  ELE03780
      S( 3, 5) =-TM(13)                                                  ELE03790
      S( 3, 9) =-TM(12)                                                  ELE03800
      S( 3,11) =-TM(14)                                                  ELE03810
      S( 4, 4) = TM( 3) * KTI                                            ELE03820
      S( 4,10) =-TM( 3)                                                  ELE03830
      S( 5, 5) = TM(7)                                                   ELE03840
      S( 5, 9) = TM(13)                                                  ELE03850
      S( 5,11) = D                                                       ELE03860
      S( 6, 6) = TM( 4)                                                  ELE03870
      S( 6, 8) =-TM(10)                                                  ELE03880
      S( 6,12) = TM( 6)                                                  ELE03890
      S( 7, 7) = TM( 2)                                                  ELE03900
      S( 8, 8) = TM( 9)                                                  ELE03910
      S( 8,12) =-TM(11)                                                  ELE03920
      S( 9, 9) = TM(12)                                                  ELE03930
      S( 9,11) = TM(14)                                                  ELE03940
      S(10,10) = TM( 3) * KTJ                                            ELE03950
      S(11,11) = TM( 8)                                                  ELE03960
      S(12,12) = TM( 5)                                                  ELE03970
      DO 55 I=1,12                                                       ELE03980
      DO 55 J=I+1,12                                                     ELE03990
55    S( J, I) = S( I, J)                                                ELE04000
C                                                                        ELE04010
C----  CALCULATE SAT                                                     ELE04020
      CALL MULTM(3,12,SAT,S,A)                                           ELE04030
      CALL MULTM(12,12,SAT,S,SAT)                                        ELE04040
C                                                                        ELE04050
C----  CALCULATE ASAT                                                    ELE04060
      CALL MULTM(6,12,ASAT,A,SAT)                                        ELE04070
C                                                                        ELE04080
C----- DEGREES OF FREEDOM                                                ELE04090
C      REMOVE DOF'S THAT ARE CONSTRAINED TO THE SAME DOF ..              ELE04100
      DO 75 I=1,6                                                        ELE04110
        IF ( IDOF(JOINTI,I).NE.IDOF(JOINTJ,I) ) THEN                     ELE04120
           LMT(I  )=IDOF(JOINTI,I)                                       ELE04130
           LMT(I+6)=IDOF(JOINTJ,I)                                       ELE04140
        ELSE                                                             ELE04150
           LMT(I  )= 0                                                   ELE04160
           LMT(I+6)= 0                                                   ELE04170
        ENDIF                                                            ELE04180
75    CONTINUE                                                           ELE04190
C                                                                        ELE04200
C----- CONVERT TO HALF STORAGE MODE  BY COLUMNS                          ELE04210
C                                                                        ELE04220
C          1  3  6 10 15 21 28 36 45 55 66 78  NUMBER INDICATES         ELE04230
C             2  5  9 14 20 27 35 44 54 65 77  LOCATION OF DATA         ELE04240
C                4  8 13 19 26 34 43 53 64 76  IN ARRAY STIFF           ELE04250
C                   7 12 18 25 33 42 52 63 75                           ELE04260
C                     11 17 24 32 41 51 62 74                           ELE04270
C                        16 23 31 40 50 61 73                           ELE04280
C                           22 30 39 49 60 72                           ELE04290
C                              29 38 48 59 71                           ELE04300
C                                 37 47 58 70                           ELE04310
C                                    46 57 69                           ELE04320
C                                       56 68                           ELE04330
C                                          67                            ELE04340
C                                                                        ELE04350
      L=0                                                                ELE04360
      IELDOF=0                                                           ELE04370
      DO 90 I=1,12                                                       ELE04380
        IF (LMT(I).EQ.0 .OR. ASAT(I,I).LE.0) GO TO 90                    ELE04390
        IELDOF=IELDOF+1                                                  ELE04400
        LM(IELDOF)=LMT(I)                                                ELE04410
        DO 80 J=I,1,-1                                                   ELE04420
          IF (LMT(J).EQ.0 .OR. ASAT(J,J).LE.0 ) GO TO 80                 ELE04430
          L=L+1                                                          ELE04440
          STIFF(L)=ASAT(I,J)                                             ELE04450
80      CONTINUE                                                         ELE04460
90    CONTINUE                                                           ELE04470
C------------------------------------------------------------------      ELE04480
C  ASSEMBLE GEOMETRIC STIFFNESS MATRIX FOR A UNIT LOAD..                 ELE04490
C------------------------------ ---------------------------------       ELE04500
      LXG=L                                                              ELE04510
      IF ( TM(1).EQ.0 .AND. KGDATA(1).NE.2 ) THEN                        ELE04520
         AXIAL=.FALSE.                                                   ELE04530
         REL=IBCLR(REL,6)                                                ELE04540
         GO TO 180                                                       ELE04550
      ENDIF                                                              ELE04560
C                                                                        ELE04570
C---------- DETERMINE IF AXIAL DEFORMATION IS CONSIDERED                 ELE04580
C             BIT #6 OF REL IS TRUE IF AXIAL DEFORMATION IS CONSIDERED   ELE04590
      DO 999 I=1,3                                                       ELE04600
         CS=CTS(I,1)                                                     ELE04610
         CE=CTE(I,1)                                                     ELE04620
         IF (ABS(CS).LE. .0001 .AND  ABS(CE).LE. .0001 ) GO TO 999       ELE04630
         IS=IDOF(JOINTI,I)                                               ELE04640
         IE=IDOF(JOINTJ,I)                                               ELE04650
         IF (IS.EQ.IE) GO TO 999                                         ELE04660
           REL=IBSET(REL,6)                                              ELE04670
999   CONTINUE                                                           ELE04680
C                                                                        ELE04690
      AXIAL=BTEST(REL,6)                                                 ELE04700
      IF (KGDATA(1).NE.0 .AND  KGDATA(2).NE.0 .AND. BTEST(REL,6)) THEN   ELE04710
C                                                                        ELE04720
C---------- DETERMINE THE AXIAL LOAD TO BE USED FOR XG..                 ELE04730
         IF (KGDATA(1) EQ.1) THEN                                        ELE04740
              PGBOM =  TM(1)                                             ELE04750
         ELSE IF (KGDATA(1).EQ.2) THEN                                   ELE04760
              PGBOM =  0.50*(F(1)-F(7))                                  ELE04770
         ENDIF                                                           ELE04780
C                                                                        ELE04790
C--ASSEMBLE LOCAL GEOMETRIC STIFFNESS MATRIX FOR A UNIT LOAD ..          ELE04800
C           ZERO LOCAL KG MATRIX                                         ELE04810
            DO 998 I=1,12                                                ELE04820
            DO 998 J=1,12                                                ELE04830
998           S(I,J)=0                                                   ELE04840
C                                                                        ELE04850
C--- ----- LUMPED MASS FORM OF GEOMETRIC STIFFNESS                       ELE04860
         IF (KGDATA(2).EQ.1) THEN                                        ELE04870
              F1=1/LENGTH                                                ELE04880
              S( 2, 2)= F1                                               ELE04890
              S( 2, 8)= -F1                                              ELE04900
              S( 8, 2)= -F1                                              ELE04910
              S( 8, 8)=  F1                                              ELE04920
              S( 3, 3)=  F1                                              ELE04930
              S( 3, 9)= -F1                                              ELE04940
              S( 9, 3)= -F1                                              ELE04950
              S( 9, 9)=  F1                                              ELE04960
C                                                                        ELE04970
C-- ------ CONSISTENT MASS FORM OF GEOMETRIC STIFFNESS                   ELE04980
         ELSE IF (KGDATA(2).EQ.2) THEN                                   ELE04990
              S( 2, 2)=  FZ                                              ELE05000
              S( 2, 6)=  DZ                                              ELE05010
              S( 2, 8)= -FZ                                              ELE05020
              S( 2,12)=  EZ                                              ELE05030
              S( 6, 2)=  DZ                                              ELE05040
              S( 6, 6)=  AZ                                              ELE05050
              S( 6, 8)= -DZ                                              ELE05060
              S( 6,12)=  CZ                                              ELE05070
              S( 8, 2)= -FZ                                              ELE05080
              S( 8, 6)= -DZ                                              ELE05090
              S( 8, 8)=  FZ                                              ELE05100
              S( 8,12)= -EZ                                              ELE05110
```

```
            S(12, 2)= -EZ                                                       ELE05110
            S(12, 6)= -CZ                                                       ELE05120
            S(12, 8)= -EZ                                                       ELE05130
            S(12,12)= -BZ                                                       ELE05140
            S( 3, 3)= -FY                                                       ELE05150
            S( 3, 5)= -DY                                                       ELE05160
            S( 3, 9)= -FY                                                       ELE05170
            S( 3,11)= -EY                                                       ELE05180
            S( 5, 3)= -DY                                                       ELE05190
            S( 5, 5)= -AY                                                       ELE05200
            S( 5, 9)= -DY                                                       ELE05210
            S( 5,11)= -CY                                                       ELE05220
            S( 9, 3)= -FY                                                       ELE05230
            S( 9, 5)= -DY                                                       ELE05240
            S( 9, 9)= -FY                                                       ELE05250
            S( 9,11)= -EY                                                       ELE05260
            S(11, 3)= -EY                                                       ELE05270
            S(11, 5)= -CY                                                       ELE05280
            S(11, 9)= -EY                                                       ELE05290
            S(11,11)= -BY                                                       ELE05300
         ENDIF                                                                  ELE05310
C                                                                               ELE05320
C--- - CALCULATE SAT                                                           ELE05330
         CALL MULTM(3,12,SAT,S,A)                                               ELE05340
C                                                                               ELE05350
C----- CALCULATE ASAT                                                          ELE05360
         CALL MULTM(0,12,ASAT,A,SAT)                                            ELE05370
C                                                                               ELE05380
C----- CONVERT TO HALF STORAGE MODE  BY COLUMNS                                ELE05390
         KGDOF=0                                                                ELE05400
         DO 1090 I=1,12                                                         ELE05410
           IF (LMT(I).EQ.0 .OR. ASAT(I,I) LE.0) GO TO 1090                      ELE05420
           KGDOF=KGDOF+1                                                        ELE05430
           LMKG(KGDOF)=LMT(I)                                                   ELE05440
           DO 1080 J=I,1,-1                                                     ELE05450
             IF (LMT(J).EQ.0  OR. ASAT(J,J).LE.0 ) GO TO 1080                   ELE05460
             L=L+1                                                              ELE05470
             STIFF(L)=ASAT(I,J)                                                 ELE05480
1080         CONTINUE                                                           ELE05490
1090         CONTINUE                                                           ELE05500
      ENDIF                                                                     ELE05510
C                                                                               ELE05520
C----- RELEASE UNUSED STORAGE                                                  ELE05530
  180 IREL=156-L                                                                ELE05540
C                                                                               ELE05550
C     PRINT COLUMN DATA                                                         ELE05560
      IF (PFIRST) WRITE(108,192)                                                ELE05570
  191 WRITE(108,193) NAME,IELNO,MAT,NODEI,NODEJ,IRELC,LENGTH,                   ELE05580
     &                VY(1),VY(2),VY(3),XS,XE,TH(1)                            ELE05590
C                                                                               ELE05600
  192 FORMAT(//' ELEMENT NO.  3D BEAM ELEMENT'//                                ELE05610
     &  T22,'#  MATL START   END',/,'  REL CD',3X,'LENGTH',3X,11('-'),          ELE05620
     &  ' Y-AXIS  ',12('-'),' START DIST    END DIST      PKG')                ELE05630
  193 FORMAT(1X,A15,4I6,2X,A,1P,G12.4,0P,                                       ELE05640
     &       F9.5,' I',5P,F9.5,' J',F9.5,' K',1P,5E,3G12.4)                     ELE05650
      RETURN                                                                    ELE05660
C---------------------- DETERMINE GEOMETRIC STIFF ----------------------------ELE05670
  200 CONTINUE                                                                  ELE05680
C- KG IS DETERMINED FOR A UNIT LOAD WITH IOPT=1,                                ELE05690
C- THE ACTUAL LOAD (PGEOM) IS DETERMINED HERE                                   ELE05700
C- KG IS MULTIPLIED BY PGEOM WHEN THE TOTAL STIFFNESS IS ASSEMBLED              ELE05710
C- NEGATIVE PGEOM IS COMPRESSION ...                                            ELE05720
      AXIAL=BTEST(REL,6)                                                        ELE05730
C                                                                               ELE05740
C---------- DETERMINE THE AXIAL LOAD TO BE USED FOR KG..                        ELE05750
      IF (AXIAL) THEN                                                           ELE05760
         IF (KGDATA(1) EQ.1) THEN                                               ELE05770
              PGEOM =  TM(1)                                                    ELE05780
         ELSE IF (KGDATA(1).EQ.2) THEN                                          ELE05790
              PGEOM =  0.50*(F(1)-F(7))                                         ELE05800
         ENDIF                                                                  ELE05810
      ENDIF                                                                     ELE05820
C                                                                               ELE05830
      RETURN                                                                    ELE05840
C---------------------- DETERMINE STIFFNESS -----------------------------------ELE05850
  300 CONTINUE                                                                  ELE05860
C                                                                               ELE05870
C     FORCE-FRAME DISPLACEMENT MATRIX                                           ELE05880
C     THE STIFFNESS IS STORED IN STIFF, THE MAPING MATRIX IN LM                 ELE05890
C     THESE HAVE THE GLOBAL ADDRESSES I2KE AND I2LM...                          ELE05900
C                                                                               ELE05910
      RETURN                                                                    ELE05920
C---------------------- DETERMINE INCRIMENTAL FORCES --------------------------ELE05930
  400 CONTINUE                                                                  ELE05940
      IF (PRINT.AND.FIRST) WRITE(108,491) STEPID                                ELE05950
C                                                                               ELE05960
C------- CALL BMLOA TO DETERMINE THE FIXED END FORCES                           ELE05970
      IF (MAXELD.GT.0) THEN                                                     ELE05980
         CALL BMLOA(IELNO,FEMFLG,LENGTH,REL,MAXELD,NELD,NLOAD,                  ELE05990
     &              FEM,IELD,ELD)                                               ELE06000
      ELSE                                                                      ELE06010
         FEMFLG=.FALSE.                                                         ELE06020
      ENDIF                                                                     ELE06030
C                                                                               ELE06040
C- ---- LOOP FOR EACH LOAD                                                     ELE06050
      DO 630 LOAD=1,NLOAD                                                       ELE06060
C                                                                               ELE06070
C-----  TRANSFORM FRAME DISPL INTO LOCAL DISPL                                  ELE06080
       DO 610 I=1,3                                                             ELE06090
        R(I  )=0.                                                               ELE06100
        R(I+3)=0.                                                               ELE06110
        R(I+6)=0.                                                               ELE06120
        R(I+9)=0.                                                               ELE06130
        DO 610 J=1,3                                                            ELE06140
          D1=DISPL(IDOF(JOINTI,J  ),LOAD)                                       ELE06150
          D2=DISPL(IDOF(JOINTI,J+3),LOAD)                                       ELE06160
          D3=DISPL(IDOF(JOINTJ,J  ),LOAD)                                       ELE06170
          D4=DISPL(IDOF(JOINTJ,J+3),LOAD)                                       ELE06180
        R(I  )=R(I  )+CTS(J,I)*D1 + BS(J,I)*D2                                  ELE06190
        R(I+3)=R(I+3)+CTS(J,I)*D2                                               ELE06200
        R(I+6)=R(I+6)+CTE(J,I)*D3 - BE(J,I)*D4                                  ELE06210
  610   R(I+9)=R(I+9)+CTE(J,I)*D4                                               ELE06220
C                                                                               ELE06230
C-- --- INCRIMENTAL FORCES                                                     ELE06240
      F( 1)= TM( 2)*(R( 1)-R( 7))                                               ELE06250
      F( 2)= TM( 9)*(R( 2)-R( 8))+TM(10)*R( 6)+TM(11)*R(12)                     ELE06260
      F( 3)= TM(12)*(R( 3)-R( 9))-TM(13)*R( 5) TM(14)*R(11)                     ELE06270
      F( 4)= TM( 3)*(R( 4)-R(10))                                               ELE06280
      F( 5)=-TM(13)*(R( 3)-R( 9))-TM(7)*R( 5)+D*R(11)                           ELE06290
      F( 6)= TM(10)*(R( 2)-R( 8))+TM( 4)*R( 6)-TM( 6)*R(12)                     ELE06300
      F( 7)=-TM( 2)*(R( 1) R( 7))                                               ELE06310
      F( 8)=-TM( 9)*(R( 2) R( 8)) TM(10)*R( 6)-TM(11)*R(12)                     ELE06320
      F( 9)=-TM(12)*(R( 3)-R( 9))+TM(13)*R( 5)+TM(14)*R(11)                     ELE06330
      F(10)=-TM( 3)*(R( 4)-R(10))                                               ELE06340
      F(11)=-TM(14)*(R( 3)-R( 9))- D*R( 5)-TM( 8)*R(11)                         ELE06350
      F(12)= TM(11)*(R( 2)-R( 8))+ TM( 6)*R( 6)+TM( 5)*R(12)                    ELE06360
C                                                                               ELE06370
C------ SUBTRACT OF FIXED END FORCES                                            ELE06380
      IF (FEMFLG) THEN                                                          ELE06390
        DO 620 I=1,12                                                           ELE06400
  620     F(I)=F(I)-FEM(I,LOAD)                                                 ELE06410
      ENDIF                                                                     ELE06420
C                                                                               ELE06430
C---- - DETERMINE MAXIMUN VALUES FOR MULTIPLE LOAD CASES                        ELE06440
      IF (NLOAD.GT.1) THEN                                                      ELE06450
        DO 625 I=1,12                                                           ELE06460
          IF ( ABS(F(I)).GT.ABS(FMAX(I*12-12-I)) ) THEN                         ELE06470
            DO 626 J=0,11                                                       ELE06480
  626         FMAX(I*12-11+J)=F(J+1)                                            ELE06490
          ENDIF                                                                 ELE06500
  625   CONTINUE                                                                ELE06510
      ENDIF                                                                     ELE06520
C                                                                               ELE06530
C------ PRINT FORCES                                                           ELE06540
      IF (BTEST(IBUG,0).AND.PRINT) WRITE(108,493) IELNO,LOAD,                   ELE06550
     &                    NODEI,(R(J),J=1,6),NODEJ,(R(J),J=7,12)                ELE06560
      IF (PRINT) WRITE(108,492) IELNO,LOAD,NODEI,(F(J),J=1,6),                  ELE06570
     &                                NODEJ,(F(J),J=7,12)                       ELE06580
C                                                                               ELE06590
  630 CONTINUE                                                                  ELE06600
C                                                                               ELE06610
C---THIS IS FOR SOLOSA USE ONLY                                                 ELE06620
      IF(ICOMN .EQ. 1 .OR. ICOMN .EQ  2 ) THEN                                  ELE06630
        DO 631 I=1,12                                                           ELE06640
  631     RSAF(I)=P(I)                                                          ELE06650
      ENDIF                                                                     ELE06660
C                                                                               ELE06670
C                                                                               ELE06680
C------ CALCULATE MEMBER FORCE ON JOINT, AND ADD TO UNBALANCED FORCES           ELE06690
C     THIS IS USED TO CALC UNBALANCED FORCES.                                   ELE06700
       DO 640 I=1,3                                                             ELE06710
        I1=IDOF(JOINTI,I  )                                                     ELE06720
        I2=IDOF(JOINTI,I+3)                                                     ELE06730
        J1=IDOF(JOINTJ,I  )                                                     ELE06740
        J2=IDOF(JOINTJ,I+3)                                                     ELE06750
        DO 640 J=1,3                                                            ELE06760
          FUB(I1)=FUB(I1)                                                       ELE06770
          FUB(I2)=FUB(I2)                                                       ELE06780
          FUB(I2)=FUB(I2)                                                       ELE06790
          FUB(J1)=FUB(J1)                                                       ELE06800
          FUB(J2)=FUB(J2)                                                       ELE06810
  640     FUB(J2)=FUB(J2)                                                       ELE06820
C                                                                               ELE06830
  491 FORMAT (/'       3D BEAM FORCES.. ',5X/7X,A/,                             ELE06840
     & '   ELEMENT LOAD          NODE',7X,'AXIAL',11X,'FY',13X,'FZ',           ELE06850
     & 11X,'TORSION',10X,'MY',13X,'MZ')                                         ELE06860
  492 FORMAT (I10,I5,2X,'FORCE',I6,1P,6G15.6,                                   ELE06870
     &       /15X    ,2X,'FORCE',I6,   6G15.6)                                  ELE06880
  493 FORMAT (I10,I5,2X,'DISPL',I6,1P,6G15.6,                                   ELE06890
     &       /15X    ,2X,'DISPL',I6,   6G15.6)                                  ELE06900
  494 FORMAT (7X,A)                                                             ELE06910
      RETURN                                                                    ELE06920
C------------------------ DETERMINE TOTAL FORCES ------------------------------ELE06930
  500 CONTINUE                                                                  ELE06940
      IF (PRINT.AND.FIRST) WRITE(108,491) STEPID                                ELE06950
      IF(ICOMN .NE  1 .AND. ICOMN .NE. 2 ) THEN                                 ELE06960
C                                                                               ELE06970
C------ TOTAL FORCES AND DISPLACEMENTS                                          ELE06980
      DO 510 I=1,12                                                             ELE06990
         FT(I)=FT(I)+F(I)                                                       ELE07000
         RT(I)=RT(I)+R(I)                                                       ELE07010
  510 CONTINUE                                                                  ELE07020
C                                                                               ELE07030
      DO 512 I=1,12                                                             ELE07040
         IF ( ABS(FT(I)).GT.ABS(FMAX(I*12-12+I)) ) THEN                         ELE07050
            DO 511 J=0,11                                                       ELE07060
  511         FMAX(I*12-11+J)=FT(J+1)                                           ELE07070
         ENDIF                                                                  ELE07080
  512 CONTINUE                                                                  ELE07090
      ELSE IF(ICOMN .EQ  1 .OR. ICOMN .EQ. 2 ) THEN                             ELE07100
        DO 513 I=1,12                                                           ELE07110
  513     FT(I)=RSAF(I)                                                         ELE07120
      ENDIF                                                                     ELE07130
C                                                                               ELE07140
  520 FORMAT ('   I:',I3,' R:',1P,G12.4,' RT:',G12.4,                           ELE07150
     &        '  F:',1P,G12.4,' FT:',G12.4)                                     ELE07160
      LOAD=1                                                                    ELE07170
      IF (BTEST(IBUG,0).AND.PRINT) WRITE(108,493) IELNO,LOAD,                   ELE07180
     &                    NODEI,(RT(J),J=1,6),NODEJ,(RT(J),J=7,12)              ELE07190
      IF (PRINT) WRITE (108,492) IELNO,LOAD,NODEI,(FT(J),J=1,6),                ELE07200
     &                                NODEJ,(FT(J),J=7,12)                      ELE07210
C                                                                               ELE07220
      RETURN                                                                    ELE07230
C                                                                               ELE07240
C------------------------ DETERMINE FIXED END FORCES, AND ADD TO FORCE----------ELE07250
  600 IF (MAXELD.LE.0) RETURN                                                   ELE07260
C                                                                               ELE07270
C------- CALL BMLOA TO DETERMINE THE FIXED END FORCES                           ELE07280
      CALL BMLOA(IELNO,FEMFLG,LENGTH,REL,MAXELD,NELD,NLOAD,                     ELE07290
     &           FEM,IELD,ELD)                                                  ELE07300
      FEMFLG=FEMFLG                                                             ELE07310
      IF (.NOT  FEMFLG) RETURN                                                  ELE07320
C                                                                               ELE07330
C------- ADD FIXED END FORCES TO LOAD MATRIX                                    ELE07340
      DO 620 LOAD=1,NLOAD                                                       ELE07350
        DO 610 I=1,3                                                            ELE07360
        GFEM(I  )=0                                                             ELE07370
        GFEM(I+3)=0.                                                            ELE07380
        GFEM(I+6)=0                                                             ELE07390
        GFEM(I+9)=0.                                                            ELE07400
        DO 610 J=1,3                                                            ELE07410
           D1=FEM(J  ,LOAD)                                                     ELE07420
           D2=FEM(J+3,LOAD)                                                     ELE07430
           D3=FEM(J+6,LOAD)                                                     ELE07440
           D4=FEM(J+9,LOAD)                                                     ELE07450
         GFEM(I  )=GFEM(I  )+CTS(I,J)*D1                                        ELE07460
         GFEM(I+3)=GFEM(I+3)+ BS(I,J)*D1 + CTS(I,J)*D2                          ELE07470
         GFEM(I+6)=GFEM(I+6)+CTE(I,J)*D3                                        ELE07480
  610    GFEM(I+9)=GFEM(I+9)+ BE(I,J)*D3 + CTE(I,J)*D4                          ELE07490
C                                                                               ELE07500
      DO 620 J=1,6                                                              ELE07510
        JI=IDOF(JOINTI,J)                                                       ELE07520
        JJ=IDOF(JOINTJ,J)                                                       ELE07530
        FORCE(JI,LOAD)=FORCE(JI,LOAD)+GFEM(J  )                                 ELE07540
  620   FORCE(JJ,LOAD)=FORCE(JJ,LOAD)-GFEM(J+6)                                 ELE07550
C                                                                               ELE07560
      RETURN                                                                    ELE07570
C                                                                               ELE07580
C-------------------- WRITE MEMBER FORCES TO OUTPUT FILE ----------------------ELE07590
  700 CONTINUE                                                                  ELE07600
C                                                                               ELE07610
      WRITE(IUNIT,710)  NODEI,(FT(J),J=1,6),NODEJ,(FT(J),J=7,12)                ELE07620
      WRITE(IUNIT,710)  NODEI,(RT(J),J=1,6),NODEJ,(RT(J),J=7,12)                ELE07630
  710 FORMAT (I6,1P,6G12.5)                                                     ELE07640
C                                                                               ELE07650
      RETURN                                                                    ELE07660
C                                                                               ELE07670
C-------------------- READ AND PRINT MEMBER FORCES FROM OUTPUT FILE ----------ELE07680
  800 CONTINUE                                                                  ELE07690
      BACKSPACE(IUNIT)                                                          ELE07700
      READ(IUNIT,*  ) ITYPE,IELNO,IWRITE,TO,DT                                 ELE07710
      WRITE(108,805)                                                            ELE07720
      ISTEP=-IWRITE                                                             ELE07730
  810 READ (IUNIT,815,END=830) NODEI,(FT(J),J=1,6),NODEJ,(FT(J),J=7,12)         ELE07740
      READ (IUNIT,815,END=830) NODEI,(RT(J),J=1,6),NODEJ,(RT(J),J=7,12)         ELE07750
  815 FORMAT (I6,6G12.5)                                                        ELE07760
          ISTEP=ISTEP-IWRITE                                                    ELE07770
          T=TO+DT*ISTEP                                                         ELE07780
          IF (MOD(ISTEP,INC) EQ.0) THEN                                         ELE07790
            WRITE(108,820) ISTEP,T,NODEI,(FT(J),J=1,6),                         ELE07800
     &                             NODEJ,(FT(J),J=7,12)                       ELE07810
            WRITE(108,920) (RT(J),J=1,12)                                       ELE07820
          ENDIF                                                                 ELE07830
          GO TO 810                                                             ELE07840
C                                                                               ELE07850
  805 FORMAT (/'    3D BEAM FORCES ..',/                                        ELE07860
     &        '      STEP     TIME          NODE',7X,'AXIAL',11X,              ELE07870
     &        'FY',13X,'FZ', 11X,'TORSION',10X,'MY',13X,'MZ')                   ELE07880
  820 FORMAT (/,I10,1P,G15.5,0P,I6,1P,6G15.6,/,25X,I6,6G15.6)                   ELE07890
  920 FORMAT ( 25X,'DISP  ',1P,6G15.6,/31X,6G15.6)                              ELE07900
C                                                                               ELE07910
  830 RETURN                                                                    ELE07920
C                                                                               ELE07930
C-------------------- DETERMINE THE LENGTH OF THE MATERIAL DATA ---------------ELE07940
  900 CONTINUE                                                                  ELE07950
      NHYST=1                                                                   ELE07960
      ISE=0                                                                     ELE07970
C                                                                               ELE07980
        MAT  = RINPUT( 1)                                                       ELE07990
        CALL MATLIB(10,LSTTYP,FALSE,EESE,EPSE,DUCT,EXCR,                        ELE08000
     &   FT,F,RT,R,0.,0 ,LELEM,LMAT,MAT,MATTYP,SE,IELNO,DAMAGE)                ELE08010
        ISE=ISE+LELEM                                                           ELE08020
```

```
      RETURN                                                            ELE08030
C                                                                       ELE08040
C*-------------------- DETERMINE THE TOTAL STRAIN ENERGY                 ELE08050
 1000 CONTINUE                                                          ELE08060
      EESE=0                                                            ELE08070
      EPSE=0                                                            ELE08080
C                                                                       ELE08090
      DO 1010 I=1,12                                                    ELE08100
 1010   EESE=EESE + (FT(I)+P(I))*(R(I)+RT(I))/2.                       ELE08110
      ENGDAT(1) = EESE                                                  ELE08120
      ENGDAT(2) = 0.                                                    ELE08130
C                                                                       ELE08140
      RETURN                                                            ELE08150
C                                                                       ELE08160
C*-------------------- DUCTILITIES AND EXCURSION RATIOS                 ELE08170
C                     NO D & F RATIOS FOR ELASTIC ELEMENT               ELE08180
 1100 RETURN                                                            ELE08190
C                                                                       ELE08200
C                                                                       ELE08210
C*-------------------- PRINT MAXIMUM ELEMENT LOADS                      ELE08220
C                                                                       ELE08230
 1200 CONTINUE                                                          ELE08240
      LOAD2=0                                                           ELE08250
      IF (FIRST) WRITE(108,494) ' COLUMN ( LOAD) REPRESENTS THE ',      ELE08260
     &                          'DOF WHICH HAS MAXIMUM VALUE'           ELE08270
      IF (FIRST) WRITE(108,491) ' MAXIMUM VALUES FOR ALL STEPS  '       ELE08280
     &          //' NOTE: MAXIMUM VALUES WITH THE OTHER DOFS FORCES'    ELE08290
     &          //' ARE PRINT OUT AT THE SAME TIME'                     ELE08300
      DO 309 I=0,11                                                     ELE08310
         LOAD2=I+1                                                      ELE08320
         IF( PMAX(I*12+1+I)  EQ. 0. ) GO TO 309                         ELE08330
         WRITE(108,492) IELNO,LOAD2,NODE1,(PMAX(I*12+J),J=1,6),         ELE08340
     &                              NODEJ,(PMAX(I*12+J),J=7,12)         ELE08350
  309 CONTINUE                                                          ELE08360
C                                                                       ELE08370
      RETURN                                                            ELE08380
C                                                                       ELE08390
C                                                                       ELE08400
C*-------------------- RESET ELEMENT FORCES                             ELE08410
C                                                                       ELE08420
 1300 CONTINUE                                                          ELE08430
C----- ZERO MATRICIES                                                   ELE08440
       DO 1340 I=1,12                                                   ELE08450
         PMAX(I*12-11)=0                                                ELE08460
         P (I)=0                                                        ELE08470
         FT(I)=0                                                        ELE08480
         R (I)=0                                                        ELE08490
 1340    RT(I)=0                                                        ELE08500
      WRITE(108,1330) IELNO                                             ELE08510
 1330 FORMAT(' 3D BEAM. . .  ELEMENT #',I6,                             ELE08520
     & ' INTERNAL FORCES AND HYSTERESIS MODELS ARE RESET TO ZERO')      ELE08530
        FALSE=.FALSE.                                                   ELE08540
        LSTTYP=0                                                        ELE08550
        CALL MATLIB(2,LSTTYP,FALSE,EESE,EPSE,DUCT,EXCR,                 ELE08560
     &   PT,P,RT,R,0 ,0.,LELEM,LMAT,MAT,MATTYP,SE,IELNO,DAMAGE)         ELE08570
C                                                                       ELE08580
      RETURN                                                            ELE08590
C                                                                       ELE08600
C*-------------------- DAMAGE INDEX                                     ELE08610
C                                                                       ELE08620
 1400 DAMAGE = 0.                                                       ELE08630
C                                                                       ELE08640
      RETURN                                                            ELE08650
      END                                                               ELE08660
C                                                                       ELE08010
C-----------------------------------------------------------------------CELE00020
C                                                                       ELE00030
      SUBROUTINE ELE02                                                  ELE00040
     &  (IOPT       ,FIRST      ,PRINT      ,ID         ,IDOF       ,  ELE00050
     &   COORD      ,COSINE     ,JTCOS      ,CONST      ,DISPL      ,  ELE00060
     &   VELOC      ,FUB        ,IREL       ,NAME       ,IELNO      ,  ELE00070
     &   RINPUT     ,EESE       ,EPSE       ,DAMAGE     ,DUCFAG     ,  ELE00080
     &   IELTYP     ,IELDOF     ,NODE1      ,NODEJ      ,JOINTI     ,  ELE00090
     &   JOINTJ     ,R          ,RT         ,F          ,FT         ,  ELE00100
     &   V          ,VT         ,ISE        ,H          ,LENGTH     ,  ELE00110
     &   MAT        ,A          ,LM         ,KTYPE      ,PMAX       ,  ELE00120
     &   SE         ,STIFF      )                                      ELE00130
C                                                                       ELE00140
      IMPLICIT REAL(A-H,O-Z)                                            ELE00150
C                                                                       ELE00160
      COMMON /RSA/ICOMN, RSAP(12)                                       ELE00170
      REAL ICOMN,LENGTH                                                 ELE00180
      LOGICAL FIRST,PRINT,FALSE,BTEST                                   ELE00190
      CHARACTER*80 NAME                                                 ELE00200
      CHARACTER*9  TYPE(6)                                              ELE00210
$INCLUDE:'ZCOMN3'                                                       ELE00220
      DIMENSION IDOF(MAXNOD,6),COORD(MAXNOD,6),COSINE(3,3,NCOS)         ELE00230
      DIMENSION CONST(MAXNOD,6),ID(MAXNOD),JTCOS(MAXNOD),SAT(2,12)      ELE00240
      DIMENSION STIFF(78),A(12,2),S(2,2),LM(12),PMAX(2)                 ELE00250
      DIMENSION CTS( 3, 3),CTE( 3, 3),VX(3),VY(3)                       ELE00260
      DIMENSION BS( 3, 3),BE( 3, 3),SE(ISE)                             ELE00270
      DIMENSION DISPL(NDOF,NLOAD),FUB(NDOF),VELOC(NDOF)                 ELE00280
      DIMENSION RINPUT(100),WORK(6),DUCT(3),EXCR(6)                     ELE00290
      DATA TYPE/'    AXIAL','  SHEAR-Y','  SHEAR-Z',                   ELE00300
     &         '  TORSION','BENDING-Y','BENDING-Z'/                     ELE00310
C                                                                       ELE00320
C-------------------------------------- .. ------------------------------ELE00330
C VARIABLES                                                             ELE00340
C-----------------------------------------------------------------------CELE00350
C----- ---- GLOBAL VARIABLES                                           ELE00360
C IOPT   - 1, INITIALIZE BEAM COLUMN ELEMENT                            ELE00370
C        - 2, CALCULATE GEOMETRIC STIFFNESS                             ELE00380
C        - 3, FORM STIFFNESS                                            ELE00390
C        - 4, CALCULATE INCREMENTAL FORCES                              ELE00400
C        - 5, CALCULATE TOTAL FORCES AND ENERGIES                       ELE00410
C IUNIT  - OUTPUT FILE UNIT # FOR PRINTING OUTPUT                       ELE00420
C FIRST  - FLAG, FIRST=.TRUE , PRINT HEADERS                            ELE00430
C PRINT  - FLAG, PRINT=.TRUE., PRINT DATA                               ELE00440
C NDOF   - # OF GLOBAL DOF                                              ELE00450
C NLOAD  - # OF LOAD COMBINATIONS                                       ELE00460
C MAXNOD - # ROW DIMENSION OF NODE ARRAY                                ELE00470
C NNODE  - # OF NODES                                                   ELE00480
C ID     - ARRAY OF EXTERNAL NODE NUMBERS                               ELE00490
C IDOF   - ARRAY OF DEGREES OF FREEDOM                                  ELE00500
C COORD  - ARRAY OF COORDINATES                                         ELE00510
C COSINE - ARRAY OF DIRECTION COSINES OF NODES                          ELE00520
C CONST  - ARRAY OF CONSTRAINT TRANSFORMATIONS                          ELE00530
C DISPL  - GLOBAL DISPLACEMENT MATRIX                                   ELE00540
C Z      - REAL GLOBAL STORTAG VECTOR                                   ELE00550
C NZ     - INTEGER GLOBAL STORTAG VECTOR                                ELE00560
C IZMAT  - LOCATION OF MATERIAL DATA IN GLOBAL STORAGE VECTOR           ELE00570
C NAME   - ELEMENT NAME                                                 ELE00580
C IELNO  - ELEMENT NUMBER                                               ELE00590
C RINPUT - INPUT DATA                                                   ELE00600
C STIFF  - OUTPUT ELEMENT STIFFNESS (UPPER TRIANGULAR FORM)             ELE00610
C LM     - OUTPUT LOCAL TO GLOBAL STIFFNESS MAPPING MATRIX              ELE00620
C--- ------ ELEMENT VARIABLES                                          ELE00630
C H,Q,AI,AJ,C,BI,BJ,S22,S26,S212,S33,S35,S311,D                         ELE00640
C        - INDIVIDUAL TERMS IN THE ELEMENT STIFFNESS (S) MATRIX         ELE00650
C PKG    - INPUT LOAD FOR FORMING GEOMETRIC STIFFNESS MATRIX...         ELE00660
C R      - INCRIMENTAL DISPLACEMENTS                                    ELE00670
C RT     - TOTAL DISPLACEMENTS                                          ELE00680
C F      - INCRIMENTAL FORCES                                           ELE00690
C FT     - TOTAL FORCES                                                 ELE00700
C LM     - LOCAL TO GLOBAL STIFFNESS MAPPING MATRIX                     ELE00710
C CTS    - LOCAL TO GLOBAL ROTATION MATRIX START                        ELE00720
C CTE    - LOCAL TO GLOBAL ROTATION MATRIX END                          ELE00730
C BS     - LOCAL TO GLOBAL ROTATED ECCENTRICITY/CONSTRAINT  MATRIX STARTELE00740
C BE     - LOCAL TO GLOBAL ROTATED ECCENTRICITY/CONSTRAINT  MATRIX END  ELE00750
C STIFF  - ELEMENT STIFFNESS                                            ELE00760
C IELTYP - ELEMENT TYPE - 10                                            ELE00770
C NODE1  - EXTERNAL JOINT NUMBER, END I                                 ELE00780
C NODEJ  - EXTERNAL JOINT NUMBER, END J                                 ELE00790
C JOINTI - INTERNAL JOINT NUMBER, END I                                 ELE00800
C JOINTJ - INTERNAL JOINT NUMBER, END J                                 ELE00810
C--- ------ TEMPROARY VARIABLES                                        ELE00820
C VX     - VECTOR DEFINING THE ELEMENT X AXIS                           ELE00830
C VY     - VECTOR DEFINING THE ELEMENT Y AXIS                           ELE00840
C COSLOC - LOCAL COSINE MATRIX                                          ELE00850
C A      - GLOBAL TO LOCAL FORCE TRANSFORMATION MATRIX                  ELE00860
C S      - ELEMENT STIFFNESS AT LOCAL COORD, AT ENDS OF FLEXABLE PART   ELE00870
C SAT    - PRODUCT OF S*A'                                              ELE00880
C ASAT   - PRODUCT OF A*SAT                                             ELE00890
C-----------------------------------------------------------------------CELE00900
C                                                                       ELE00910
C                                                                       ELE00920
C--- --- CHOOSE OPTION                                                 ELE00930
      IF (IOPT.LT.1 .OR IOPT.GT.14)                                     ELE00940
     &     WRITE(108,*) 'INVALID OPTION IN ELE-02, IOPT=',IOPT          ELE00950
C                                                                       ELE00960
      GO TO (100,200,300,400,500,600,700,800,900,1000,1100,             ELE00970
     &                                  1200,1300,1400) IOPT            ELE00980
C                                                                       ELE00990
  100 CONTINUE                                                          ELE01000
C                                                                       ELE01010
C----- INTERPERTATE INPUT DATA                                          ELE01020
      IELTYP= 2                                                         ELE01030
      MAT   = RINPUT( 1)                                                ELE01040
      NODEI = RINPUT( 2)                                                ELE01050
      NODEJ = RINPUT( 3)                                                ELE01060
      KTYPE = RINPUT( 4)                                                ELE01070
      XLEN  = RINPUT( 5)                                                ELE01080
      VY(1) = RINPUT( 6)                                                ELE01090
      VY(2) = RINPUT( 7)                                                ELE01100
      VY(3) = RINPUT( 8)                                                ELE01110
      XS    = RINPUT( 9)                                                ELE01120
      XE    = RINPUT(10)                                                ELE01130
C                                                                       ELE01140
C  --- GET INTERNAL NODE # ASSOCIATED WITH EXTERNAL NODE #'S            ELE01150
      JOINTI=IQUICK(NODEI,ID,NNODE)                                     ELE01160
      JOINTJ=IQUICK(NODEJ,ID,NNODE)                                     ELE01170
C                                                                       ELE01180
      IF (KTYPE.LT.1 .OR  KTYPE.GT.6) THEN                              ELE01190
         WRITE(108,102) IELNO,NODEI,NODEJ,KTYPE                         ELE01200
  102    FORMAT(1X,20('*'),'ER','ROR - INVALID KTYPE',/5X,'IELNO=',     ELE01210
     &          I5,5X,'JOINTI=',I5,5X,'JOINTJ=',I5,5X,'KTYPE=',I5,/    ELE01220
     &          5X,'REVISE INPUT  1 LE KTYPE LE 6 ...     .....'/)     ELE01230
         LENGTH=1.0                                                     ELE01240
      ENDIF                                                             ELE01250
C                                                                       ELE01260
C----- PRINT DATA FOR PLOTTING                                          ELE01270
      IF (BTEST(IBUG,2)) WRITE(7,10) NODEJ,NODEI,(COORD(JOINTJ,I),      ELE01280
     &                                I=1,3),(COORD(JOINTI,J),J=1,3)    ELE01290
   10  FORMAT (I5,1X,I5,1P,6G12.4)                                      ELE01300
C                                                                       ELE01310
C-----------------------------------------------------------------------CELE01320
C  GET GLOBAL TO LOCAL TRANSFORMATION MATRIX                            ELE01330
C-----------------------------------------------------------------------CELE01340
C----- DEFINE VECTOR X, LENGTH                                          ELE01350
      VX(1)=COORD(JOINTJ,1)-COORD(JOINTI,1)                             ELE01360
      VX(2)=COORD(JOINTJ,2)-COORD(JOINTI,2)                             ELE01370
      VX(3)=COORD(JOINTJ,3)-COORD(JOINTI,3)                             ELE01380
      LENGTH=SQRT(VX(1)**2+VX(2)**2+VX(3)**2)-XS-XE                     ELE01390
      IF (LENGTH.EQ.0) THEN                                             ELE01400
        DO 103 II=1,3                                                   ELE01410
          VX(II)= COSINE(1,II,JTCOS(JOINTI))                            ELE01420
  103     VY(II)= COSINE(2,II,JTCOS(JOINTI))                            ELE01430
        LENGTH=1.00                                                     ELE01440
      ELSE                                                              ELE01450
        LENGTH=LENGTH-XS-XE                                             ELE01460
      ENDIF                                                             ELE01470
      IF (XLEN.NE.0) THEN                                               ELE01480
         LENGTH=XLEN                                                    ELE01490
         XS=0                                                           ELE01500
         XE=0                                                           ELE01510
      ENDIF                                                             ELE01520
      IF (LENGTH.LE.0) THEN                                             ELE01530
         WRITE(108,101) IELNO,JOINTI,JOINTJ,LENGTH                      ELE01540
  101    FORMAT(1X,20('*'),'ER','ROR - LENGTH IS LE 0',/5X,'IELNO=',    ELE01550
     &       I5,5X,'JOINTI=',I5,5X,'JOINTJ=',I5,5X,'LENGTH',1P,G15.6/   ELE01560
     &       5X,'REVISE INPUT.....     ... .... - .....'/)             ELE01570
         LENGTH=1.0                                                     ELE01580
      ENDIF                                                             ELE01590
C----- GET LOCAL TO GLOBAL TRANSFORMATION MATRICIES CT AND B            ELE01600
      ICS=JTCOS(JOINTI)                                                 ELE01610
      ICE=JTCOS(JOINTJ)                                                 ELE01620
      CALL ROTXYZ(VX,VY,COSINE(1,1,ICS),CTS, XS ,0. ,0. ,BS,            ELE01630
     &            CONST(JOINTI,1),MAXNOD)                               ELE01640
      CALL ROTXYZ(VX,VY,COSINE(1,1,ICE),CTE,-XE ,0. ,0. ,BE,            ELE01650
     &            CONST(JOINTJ,1),MAXNOD)                               ELE01660
C                                                                       ELE01670
C----- GET GLOBAL CONSTRAINT MATRIX BS, BE                              ELE01680
C                                                                       ELE01690
C                                                                       ELE01700
C----- ZERO MATRICIES                                                   ELE01710
        F =0                                                            ELE01720
        FT=0                                                            ELE01730
        R =0                                                            ELE01740
        RT=0                                                            ELE01750
        V =0                                                            ELE01760
        VT=0                                                            ELE01770
        PMAX(1)=0                                                       ELE01780
        PMAX(2)=0                                                       ELE01790
C                                                                       ELE01800
C----- ASSEMBLE TRANSFORMATION MATRIX A                                 ELE01810
C  NOTE. A IS DIMMENSIONED AS A 12X2 MATRIX, BUT ONLY THE FIRST IELDOF  ELE01820
C        ROW'S CONTAIN INFORMATION. THE BALANCE OF THE MATRIX IS        ELE01830
C        UNDEFINED.  THE GLOBAL DOF COORESPONDING TO THE ROW'S OF A     ELE01840
C        ARE STORED IN VECTOR LM.  AGAIN, ONLY THE FIRST IELDOF ROW'S   ELE01850
C        OF LM ARE DEFINED                                              ELE01860
        IELDOF=0                                                        ELE01870
        DO 50 I=1,12                                                    ELE01880
C                                                                       ELE01890
C               REMOVE DOF'S THAT ARE CONSTRAINED TO THE SAME DOF.      ELE01900
          J=I                                                           ELE01910
          IF (I.GT.6) J= I-6                                            ELE01920
          IF ( IDOF(JOINTI,J) EQ.IDOF(JOINTJ,J) ) GO TO 50              ELE01930
          IF (I.LE.6) THEN                                              ELE01940
              LMTI=IDOF(JOINTI,I)                                       ELE01950
          ELSE                                                          ELE01960
              LMTI=IDOF(JOINTJ,J)                                       ELE01970
          ENDIF                                                         ELE01980
C                                                                       ELE01990
          A1=0                                                          ELE02000
          A2=0                                                          ELE02010
          IF (I.LE.3) THEN                                              ELE02020
              IF (KTYPE.LE.3) A1=CTS(I,KTYPE)                           ELE02030
          ELSE IF (I.LE.6) THEN                                         ELE02040
              IF (KTYPE.LE.3) THEN                                      ELE02050
                  A1=BS(I-3,KTYPE)                                      ELE02060
              ELSE                                                      ELE02070
                  A1=CTS(I-3,KTYPE-3)                                   ELE02080
              ENDIF                                                     ELE02090
          ELSE IF (I.LE.9) THEN                                         ELE02100
              IF (KTYPE.LE.3) A2=CTE(I-6,KTYPE)                         ELE02110
          ELSE                                                          ELE02120
              IF (KTYPE.LE.3) THEN                                      ELE02130
                  A2=BE(I-9,KTYPE)                                      ELE02140
              ELSE                                                      ELE02150
                  A2=CTE(I-9,KTYPE-3)                                   ELE02160
              ENDIF                                                     ELE02170
          ENDIF                                                         ELE02180
          IF (A1.EQ.0 .AND  A2.EQ.0) GO TO 50                           ELE02190
          IELDOF=IELDOF+1                                               ELE02200
          A(IELDOF,1)=A1                                                ELE02210
          A(IELDOF,2)=A2                                                ELE02220
          LM(IELDOF)=LMTI                                               ELE02230
C                                                                       ELE02240
   50     CONTINUE                                                      ELE02250
C                                                                       ELE02260
C----- GET SPRING STIFFNESS                                             ELE02270
        FALSE=.FALSE                                                    ELE02280
```

```
      LSTTYP=0                                                          ELE02290
      CALL MATLIB(2,LSTTYP,FALSE,EESE,EPSE,DUCT,EXCR,                   ELE02300
     &     FT,F,RT,R,0.,0.,LELEM,LMAT,MAT,MATTYP,SE,IELNO,DAMAGE)       ELE02310
C                                                                       ELE02320
C--- --- H IS THE SPRING STIFFNESS                                      ELE02330
C                                                                       ELE02340
      H = SE(1)                                                         ELE02350
      HH= SE(1)/LENGTH                                                  ELE02360
      S(1,1)= HH                                                        ELE02370
      S(1,2)=-HH                                                        ELE02380
      S(2,1)=-HH                                                        ELE02390
      S(2,2)= HH                                                        ELE02400
C                                                                       ELE02410
C----- CALCULATE SAT                                                    ELE02420
      DO 60 I=1,2                                                       ELE02430
        DO 60 J=1,IELDOF                                                ELE02440
        SAT(I,J)=0                                                      ELE02450
          DO 60 K=1,2                                                   ELE02460
   60       SAT(I,J)=SAT(I,J)+S(I,K)*A(J,K)                             ELE02470
C                                                                       ELE02480
C ---- CALCULATE ASAT AND CONVERT TO HALF STORAGE MODE  BY COLUMNS       ELE02490
C                                                                       ELE02500
C          1  3  6 10 15 21 28 36 45 55 66 78  NUMBER INDICATES        ELE02510
C             2  5  9 14 20 27 35 44 54 65 77  LOCATION OF DATA         ELE02520
C                4  8 13 19 26 34 43 53 64 76  IN ARRAY STIFF          ELE02530
C                   7 12 18 25 33 42 52 63 75                           ELE02540
C                  11 17 24 32 41 51 62 74                              ELE02550
C                     16 23 31 40 50 61 73                              ELE02560
C                        22 30 39 49 60 72                              ELE02570
C                           29 38 48 59 71                              ELE02580
C                              37 47 58 70                              ELE02590
C                                 46 57 69                              ELE02600
C                                    56 68                              ELE02610
C                                       67                              ELE02620
C                                                                       ELE02630
      L=0                                                               ELE02640
      DO 70 J=1,IELDOF                                                  ELE02650
        DO 70 I=J,1,-1                                                  ELE02660
        L=L+1                                                           ELE02670
        STIFF(L)=0                                                      ELE02680
          DO 70 K=1,2                                                   ELE02690
   70       STIFF(L)=STIFF(L)+A(I,K)*SAT(K,J)                           ELE02700
C                                                                       ELE02710
C----- RELEASE UNUSED STORAGE                                           ELE02720
  180 IREL=78-L                                                         ELE02730
C                                                                       ELE02740
C     PRINT COLUMN DATA                                                 ELE02750
      IF (FIRST) WRITE(108,192)                                         ELE02760
  191 WRITE(108,193) NAME,IELNO,MAT,NODEI,NODEJ,TYPE(KTYPE),LENGTH,     ELE02770
     &                VY(1),VY(2),VY(3),XS,-XE                         ELE02780
C                                                                       ELE02790
  192 FORMAT(/' ELEMENT 02, ONE (LOCAL) DOF SPRING ELEMENT'//           ELE02800
     &  T22,'# MATL START   END',2X,'---TYPE----',6X,'LENGTH',3X,        ELE02810
     & 11('-'),'  Y-AXIS  ',12('-'),'  START DIST    END DIST')         ELE02820
  193 FORMAT(1X,A15,4I6,2X,A9,3X,1P,G12.4,0P,F9.5,' I',                 ELE02830
     &     5P,F9.5,' J',F9.5,' X',1P,SS,3G12.4)                         ELE02840
      RETURN                                                            ELE02850
C----------------------- DETERMINE GEOMETRIC STIFF ---------------------ELE02860
  200 RETURN                                                            ELE02870
C     THE GEOMETRIC STIFFNESS MATRIX IS NOT AVIALABLE FOR THIS ELEMENT  ELE02880
C                                                                       ELE02890
C----------------------- DETERMINE STIFFNESS ---------------------------ELE02900
  300 CONTINUE                                                          ELE02910
C                                                                       ELE02920
C  ---- H CONTAINS THE SPRING STIFFNESS, WHICH WAS DETERMINED IN THE    ELE02930
C       LAST CALL TO MAT_LIB                                            ELE02940
C       IF H=SE(1) THEN THE STIFFNESS HAS NOT CHANGED, RETURN           ELE02950
C                                                                       ELE02960
      IF (H.EQ.SE(1)) RETURN                                            ELE02970
C                                                                       ELE02980
      H = SE(1)                                                         ELE02990
      HH= SE(1)/LENGTH                                                  ELE03000
      S(1,1)= HH                                                        ELE03010
      S(1,2)=-HH                                                        ELE03020
      S(2,1)= HH                                                        ELE03030
      S(2,2)= HH                                                        ELE03040
C                                                                       ELE03050
C----- CALCULATE SAT                                                    ELE03060
      DO 360 I=1,2                                                      ELE03070
        DO 360 J=1,IELDOF                                               ELE03080
        SAT(I,J)=0                                                      ELE03090
          DO 360 K=1,2                                                  ELE03100
  360       SAT(I,J)=SAT(I,J)+S(I,K)*A(J,K)                             ELE03110
C                                                                       ELE03120
C ---- CALCULATE ASAT, STORE IN UPPER TRIANGULAR MATRIX                 ELE03130
      L=0                                                               ELE03140
      DO 370 J=1,IELDOF                                                 ELE03150
        DO 370 I=J,1, 1                                                 ELE03160
        L=L+1                                                           ELE03170
        STIFF(L)=0                                                      ELE03180
          DO 370 K=1,2                                                  ELE03190
  370       STIFF(L)=STIFF(L)+A(I,K)*SAT(K,J)                           ELE03200
C                                                                       ELE03210
      RETURN                                                            ELE03220
C----------------------- DETERMINE INCRIMENTAL FORCES ------------------ELE03230
  400 CONTINUE                                                          ELE03240
      IF (PRINT.AND.FIRST) WRITE(108,491) STEPID                        ELE03250
C                                                                       ELE03260
C------ LOOP FOR EACH LOAD                                             ELE03270
      DO 430 LOAD=1,NLOAD                                               ELE03280
C                                                                       ELE03290
C------ TRANSFORM FRAME DISPL INTO LOCAL DISPL                          ELE03300
C... LOCAL DISPL ARE AT THE END END OF THE SPRING, THUS                 ELE03310
C... POSITIVE IS TENSION AND NEGATIVE IS COMPRESSION                    ELE03320
        R=0                                                             ELE03330
        V=0                                                             ELE03340
        DO 410 I=1,IELDOF                                               ELE03350
          J=LM(I)                                                       ELE03360
            AI1= A(I,1)                                                 ELE03370
            AI2= A(I,2)                                                 ELE03380
            DISP= DISPL(J,LOAD)                                         ELE03390
            VEL= VELOC(J)                                               ELE03400
          R= R + (A(I,2) A(I,1)) * DISPL(J,LOAD)                        ELE03410
  410     V= V + (A(I,2)-A(I,1)) * VELOC(J)                             ELE03420
C                                                                       ELE03430
C------ INCRIMENTAL FORCES                                             ELE03440
      F= H*R/LENGTH                                                     ELE03450
C                                                                       ELE03460
C------ CHECK STIFFNESS AND CALCULATE UNBALANCED FORCES                 ELE03470
C       THIS IS PREFORMED ONLY FOR INELASTIC ANALYSIS                   ELE03480
      P =FT-F                                                           ELE03490
      IF (.NOT. ELSTIC) THEN                                            ELE03500
C           - SET UP TEMPORARY - TOTAL LOADS, DISPL'S AND VELOC         ELE03510
          PL=FT                                                         ELE03520
          D =(RT+R)/LENGTH                                              ELE03530
          DL=RT /LENGTH                                                 ELE03540
          VC=VT+V                                                       ELE03550
          VL=VT                                                         ELE03560
C         -- TRANSFER PERMANENT HYSTERESIS DATA TO TEMPROARY STORAGE.    ELE03570
          ISE2=ISE/2                                                    ELE03580
          DO 415 I=1,ISE2                                               ELE03590
  415       SE(I+ISE2)=SE(I)                                            ELE03600
C         -- CALL HYST MODEL TO GET NEW STIFFNESS AND                   ELE03610
C            THE TOTAL FORCE P AT DISPL D                               ELE03620
          CALL MATLIB(3,LSTTYP,FALSE,EESE,EPSE,DUCT,EXCR,P,PL,D,DL,     ELE03630
     &           VC,VL,LELEM,LMAT,MAT,MATTYP,SE(ISE2+1),IELNO,DAMAGE)   ELE03640
        IF (SE(ISE2+1) .NE.SE(1)) NEWX=.TRUE.                           ELE03650
      ENDIF                                                             ELE03660
C                                                                       ELE03670
C------ SAVE PEAK VALUES FOR MULTIPLE LOAD CASES                        ELE03680
      IF (NLOAD.GT.1  AND.  ABS(F) GT.ABS(FMAX(1)) ) THEN               ELE03690
        FMAX(1)=F                                                       ELE03700
        FMAX(2)=R                                                       ELE03710
      ENDIF                                                             ELE03720
C                                                                       ELE03730
C------ PRINT DATA                                                      ELE03740
      HH= H/LENGTH                                                      ELE03750
      IF (PRINT) WRITE(108,492) IELNO,LOAD,TYPE(KTYPE),NODEI,NODEJ,     ELE03760
     1                          F,R,HH                                  ELE03770
C                                                                       ELE03780
  430 CONTINUE                                                          ELE03790
C                                                                       ELE03800
C------ THIS IS FOR SOL65A USED ONLY                                    ELE03810
      IF(ICOMN .EQ. 1 .OR. ICOMN .EQ  2 ) THEN                          ELE03820
        RSAP(1)=F                                                       ELE03830
      ENDIF                                                             ELE03840
C                                                                       ELE03850
C------ CALCULATE THE UNBALANCED MEMBER FORCE ON A JOINT.               ELE03860
            UBAL= F+FT-P                                                ELE03870
            F   = F                                                     ELE03880
            FT  = FT                                                    ELE03890
            P   = P                                                     ELE03900
            DO 440 I=1,IELDOF                                           ELE03910
               J=LM(I)                                                  ELE03920
               AI1= A(I,1)                                              ELE03930
               AI2= A(I,2)                                              ELE03940
               FUB(J)= FUB(J) + (A(I,1)-A(I,2)) * UBAL                  ELE03950
  440       CONTINUE                                                    ELE03960
C                                                                       ELE03970
  491 FORMAT (/'       SPRING FORCES...',5X,A/                          ELE03980
     & '   ELEMENT LOAD',T17,'      TYPE',' NODEI',' NODEJ',           ELE03990
     & '       FORCE     ',' DISPLACMENT   ','    STIFFNESS    ')      ELE04000
  492 FORMAT (I10,I5,2X,A8,2I6,1P,3G15.6)                               ELE04010
C                                                                       ELE04020
C------ ADJUST F FOR UNBALANCED LOAD P                                  ELE04030
          F=P-FT                                                        ELE04040
      RETURN                                                            ELE04050
C----------------------- DETERMINE TOTAL FORCES, ENERGIES ECT ----------ELE04060
  500 CONTINUE                                                          ELE04070
      IF( ICOMN .NE. 1 .AND. ICOMN .NE. 2 ) THEN                        ELE04080
       IF (PRINT.AND.FIRST) WRITE(108,491) STEPID                       ELE04090
C                                                                       ELE04100
C                                                                       ELE04110
C------ TOTAL FORCES, DISPLACEMENTS AND VELOCITIES                      ELE04120
        FT=FT+F                                                         ELE04130
        RT=RT+R                                                         ELE04140
        VT=VT+V                                                         ELE04150
        IF ( ABS(FT) GT.ABS(FMAX(1)) ) THEN                             ELE04160
          FMAX(1)=FT                                                    ELE04170
          FMAX(2)=RT                                                    ELE04180
        ENDIF                                                           ELE04190
C                                                                       ELE04200
C------ PRINT DATA                                                      ELE04210
        HH= H/LENGTH                                                    ELE04220
        IF (PRINT) WRITE(108,492) IELNO,1,TYPE(KTYPE),NODEI,NODEJ,      ELE04230
     1                            FT,RT,HH                              ELE04240
      ELSE IF( ICOMN .EQ. 1 .OR  ICOMN .EQ. 2 ) THEN                    ELE04250
        FT=RSAP(1)                                                      ELE04260
        IF (PRINT.AND.FIRST) WRITE(108,497) STEPID                      ELE04270
        IF (PRINT) WRITE(108,492) IELNO,1,TYPE(KTYPE),NODEI,NODEJ,FT    ELE04280
      ENDIF                                                             ELE04290
C                                                                       ELE04300
  497 FORMAT (/'       SPRING FORCES...',5X,A/                          ELE04310
     & '   ELEMENT LOAD',T17,'      TYPE',' NODEI',' NODEJ',           ELE04320
     & '       FORCE     ')                                            ELE04330
C                                                                       ELE04340
C------ TRANSFER TEMPROARY HYSTERESIS DATA TO PERMANENT STORAGE...      ELE04350
      ISE2=ISE/2                                                        ELE04360
      IF (.NOT. ELSTIC) THEN                                            ELE04370
        DO 510 I=1,ISE2                                                 ELE04380
  510     SE(I)=SE(I+ISE2)                                              ELE04390
      ENDIF                                                             ELE04400
C                                                                       ELE04410
      RETURN                                                            ELE04420
C                                                                       ELE04430
C----------------------- DETERMINE FIXED END FORCES, AND ADD TO FORCE---ELE04440
  600 RETURN                                                            ELE04450
C                                                                       ELE04460
C-- ---- FIXED END FORCES ARE NOT AVAILABLE FOR THE SPRING              ELE04470
C                                                                       ELE04480
C---------------------- WRITE MEMBER FORCES TO OUTPUT FILE -------------ELE04490
  700 CONTINUE                                                          ELE04500
C                                                                       ELE04510
      CALL MATLIB(4,LSTTYP,FALSE,EESE,EPSE,DUCT,EXCR,                   ELE04520
     &   FT,F,RT,R,0.,0.,LELEM,LMAT,MAT,MATTYP,SE,IELNO,DAMAGE)        ELE04530
C                                                                       ELE04540
      WRITE (IUNIT,710)  NODEI,NODEJ,KTYPE,FT,RT,H,EESE,EPSE,           ELE04550
     &                   (DUCT(I),EXCR(I),I=1,3)                       ELE04560
  710 FORMAT (3I6,1P,5G12.4/6G12.4)                                     ELE04570
C                                                                       ELE04580
      RETURN                                                            ELE04590
C                                                                       ELE04600
C---------------------- READ AND PRINT MEMBER FORCES FROM OUTPUT FILE --ELE04610
  800 CONTINUE                                                          ELE04620
      BACKSPACE(IUNIT)                                                  ELE04630
      READ (IUNIT,*  ) ITYPE,IELNO,IWRITE,TO,DT                        ELE04640
      ISTEP=-IWRITE                                                     ELE04650
C                                                                       ELE04660
  810 READ (IUNIT,815,END=830) NODEI,NODEJ,KTYPE,FT,RT,H,ESE,PSE,       ELE04670
     &                         (WORK(I),I=1,6)                        ELE04680
  815 FORMAT (3I6,5G12.4/6G12.4)                                        ELE04690
        ISTEP=ISTEP+IWRITE                                              ELE04700
        T=TO+DT*ISTEP                                                   ELE04710
        IF (ISTEP.EQ.0) WRITE(108,805) NODEI,NODEJ,TYPE(KTYPE)          ELE04720
        IF (MOD(ISTEP,INC).EQ.0) WRITE(108,820) ISTEP,T,FT,RT,          ELE04730
     &                                          H,ESE,PSE               ELE04740
        GO TO 810                                                       ELE04750
C                                                                       ELE04760
  805 FORMAT (/'       SPRING FORCES...',/                              ELE04770
     & 5X,'NODEI -',I6,5X,'NODEJ -',I6,5X,'SPRING TYPE -',A,//          ELE04780
     & '     STEP      TIME         ',                                 ELE04790
     & '     FORCE     ',' DISPLACMENT ',' STIFFNESS',                  ELE04800
     & '     ESE       ',' PSE       ')                               ELE04810
  820 FORMAT (I10,1P,G15.5,6G13.5)                                      ELE04820
C                                                                       ELE04830
  830 WRITE(108,840) (WORK(I),I=1,6)                                    ELE04840
  840 FORMAT (/T10,' MAXIMUN DUCTILITIES AND EXCURSION RATIOS'/         ELE04850
     &        T10,'-----------------------------------------'/         ELE04860
     &  T15,'DISPLACEMENT DEFINATION: U1=',1P,G12.4,5X,'E1=',G12.4/     ELE04870
     &  T15,'ENERGY DEFINATION #1:    U2=',1P,G12.4,5X,'E2=',G12.4/     ELE04880
     &  T15,'ENERGY DEFINATION #2:    U3=',1P,G12.4,5X,'E3=',G12.4)     ELE04890
      RETURN                                                            ELE04900
C                                                                       ELE04910
C------------------- DETERMINE THE LENGTH OF THE MATERIAL DATA ---------ELE04920
  900 CONTINUE                                                          ELE04930
      NMYST=1                                                           ELE04940
      ISE=0                                                             ELE04950
C                                                                       ELE04960
        MAT   = RINPUT( 1)                                              ELE04970
        CALL MATLIB(0,LSTTYP,FALSE,EESE,EPSE, DUCT,EXCR,                ELE04980
     &   FT,F,RT,R,0.,0.,LELEM,LMAT,MAT,MATTYP,SE,IELNO,DAMAGE)         ELE04990
        ISE=ISE+LELEM                                                   ELE05000
C                                                                       ELE05010
C------  DOUBLE THE STORAGE, TO ALLOW FOR TEMPROARY VALUES.             ELE05020
        ISE=ISE*2                                                       ELE05030
      RETURN                                                            ELE05040
C                                                                       ELE05050
C------------------- DETERMINE THE STRAIN ENERGY ----------------------ELE05060
 1000 CONTINUE                                                          ELE05070
C                                                                       ELE05080
C-------------------- DUCTILITIES AND EXCURSION RATIO'S ----------------ELE05090
 1100 CONTINUE                                                          ELE05100
C .  OPTIONS 10 AND 11 HAVE THE SAME CODE FOR THIS ELEMENT..            ELE05110
C... THE ENERGY WAS CALCULATED WITH THE LAST CALL FOR INCRIMENTAL FORCE ELE05120
C     THE TOTAL FORCES HAVE NOT BEEN CALLED YET,THUS THE ENERGIES RESIDE ELE05130
C     IN THE MATRIX SE BEGINING AT ADDRESS ISE21.                       ELE05140
C                                                                       ELE05150
      ISE21=ISE/2 +1                                                    ELE05160
      CALL MATLIB(4,LSTTYP,FALSE,EESE,EPSE, DUCT,EXCR,                  ELE05170
     &   FT,F,RT,R,0.,0 ,LELEM,LMAT,MAT,MATTYP,SE(ISE21),IELNO,DAMAGE)  ELE05180
C                                                                       ELE05190
      IF ( DUCFAG .EQ. 0 ) THEN                                         ELE05200
        IF( ABS(DUCT(1))  LT  1 ) DUCFAG=C                              ELE05200
```

```
        IF( ABS(DUCT(1)) .GE. 1 ) DUCFAG=ABS(DUCT(1))                   ELE05210
      ENDIF                                                             ELE05220
C                                                                       ELE05230
      IF (ELSTIC) EESE=(FT*P)*(RT*R)*0.50                               ELE05240
      IF (PRINT.AND.FIRST) WRITE(108,1191)                              ELE05250
      IF (PRINT) WRITE(108,1192) IELNO,(DUCT(I),EXCR(I),I=1,3)          ELE05260
 1191 FORMAT (/'  SPRING DUCTILITIES AND EXCURSION RATIOS....',5X/      ELE05270
     & '   ELEMENT',T12,'!- DUCTILITY DEFINATION #1 -! ',              ELE05280
     &                  '!- DUCTILITY DEFINATION #2 -! ',              ELE05290
     &                  '!- DUCTILITY DEFINATION #3 -! ',// T12,       ELE05300
     & '   DUCTILITY     EXCURSION     ',                              ELE05310
     & '   DUCTILITY     EXCURSION     ',                              ELE05320
     & '   DUCTILITY     EXCURSION     '}                              ELE05330
 1192 FORMAT (I10,1P,6G15.6)                                            ELE05340
C                                                                       ELE05350
      RETURN                                                            ELE05360
C                                                                       ELE05370
C-------------------- MAXIMUM FORCES                                    ELE05380
 1200 CONTINUE                                                          ELE05390
      IF(FIRST) WRITE(108,491) 'MAXIMUM LOAD AND DISPL AT MAXIMUM LOAD' ELE05400
     &     //' NOTE: MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'      ELE05410
      WRITE(108,492) IELNO,0,TYPE(KTYPE),NODEI,NODEJ,FMAX(1),FMAX(2)    ELE05420
      RETURN                                                            ELE05430
C                                                                       ELE05440
C                                                                       ELE05450
C-------------------- RESET INTERNAL FORCES                             ELE05460
 1300 CONTINUE                                                          ELE05470
C                                                                       ELE05480
C----- ZERO MATRICIES                                                   ELE05490
        F =0                                                            ELE05500
        FT=0                                                            ELE05510
        R =0                                                            ELE05520
        RT=0                                                            ELE05530
        V =0                                                            ELE05540
        VT=0                                                            ELE05550
        FMAX(1)=0                                                       ELE05560
        FMAX(2)=0                                                       ELE05570
      DO 1310 I=1,ISE                                                   ELE05580
 1310   SE(I)=0                                                         ELE05590
      WRITE(108,1330) IELNO                                             ELE05600
 1330 FORMAT (' SPRING. .. .... ELEMENT #',I6,                          ELE05610
     & '  INTERNAL FORCES AND HYSTERESIS MODELS ARE RESET TO ZERO')     ELE05620
      CALL MATLIB(2,LSTTYP,FALSE,EESE,EPSE, DUCT,EXCR,                  ELE05630
     &  FT,F,RT,R,0.,0.,LELEM,LMAT,MAT,MATTYP,SE,IELNO,DAMAGE)          ELE05640
      RETURN                                                            ELE05650
C                                                                       ELE05660
C                                                                       ELE05670
C-------------------- DAMAGE INDEX ----------------------              ELE05680
 1400 IF (ELSTIC) RETURN                                                ELE05690
C                                                                       ELE05700
      FMAX1=FMAX(1)                                                     ELE05710
      FMAX2=FMAX(2)                                                     ELE05720
      CALL MATLIB(5,LSTTYP,FALSE,EESE,EPSE, DUCT,EXCR,                  ELE05730
     &   FMAX1,0 ,FMAX2,0.,0.,0.,LELEM,LMAT,MAT,MATTYP,SE,IELNO,DAMAGE) ELE05740
      IF (PRINT.AND.FIRST) WRITE(108,1410)                              ELE05750
 1410 FORMAT (/'   ELEMENT',T12,' DAMAGE INDEX       F MAX      ',     ELE05760
     &          '       D MAX          ESE          ,'      PSE       ') ELE05770
      IF (PRINT) WRITE(108,1420) IELNO,DAMAGE,FMAX1,FMAX2,EESE,EPSE     ELE05780
 1420 FORMAT (I10,1P,6G15.6)                                            ELE05790
C                                                                       ELE05800
      DAMAGE=DAMAGE*(EESE*EPSE)                                         ELE05810
C                                                                       ELE05820
      RETURN                                                            ELE05830
      END                                                               ELE05840
C                                                                       ELE05850
C*********************************************************************ELE10020
C                                                                       ELE10030
      SUBROUTINE ELE12                                                  ELE10040
     &  (IOPT       ,FIRST      ,PRINT      ,IREL       ,NAME       ,  ELE10050
     &   ID         ,IDOF       ,COORD      ,COSINE     ,JTCOS      ,  ELE10060
     &   CONST      ,DISPL      ,RINPUT     ,PUB        ,EESE       ,  ELE10070
     &   EPSE       ,DAMAGE     ,IELTYP     ,IELDOF     ,R          ,  ELE10080
     &   RT         ,F          ,FT         ,LM         ,CTS        ,  ELE10090
     &   CTE        ,BS         ,BE         ,NODEI      ,NODEJ      ,  ELE10100
     &   JOINTI     ,JOINTJ     ,REL        ,LENGTH     ,IELNO      ,  ELE10110
     &   ISE        ,ENGDAT     ,FMAX       ,S          ,A          ,  ELE10120
     &   MAT        ,SE         ,STIFF      )                          ELE10130
C                                                                       ELE10140
      IMPLICIT REAL(A-H,O-Z)                                            ELE10150
C                                                                       ELE10160
      COMMON /RSA/ICOMN,RSAP(12)                                        ELE10170
      REAL ICOMN,LENGTH                                                 ELE10180
      INTEGER REL                                                       ELE10190
      LOGICAL FIRST,PRINT,FALSE,BTEST                                   ELE10200
      CHARACTER*8G NAME                                                 ELE10210
$INCLUDE 'ECOMN3'                                                       ELE10220
      DIMENSION  IDOF(MAXNOD,6),COORD(MAXNOD,6),COSINE(3,3,NCOS)       ELE10230
      DIMENSION CONST(MAXNOD,6),   ID(MAXNOD)  ,JTCOS(MAXNOD)          ELE10240
      DIMENSION ASAT(12,12),SAT(12,12),STIFF(78)                        ELE10250
      DIMENSION A(12,12),  S(12,12),LM(12),LMT(12),FMAX(144)            ELE10260
      DIMENSION CTS( 3, 3),CTE( 3, 3),VX(3),VY(3)                       ELE10270
      DIMENSION BS( 3, 3),BE( 3, 3),SE(ISE),ENGDAT(2)                   ELE10280
      DIMENSION R(12),DISPL(NDOF,NLOAD),F(12),RT(12),FT(12),FUB(NDOF)   ELE10290
      DIMENSION RINPUT(100),DUCT(3),EXCR(6),P(12),PL(12),D(12),DL(12)   ELE10300
C                                                                       ELE10310
C-----------------------------------------------------------------------ELE10320
C VARIABLES:                                                            ELE10330
C-----------------------------------------------------------------------ELE10340
C--------- GLOBAL VARIABLES                                            ELE10350
C IOPT   - 1, INITIALIZE BEAM COLUMN ELEMENT                            ELE10360
C        - 2, CALCULATE GEOMETRIC STIFFNESS                             ELE10370
C        - 3, FORM STIFFNESS                                            ELE10380
C        - 4, CALCULATE INCREMENTAL FORCES                              ELE10390
C        - 5, CALCULATE TOTAL FORCES AND ENERGIES                       ELE10400
C IUNIT  - OUTPUT FILE UNIT # FOR PRINTING OUTPUT                       ELE10410
C FIRST  - FLAG, FIRST= TRUE , PRINT HEADERS                            ELE10420
C PRINT  - FLAG, PRINT=.TRUE., PRINT DATA                               ELE10430
C NDOF   - # OF GLOBAL DOF                                              ELE10440
C NLOAD  - # OF LOAD COMBINATIONS                                       ELE10450
C MAXNOD - # ROW DIMENSION OF NODE ARRAY                                ELE10460
C NNODE  - # OF NODES                                                   ELE10470
C ID     - ARRAY OF EXTERNAL NODE NUMBERS                               ELE10480
C IDOF   - ARRAY OF DEGREES OF FREEDOM                                  ELE10490
C COORD  - ARRAY OF COORDINATES                                         ELE10500
C COSINE - ARRAY OF DIRECTION COSINES OF NODES                          ELE10510
C CONST  - ARRAY OF CONSTRAINT TRANSFORMATIONS                          ELE10520
C DISPL  - GLOBAL DISPLACEMENT MATRIX                                   ELE10530
C IZMAT  - LOCATION OF MATERIAL DATA IN GLOBAL STORAGE VECTOR           ELE10540
C NAME   - ELEMENT NAME                                                 ELE10550
C IELNO  - ELEMENT NUMBER                                               ELE10560
C RINPUT - INPUT DATA                                                   ELE10570
C STIFF  - OUTPUT ELEMENT STIFFNESS (UPPER TRIANGULAR FORM)             ELE10580
C LM     - OUTPUT LOCAL TO GLOBAL STIFFNESS MAPPING MATRIX              ELE10590
C---- ----- ELEMENT VARIABLES                                          ELE10600
C H,Q,AI,AO,C,BI,BJ,S22,S26,S212,S33,S35,S311,D                         ELE10610
C        - INDIVIDUAL TERMS IN THE ELEMENT STIFFNESS (S) MATRIX         ELE10620
C R      - INCRIMENTAL DISPLACEMENTS                                    ELE10630
C RT     - TOTAL DISPLACEMENTS                                          ELE10640
C P      - INCRIMENTAL FORCES                                           ELE10650
C FT     - TOTAL FORCES                                                 ELE10660
C LM     - LOCAL TO GLOBAL STIFFNESS MAPPING MATRIX                     ELE10670
C CTS    - LOCAL TO GLOBAL ROTATION MATRIX START                        ELE10680
C CTE    - LOCAL TO GLOBAL ROTATION MATRIX END                          ELE10690
C BS     - LOCAL TO GLOBAL ROTATED ECCENTRICITY/CONSTRAINT  MATRIX STARTELE10700
C BE     - LOCAL TO GLOBAL ROTATED ECCENTRICITY/CONSTRAINT  MATRIX END  ELE10710
C STIFF  - ELEMENT STIFFNESS                                            ELE10720
C IELTYP - ELEMENT TYPE = 12                                            ELE10730
C NODEI  - EXTERNAL JOINT NUMBER, END I                                 ELE10740
C NODEJ  - EXTERNAL JOINT NUMBER, END J                                 ELE10750
C JOINTI - INTERNAL JOINT NUMBER, END I                                 ELE10760
C JOINTJ - INTERNAL JOINT NUMBER, END J                                 ELE10770
C        --------- TEMPROARY VARIABLES                                  ELE10780
C VX     - VECTOR DEFINING THE ELEMENT X AXIS                           ELE10790
C VY     - VECTOR DEFINING THE ELEMENT Y AXIS                           ELE10800
C A      - GLOBAL TO LOCAL FORCE TRANSFORMATION MATRIX                  ELE10810
C S      - ELEMENT STIFFNESS AT LOCAL COORD, AT ENDS OF FLEXABLE PART   ELE10820
C SAT    - PRODUCT OF S*A'                                              ELE10830
C ASAT   - PRODUCT OF A*SAT                                             ELE10840
C-----------------------------------------------------------------------ELE10850
C                                                                       ELE10860
C------- CHOOSE OPTION                                                  ELE10870
       IF (IOPT.LT.1  OR.IOPT.GT.14) THEN                               ELE10880
        WRITE(108,*) 'INVALID OPTION IN ELE-09, IOPT=',IOPT             ELE10890
        STOP                                                            ELE10900
       ENDIF                                                            ELE10910
C                                                                       ELE10920
      GO TO (100,200,300,400,500,600,700,800,900,1000,                  ELE10930
     &                              1100,1200,1300,1400) IOPT           ELE10940
C                                                                       ELE10950
  100 CONTINUE                                                          ELE10960
C                                                                       ELE10970
C----- ZERO STRAIN ENERGY                                               ELE10980
      ENGDAT(1) = 0                                                     ELE10990
      ENGDAT(2) = 0                                                     ELE11000
C                                                                       ELE11010
C----- INTERPERTATE INPUT DATA                                          ELE11020
      IELTYP= 12                                                        ELE11030
      MAT   = RINPUT( 1)                                                ELE11040
      NODEI  = RINPUT( 2)                                               ELE11050
      NODEJ  = RINPUT( 3)                                               ELE11060
      VY(1)  = RINPUT( 4)                                               ELE11070
      VY(2)  = RINPUT( 5)                                               ELE11080
      VY(3)  = RINPUT( 6)                                               ELE11090
      XS     = RINPUT( 7)                                               ELE11100
      XE     = RINPUT( 8)                                               ELE11110
C                                                                       ELE11120
C----- GET INTERNAL NODE # ASSOCIATED WITH EXTERNAL NODE #'S            ELE11130
      JOINTI=IQUICK(NODEI,ID,NNODE)                                     ELE11140
      JOINTJ=IQUICK(NODEJ,ID,NNODE)                                     ELE11150
C                                                                       ELE11160
C----- PRINT DATA FOR PLOTTING                                          ELE11170
      IF (BTEST(IBUG,2)) THEN                                           ELE11180
       WRITE(7,10) NODEJ, NODEI,(COORD(JOINTJ,I),I=1,3),                ELE11190
     &             (COORD(JOINTI,J),J=1,3)                             ELE11200
   10  FORMAT(I5,1X,I5,1P,6G12.4)                                       ELE11210
      ENDIF                                                             ELE11220
C                                                                       ELE11230
C-------------------------------------------------------- --- ----     ELE11240
C  GET GLOBAL TO LOCAL TRANSFORMATION MATRIX                            ELE11250
C-------------------------------------------------- ------ ------- ---- ELE11260
C----  DEFINE VECTOR X, LENGTH                                          ELE11270
      VX(1)=COORD(JOINTJ,1)-COORD(JOINTI,1)                             ELE11280
      VX(2)=COORD(JOINTJ,2)-COORD(JOINTI,2)                             ELE11290
      VX(3)=COORD(JOINTJ,3)-COORD(JOINTI,3)                             ELE11300
      LENGTH=SQRT(VX(1)**2+VX(2)**2+VX(3)**2)-XS-XE                     ELE11310
      SE(79)=LENGTH                                                     ELE11320
C                                                                       ELE11330
      IF (LENGTH.LE 0) THEN                                             ELE11340
         WRITE(108,101) IELNO,JOINTI,JOINTJ,LENGTH                      ELE11350
  101    FORMAT(1X,20('*'),'ER','RCR - LENGTH IS LE 0',/                ELE11360
     &    5X,'IELNO=',I5,5X,'JOINTI=',I5,5X,'JOINTJ=',I5,               ELE11370
     &    5X,'LENGTH',1P,G15.6/                                         ELE11380
     &    5X,'REVISE INPUT.... ..... ... ..............'/)             ELE11390
         LENGTH=1.0                                                     ELE11400
      ENDIF                                                             ELE11410
C----- GET LOCAL TO GLOBAL TRANSFORMATION MATRICIES CT AND B            ELE11420
      ICS=JTCOS(JOINTI)                                                 ELE11430
      ICE=JTCOS(JOINTJ)                                                 ELE11440
C                                                                       ELE11450
      CALL ROTXYZ(VX,VY,COSINE(1,1,ICS),CTS, XS ,0.,0.,BS,              ELE11460
     &            CONST(JOINTI,1),MAXNOD)                               ELE11470
      CALL ROTXYZ(VX,VY,COSINE(1,1,ICE),CTE,-XE ,0.,0.,BE,              ELE11480
     &            CONST(JOINTJ,1),MAXNOD)                               ELE11490
C                                                                       ELE11500
C----- ZERO MATRICIES                                                   ELE11510
      DO 40 I=1,12                                                      ELE11520
        FMAX(I)=0                                                       ELE11530
        F (I)=0                                                         ELE11540
        FT(I)=0                                                         ELE11550
        R (I)=0                                                         ELE11560
        RT(I)=0                                                         ELE11570
        DO 40 J=1,12                                                    ELE11580
          S(I,J)=0                                                      ELE11590
   40     A(I,J)=0                                                      ELE11600
C                                                                       ELE11610
C----- ASSEMBLE TRANSFORMATION MATRIX A                                 ELE11620
      DO 50 I=1,3                                                       ELE11630
      DO 50 J=1,3                                                       ELE11640
        A(I  ,J  )=CTS(I,J)                                             ELE11650
        A(I+3,J+3)=CTS(I,J)                                             ELE11660
        A(I+6,J+6)=CTE(I,J)                                             ELE11670
        A(I+9,J+9)=CTE(I,J)                                             ELE11680
        A(I+3,J  )=BS(I,J)                                              ELE11690
   50   A(I+9,J+6)=BE(I,J)                                             ELE11700
C                                                                       ELE11710
C----- GET MATERIAL PROPERTIES                                          ELE11720
       FALSE= FALSE.                                                    ELE11730
       LSTTYP=0                                                         ELE11740
       CALL MATLIB(2,LSTTYP,FALSE,EESE,EPSE,DUCT,EXCR,FT,F,RT,R,        ELE11750
     &             0.,0.,LELEM,LMAT,MAT,MATTYP,SE,IELNO,DAMAGE)         ELE11760
       IF( MATTYP .NE. 12) THEN                                         ELE11770
         WRITE(108,190) IELNO,MATTYP,NODEI,NODEJ                        ELE11780
  190    FORMAT (' ELEMENT ',I6,' INCOMPATIBLE MATERIAL PROP.#',        ELE11790
     &    I5,' START JOINT ',I5,' END JOINT',I5,/                       ELE11800
     &     ' THE ELEMENT IS REJECTED AND ERROR CHECKING CONTINUES')     ELE11810
         GO TO 191                                                      ELE11820
       ENDIF                                                            ELE11830
C                                                                       ELE11840
      EESE=0                                                            ELE11850
      EPSE=0                                                            ELE11860
C                                                                       ELE11870
C----- --- NO END RELEASE FOR THIS ELEMENT                             ELE11880
      REL=0                                                             ELE11890
C                                                                       ELE11900
C-------- SET STIFFNESS COIEFFICIENTS...                                ELE11910
C                                                                       ELE11920
C------------------------------------------ ---------- ------ ---- ------ ELE11930
C  ASSEMBLE LOCAL STIFFNESS MATRIX                                      ELE11940
C-------------------------------------------------- ---------- ------- ELE11950
      K=0                                                               ELE11960
      DO 77 I=1,12                                                      ELE11970
      DO 77 J=1,12                                                      ELE11980
        K=K+1                                                           ELE11990
        IF( ABS(SE(K)) .LT  1.E-6) SE(K)=0                              ELE12000
        S(I,J)=SE(K)                                                    ELE12010
        S(J,I)=S(I,J)                                                   ELE12020
   77 CONTINUE                                                          ELE12030
C                                                                       ELE12040
C----- CALCULATE SAT                                                    ELE12050
      CALL MULTM(3,12,SAT,S,A)                                          ELE12060
C                                                                       ELE12070
C----- CALCULATE ASAT                                                   ELE12080
      CALL MULTM(0,12,ASAT,A,SAT)                                       ELE12090
C                                                                       ELE12100
C----- DEGREES OF FREEDOM                                               ELE12110
C      REMOVE DOF'S THAT ARE CONSTRAINED TO THE SAME DOF...             ELE12120
      DO 75 I=1,6                                                       ELE12130
        IF ( IDOF(JOINTI,I).NE.IDOF(JOINTJ,I) ) THEN                    ELE12140
           LMT(I  )=IDOF(JOINTI,I)                                      ELE12150
           LMT(I+6)=IDOF(JOINTJ,I)                                      ELE12160
        ELSE                                                            ELE12170
           LMT(I  )= 0                                                  ELE12180
           LMT(I+6)= 0                                                  ELE12190
        ENDIF                                                           ELE12200
   75   CONTINUE                                                        ELE12210
C                                                                       ELE12220
C--- - CONVERT TO HALF STORAGE MODE  BY COLUMNS                         ELE12230
C                                                                       ELE12240
      L=0                                                               ELE12250
      IELDOF=0                                                          ELE12260
      DO 90 I=1,12                                                      ELE12270
        IF (LMT(I).EQ 0) GO TO 90                                       ELE12280
```

```
      IELDOF=IELDOF+1                                                   ELE12290
      LM(IELDOF)=LMT(I)                                                 ELE12300
      DO 80 J=1,1,-1                                                    ELE12310
        IF (LMT(J).EQ.0) GO TO 80                                       ELE12320
        L=L+1                                                           ELE12330
        STIFF(L)=ASAT(I,J)                                              ELE12340
   80   CONTINUE                                                        ELE12350
   90   CONTINUE                                                        ELE12360
C                                                                       ELE12370
C----- RELEASE UNUSED STORAGE                                           ELE12380
  180 IREL= 78-L                                                        ELE12390
C                                                                       ELE12400
C     PRINT COLUMN DATA                                                 ELE12410
      IF (FIRST) WRITE(108,192)                                         ELE12420
  191 WRITE(108,193) NAME,IELNO,MAT,NODEI,NODEJ,LENGTH,                 ELE12430
     &                VY(1),VY(2),VY(3),XS,-XE                         ELE12440
C                                                                       ELE12450
  192 FORMAT(/' ELEMENT 12,  STABILITY ELEMENT'//                       ELE12460
     &  T22,'#   MATL   START    END',3X,'LENGTH',3X,                  ELE12470
     &  11('-'),'  Y-AXIS  ',                                           ELE12480
     &  12('-'),'  START DIST     END DIST    '/)                       ELE12490
  193 FORMAT(1X,A15,316,1X,I6,2X,1P,G10.4,0P,                           ELE12500
     &       F9.5,' I',SP,F9.5,' J',F9.5,' K',1P,SS,2G12.4)             ELE12510
      RETURN                                                            ELE12520
C------------------------ DETERMINE GEOMETRIC STIFF ---------------------ELE12530
  200 CONTINUE                                                          ELE12540
C----GEOMETRIC STIFFNESS IS INCLUDED IN THE ELEMENT STIFFNESS.          ELE12550
C    NO NEED TO CALCULATED                                              ELE12560
C                                                                       ELE12570
      RETURN                                                            ELE12580
C------------------------ DETERMINE STIFFNESS ---------------------------ELE12590
  300 CONTINUE                                                          ELE12600
C                                                                       ELE12610
C---- STIFFNESS CHANGES AT EACH STEP FOR STABILITY ELEMENT              ELE12620
C                                                                       ELE12630
C---------- ---------------------------------------------------------- ELE12640
C  ASSEMBLE LOCAL STIFFNESS MATRIX                                      ELE12650
C---------------------------------------------------- ---------------- ELE12660
      K=0                                                               ELE12670
      DO 87 I=1,12                                                      ELE12680
      DO 87 J=1,12                                                      ELE12690
         K=K+1                                                          ELE12700
         IF( ABS(SE(K)) .LT. 1.E-6) SE(K)=0                             ELE12710
         S(I,J)=SE(K)                                                   ELE12720
         S(J,I)=S(I,J)                                                  ELE12730
   87 CONTINUE                                                          ELE12740
C                                                                       ELE12750
C----- CALCULATE SAT                                                    ELE12760
      CALL MULTH(3,12,SAT,S,A)                                          ELE12770
C                                                                       ELE12780
C ---- CALCULATE ASAT                                                   ELE12790
      CALL MULTH(0,12,ASAT,A,SAT)                                       ELE12800
C                                                                       ELE12810
C----- DEGREES OF FREEDOM                                               ELE12820
C     REMOVE DOF'S THAT ARE CONSTRAINED TO THE SAME DOF...              ELE12830
      DO 12 I=1,6                                                       ELE12840
        IF ( IDOF(JOINTI,I).NE.IDOF(JOINTJ,I) ) THEN                    ELE12850
           LMT(I  )=IDOF(JOINTI,I)                                      ELE12860
           LMT(I+6)=IDOF(JOINTJ,I)                                      ELE12870
        ELSE                                                            ELE12880
           LMT(I  )= 0                                                  ELE12890
           LMT(I+6)= 0                                                  ELE12900
        ENDIF                                                           ELE12910
   12   CONTINUE                                                        ELE12920
C                                                                       ELE12930
C----- CONVERT TO HALF STORAGE MODE  BY COLUMNS                         ELE12940
C                                                                       ELE12950
      L=0                                                               ELE12960
      IELDOF=0                                                          ELE12970
      DO 92 I=1,12                                                      ELE12980
        IF (LMT(I).EQ.0) GO TO 92                                       ELE12990
        IELDOF=IELDOF+1                                                 ELE13000
        LM(IELDOF)=LMT(I)                                               ELE13010
        DO 82 J=I,1,-1                                                  ELE13020
          IF (LMT(J).EQ.0) GO TO 82                                     ELE13030
          L=L+1                                                         ELE13040
          STIFF(L)=ASAT(I,J)                                            ELE13050
   82   CONTINUE                                                        ELE13060
   92   CONTINUE                                                        ELE13070
C                                                                       ELE13080
      RETURN                                                            ELE13090
C------------------------ DETERMINE INCRIMENTAL FORCES ------------------ELE13100
  400 CONTINUE                                                          ELE13110
      IF (PRINT.AND.FIRST) WRITE(108,491) STEPID                        ELE13120
C                                                                       ELE13130
C                                                                       ELE13140
C------ LOOP FOR EACH LOAD                                             ELE13150
      DO 430 LOAD=1,NLOAD                                               ELE13160
C                                                                       ELE13170
C------ TRANSFORM FRAME DISPL INTO LOCAL DISPL                          ELE13180
       DO 410 I=1,3                                                     ELE13190
         R(I  )=0.                                                      ELE13200
         R(I+3)=0.                                                      ELE13210
         R(I+6)=0.                                                      ELE13220
         R(I+9)=0                                                       ELE13230
         DO 410 J=1,3                                                   ELE13240
            D1=DISPL(IDOF(JOINTI,J  ),LOAD)                             ELE13250
            D2=DISPL(IDOF(JOINTI,J+3),LOAD)                             ELE13260
            D3=DISPL(IDOF(JOINTJ,J  ),LOAD)                             ELE13270
            D4=DISPL(IDOF(JOINTJ,J+3),LOAD)                             ELE13280
         R(I  )=R(I  )+CTS(J,I)*D1 + BS(J,I)*D2                         ELE13290
         R(I+3)=R(I+3)+CTS(J,I)*D2                                      ELE13300
         R(I+6)=R(I+6)+CTE(J,I)*D3 + BE(J,I)*D4                         ELE13310
  410    R(I+9)=R(I+9)+CTE(J,I)*D4                                      ELE13320
C                                                                       ELE13330
C------ INCRIMENTAL FORCES                                             ELE13340
      DO 435 I=1,12                                                     ELE13350
        F(I)=0                                                          ELE13360
        DO 435 K=1,12                                                   ELE13370
  435     F(I)=F(I)+S(I,K)*R(K)                                         ELE13380
C                                                                       ELE13390
C                                                                       ELE13400
C------ DETERMINE MAXIMUN VALUES FOR MULTIPLE LOAD CASES                ELE13410
      IF (NLOAD.GT.1) THEN                                              ELE13420
        DO 425 I=1,12                                                   ELE13430
          IF ( ABS(F(I)).GT.ABS(FMAX(I+12*12+I)) ) THEN                 ELE13440
            DO 426 J=0,11                                               ELE13450
  426         FMAX(I+12*11+J)=F(J+1)                                    ELE13460
          ENDIF                                                         ELE13470
  425   CONTINUE                                                        ELE13480
      ENDIF                                                             ELE13490
C                                                                       ELE13500
C------ PRINT FORCES AND DEFORMATIONS                                   ELE13510
      SP=SE(84)                                                         ELE13520
      FLP=SE(98)                                                        ELE13530
      IF (PRINT) WRITE(108,493) IELNO,LOAD,NODEI,(R(J),J=1,6),          ELE13540
     &                          NODEJ,(R(J),J=7,12),FLP                 ELE13550
      IF (PRINT) WRITE(108,492) IELNO,LOAD,NODEI,(F(J),J=1,6),          ELE13560
     &                          NODEJ,(F(J),J=7,12),SP                  ELE13570
C                                                                       ELE13580
  430 CONTINUE                                                          ELE13590
C                                                                       ELE13600
C ------THIS IS FOR SOL05A USE ONLY                                     ELE13610
      IF( ICOMN .EQ. 1 .OR. ICOMN .EQ. 2 ) THEN                         ELE13620
      DO 431 I=1,12                                                     ELE13630
  431   RSAF(I)=F(I)                                                    ELE13640
      ENDIF                                                             ELE13650
C                                                                       ELE13660
C                                                                       ELE13670
C  --- CALLL MATLIB TO GET STIFFNESS                                    ELE13680
C       THIS IS PERFORMED ONLY FOR INELASTIC ANALYSIS                   ELE13690
      IF(.NOT. ELSTIC) THEN                                             ELE13700
           DO 93 I=1,12                                                 ELE13710
             P(I)=FT(I)+P(I)                                            ELE13720
             D(I)=RT(I)+R(I)                                            ELE13730
             PL(I)=FT(I)                                                ELE13740
             DL(I)=RT(I)                                                ELE13750
   93      CONTINUE                                                     ELE13760
C                                                                       ELE13770
C    ----- TRANSFER PERMENANT HYSTERESIS DATA TO TEMPERARY STORAGE      ELE13780
         ISE2=ISE/2                                                     ELE13790
         DO 522 I=1,ISE2                                                ELE13800
  522      SE(I+ISE2)=SE(I)                                             ELE13810
C                                                                       ELE13820
         CALL MATLIB(3,LSTTYP,FALSE,EESE,EPSE,DUCT,EXCR,P,PL,D,DL,      ELE13830
     &              0.,0.,LELEM,LMAT,MAT,MATTYP,SE(ISE2+1),IELNO,DAMAGE) ELE13840
         NEWX=.TRUE.                                                    ELE13850
      ENDIF                                                             ELE13860
C                                                                       ELE13870
C                                                                       ELE13880
C------ CALCULATE THE UNBALACED MEMBER FORCES ON THE JOINT              ELE13890
C       THE EXACT MEMBER FORCES ARE DETERMINED FROM HYSTERESIS MODEL    ELE13900
C       BASED ON DISPLACEMENT D. BECAUSE NO HYSTERESIS MODEL FOR MAT12, ELE13910
C       THE MEMBER FUE ARE NOT CALCULATED IN THIS ELEMENT               ELE13920
      DO 440 I=1,3                                                      ELE13930
        I1=IDOF(JOINTI,I  )                                             ELE13940
        I2=IDOF(JOINTI,I+3)                                             ELE13950
        J1=IDOF(JOINTJ,I  )                                             ELE13960
        J2=IDOF(JOINTJ,I+3)                                             ELE13970
        DO 440 J=1,3                                                    ELE13980
          FUB(I1)=FUB(I1)+CTS(I,J)*D                                    ELE13990
          FUB(I2)=FUB(I2)+ BS(I,J)*D                                    ELE14000
          FUB(I2)=FUB(I2)+CTS(I,J)*D                                    ELE14010
          FUB(J1)=FUB(J1)+CTE(I,J)*D                                    ELE14020
          FUB(J2)=FUB(J2)+ BE(I,J)*D                                    ELE14030
  440     FUB(J2)=FUB(J2)+CTE(I,J)*D                                    ELE14040
C                                                                       ELE14050
  491 FORMAT (/'     STABILITY ELEMENT  FORCES...',5X/7X,A/,            ELE14060
     &  ' ELEMENT LOAD       NODE',7X,'AXIAL',11X,'FY',13X,'FZ',        ELE14070
     & 11X,'TORSION',10X,'MY',13X,'MZ      FLP,SP ')                   ELE14080
  492 FORMAT (/,I10,I5,2X,'FORCE',I6,1P,6G15.6,                         ELE14090
     &       /15X   ,2X,'FORCE',I6,   6G15.6,1X,G10.3)                  ELE14100
  493 FORMAT (/,I10,I5,2X,'DISPL',I6,1P,6G15.6,                         ELE14110
     &       /15X   ,2X,'DISPL',I6,   6G15.6,1X,G10.3)                  ELE14120
  494 FORMAT (7X,A)                                                     ELE14130
      RETURN                                                            ELE14140
C------------------------ DETERMINE TOTAL FORCES ------------------------ELE14150
  500 CONTINUE                                                          ELE14160
      IF (PRINT.AND.FIRST) WRITE(108,491) STEPID                        ELE14170
      IF( ICOMN .NE. 1 .AND. ICOMN .NE. 2 ) THEN                        ELE14180
C                                                                       ELE14190
C ----- TOTAL FORCES AND DISPLACEMENTS                                  ELE14200
      DO 510 I=1,12                                                     ELE14210
           FT(I)=FT(I)+P(I)                                             ELE14220
           RT(I)=RT(I)+R(I)                                             ELE14230
  510 CONTINUE                                                          ELE14240
C                                                                       ELE14250
      DO 512 I=1,12                                                     ELE14260
           IF ( ABS(FT(I)) GT.ABS(FMAX(I+12*12+I)) ) THEN               ELE14270
              DO 511 J=0,11                                             ELE14280
  511           FMAX(I+12*11+J)=FT(J+1)                                 ELE14290
           ENDIF                                                        ELE14300
  512 CONTINUE                                                          ELE14310
  520 FORMAT ('   I:',I3,' R:',1P,G12.4,' RT ',G12.4,                   ELE14320
     &                   ' P:',1P,G12.4,' FT:',G12.4)                 ELE14330
      ELSE IF( ICOMN .EQ. 1 .OR. ICOMN .EQ. 2 ) THEN                    ELE14340
        DO 513 I=1,12                                                   ELE14350
  513     FT(I)=RSAF(I)                                                 ELE14360
      ENDIF                                                             ELE14370
C                                                                       ELE14380
      LOAD=1                                                            ELE14390
      ISE2=ISE/2                                                        ELE14400
      SP=SE(ISE2+84)                                                    ELE14410
      FLP=SE(ISE2+98)                                                   ELE14420
      IF (PRINT) WRITE(108,493) IELNO,LOAD,NODEI,(RT(J),J=1,6),         ELE14430
     &                          NODEJ,(RT(J),J=7,12),FLP                ELE14440
      IF (PRINT) WRITE(108,492) IELNO,LOAD,NODEI,(FT(J),J=1,6),         ELE14450
     &                          NODEJ,(FT(J),J=7,12),SP                 ELE14460
C                                                                       ELE14470
C     ---- TRANSFER TEMPERARY HYSTERESIS DATA TO PERMANENT STORAGE      ELE14480
         ISE2=ISE/2                                                     ELE14490
         IF( .NOT. ELSTIC) THEN                                         ELE14500
         DO 415 I=1,ISE2                                                ELE14510
  415      SE(I)=SE(I+ISE2)                                             ELE14520
         ENDIF                                                          ELE14530
C                                                                       ELE14540
      RETURN                                                            ELE14550
C                                                                       ELE14560
C------------------------ DETERMINE FIXED END FORCES, AND ADD TO FORCE--ELE14570
  600 CONTINUE                                                          ELE14580
C-----FIXED END FORCES ARE NOT AVAILABLE FOR STABILITY ELEMENT          ELE14590
C                                                                       ELE14600
      RETURN                                                            ELE14610
C                                                                       ELE14620
C-------------------- WRITE MEMBER FORCES TO OUTPUT FILE ---------------ELE14630
  700 CONTINUE                                                          ELE14640
C                                                                       ELE14650
      SP=SE(84)                                                         ELE14660
      FLP=SE(98)                                                        ELE14670
      WRITE (IUNIT,710) NODEI,(FT(J),J=1,6),NODEJ,(FT(J),J=7,12),SP     ELE14680
      WRITE (IUNIT,710) NODEI,(RT(J),J=1,6),NODEJ,(RT(J),J=7,12),FLP    ELE14690
  710 FORMAT (16,1P,6G12.5/,16,1P,6G12.5,1X,G10.3)                      ELE14700
C                                                                       ELE14710
      RETURN                                                            ELE14720
C                                                                       ELE14730
C------------------- READ AND PRINT MEMBER FORCES FROM OUTPUT FILE ----ELE14740
  800 CONTINUE                                                          ELE14750
      BACKSPACE(IUNIT)                                                  ELE14760
      READ (IUNIT,*  ) ITYPE,IELNO,IWRITE,TO,DT                        ELE14770
      WRITE(108,805)                                                    ELE14780
      ISTEP=-IWRITE                                                     ELE14790
  810 READ(IUNIT,815,END=830)                                           ELE14800
     &              NODEI,(FT(J),J=1,6),NODEJ,(FT(J),J=7,12),SP        ELE14810
      READ (IUNIT,815,END=830)                                          ELE14820
     &              NODEI,(RT(J),J=1,6),NODEJ,(RT(J),J=7,12),FLP       ELE14830
  815 FORMAT (16,6G12.5/,16,6G12.5,1X,G10.3)                            ELE14840
      ISTEP=ISTEP+IWRITE                                                ELE14850
      T=TO+DT*ISTEP                                                     ELE14860
      IF (MOD(ISTEP,INC).EQ.0) THEN                                     ELE14870
        WRITE(108,820) ISTEP,T,NODEI,(FT(J),J=1,6),                     ELE14880
     &                 NODEJ,(FT(J),J=7,12),SP                         ELE14890
        WRITE(108,920) (RT(J),J=1,12),FLP                               ELE14900
      ENDIF                                                             ELE14910
      GO TO 810                                                         ELE14920
C                                                                       ELE14930
  805 FORMAT (/'     STABILITY ELEMENT FORCES...',/                     ELE14940
     &  '       STEP    TIME          NODE',7X,'AXIAL',11X,'FY',13X,'FZ',ELE14950
     & 11X,'TORSION',10X,'MY',13X,'MZ      FLP,SP')                    ELE14960
  820 FORMAT (/,I10,1P,G15.5,0P,I6,1P,6G15.6,/25X,I6,6G15.6,1X,G10.3)    ELE14970
  920 FORMAT ( 25X,'DISP  ',1P,6G15.6,/31X,6G15.6,1X,G10.3)             ELE14980
C                                                                       ELE14990
  830 RETURN                                                            ELE15000
C                                                                       ELE15010
C------------------- DETERMINE THE LENGTH OF THE MATERIAL DATA ---------ELE15020
  900 CONTINUE                                                          ELE15030
      ISE=0                                                             ELE15040
        MAT  = RINPUT( 1)                                               ELE15050
        CALL MATLIB(0,LSTTYP,FALSE,EESE,EPSE,DUCT,EXCR,PT,F,RT,R,       ELE15060
     &              0.,0.,LELEM,LMAT,MAT,MATTYP,SE,IELNO,DAMAGE)        ELE15070
        ISE=ISE+LELEM                                                   ELE15080
C                                                                       ELE15090
C----- DOUBLE THE STORAGE, TO ALLOW FOR TEMPOARY VALUES                 ELE15100
       ISE=ISE*2                                                        ELE15110
C                                                                       ELE15120
      RETURN                                                            ELE15130
C                                                                       ELE15140
C------------------- DETERMINE THE TOTAL STRAIN ENERGY ----------------ELE15150
 1000 CONTINUE                                                          ELE15160
      EESE=0                                                            ELE15170
      EPSE=0                                                            ELE15180
C                                                                       ELE15190
C     /* ENERGY FORMULATION FOR ELE12 ELEMENT IS NOT AVAILABLE */       ELE15200
```

```
      DO 1010 I=1,12                                                    ELE15210
 1010   EESE=EESE + (FT(I)+P(I))*(R(I)+RT(I))/2.                        ELE15220
      ENGDAT(1) = EESE                                                  ELE15230
      ENGDAT(2) = 0                                                     ELE15240
C                                                                       ELE15250
      RETURN                                                            ELE15260
C                                                                       ELE15270
C-------------------- DUCTILITIES AND EXCURSION RATIOS                  ELE15280
C                     NOT AVAILABLE CURRENTLY                           ELE15290
 1100 RETURN                                                            ELE15300
C                                                                       ELE15310
C-------------------- PRINT MAXIMUM ELEMENT LOADS                       ELE15320
C                                                                       ELE15330
 1200 CONTINUE                                                          ELE15340
      LOAD2=0                                                           ELE15350
      IF (FIRST) WRITE(108,494)                                         ELE15360
     &          ' COLUMN ( LOAD) REPRESENTS THE DOF WHICH HAS',        ELE15370
     &          ' MAXIMUM VALUE'                                        ELE15380
      IF (FIRST) WRITE(108,491) ' MAXIMUM VALUES FOR ALL STEPS  '       ELE15390
     &          //' NOTE: MAXIMUM VALUES WITH THE OTHER DOFS FORCES'   ELE15400
     &          //' ARE PRINT OUT AT THE SAME TIME'                    ELE15410
      DO 309 I=0,11                                                     ELE15420
           LOAD2=I+1                                                    ELE15430
           IF( PMAX(I*12+1+I) .EQ. 0. ) GO TO 309                       ELE15440
           WRITE(108,492) IELNO,LOAD2,NODEI,(PMAX(I*12+J),J=1,6),       ELE15450
     &                    NODEJ,(PMAX(I*12+J),J=7,12)                   ELE15460
  309 CONTINUE                                                          ELE15470
C                                                                       ELE15480
      RETURN                                                            ELE15490
C                                                                       ELE15500
C-------------------- RESET ELEMENT FORCES                              ELE15510
C                                                                       ELE15520
 1300 CONTINUE                                                          ELE15530
C----- ZERO MATRICIES                                                   ELE15540
      DO 1340 I=1,12                                                    ELE15550
        PMAX(I*12+11)=0                                                 ELE15560
        F (I)=0                                                         ELE15570
        FT(I)=0                                                         ELE15580
        R (I)=0                                                         ELE15590
 1340   RT(I)=0                                                         ELE15600
      DO 1310 I=1,ISE                                                   ELE15610
 1310   SE(I)=0                                                         ELE15620
      SE(79)=LENGTH                                                     ELE15630
C                                                                       ELE15640
      WRITE(108,1330) IELNO                                             ELE15650
 1330 FORMAT (' STABILITY ELEMENT ..... ELEMENT #',I6,                  ELE15660
     & '  INTERNAL FORCES AND HYSTERESIS MODELS ARE RESET TO ZERO')     ELE15670
        FALSE=.FALSE.                                                   ELE15680
        LSTTYP=0                                                        ELE15690
        CALL MATLIB(2,LSTTYP,FALSE,EESE,EPSE,DUCT,EXCR,FT,F,RT,R,       ELE15700
     &              0.,0 ,LELEM,LMAT,MAT,MATTYP,SE,IELNO,DAMAGE)        ELE15710
C                                                                       ELE15720
      RETURN                                                            ELE15730
C                                                                       ELE15740
C-------------------- DAMAGE INDEX                                      ELE15750
C                                                                       ELE15760
 1400 DAMAGE = 0.                                                       ELE15770
C                                                                       ELE15780
      RETURN                                                            ELE15790
      END                                                               ELE15800
C*************************************************************************ELE20010
      SUBROUTINE ELE09                                                  ELE20030
     & (IOPT       ,FIRST      ,PRINT      ,NAME       ,AXIAL      ,    ELE20040
     &  ID         ,IDOF       ,COORD      ,COSINE     ,JTCOS      ,    ELE20050
     &  CONST      ,DISPL      ,FORCE      ,IREL       ,           ,    ELE20060
     &  KGDATA     ,PGEOM      ,IELD       ,ELD        ,IELNO      ,    ELE20070
     &  RINPUT     ,PUB        ,EESE       ,EPSE       ,DAMAGE     ,    ELE20080
     &  DUCFAG     ,IELTYP     ,IELDOF     ,MYZX       ,STY        ,    ELE20090
     &  ISEMLI     ,PKG        ,R          ,RT         ,F          ,    ELE20100
     &  FT         ,LM         ,CTS        ,CTE        ,BS         ,    ELE20110
     &  BE         ,NODEI      ,NODEJ      ,JOINTI     ,JOINTJ     ,    ELE20120
     &  REL        ,LENGTH     ,KGDOF      ,LKG        ,LKKG       ,    ELE20130
     &  ISE        ,ENGDAT     ,PMAX       ,IAPAG      ,STBFAG     ,    ELE20140
     &  MATMLI     ,S23        ,ROIELI     ,A          ,           ,    ELE20150
     &  SE         ,STIFF      ,FEM  )                                 ELE20160
C                                                                       ELE20170
      COMMON /IDSTEP/ JSTEP                                             ELE20180
      COMMON /BSA/ICOMN,RSAF(12)                                        ELE20190
      REAL ICOMN,LENGTH                                                 ELE20200
      INTEGER IAPAG,STBFAG,REL                                          ELE20210
      LOGICAL FIRST,PRINT,FEMFLG,FALSE,AXIAL,BTEST                      ELE20220
      CHARACTER*80 NAME                                                 ELE20230
$INCLUDE:'ZCOMN3'                                                       ELE20240
      DIMENSION KGDATA(3),S23(0:13),IDOF(MAXNOD,6),COORD(MAXNOD,6)      ELE20250
      DIMENSION COSINE(3,3,NCOS),CONST(MAXNOD,6),ID(MAXNOD)             ELE20260
      DIMENSION ASAT(12,12),SAT(12,12),STIFF(156),JTCOS(MAXNOD)         ELE20270
      DIMENSION A(12,12),S(12,12),LM(12),LMT(12),LMKG(12),FMAX(144)     ELE20280
      DIMENSION CTS(3,3),CTE(3,3),VX(3),VY(3)                           ELE20290
      DIMENSION BS(3,3),BE(3,3),SE(ISE),ENGDAT(2),STY(10)               ELE20300
      DIMENSION R(12),DISPL(NDOF,NLOAD),F(12),RT(12),FT(12),FUB(NDOF)   ELE20310
      DIMENSION FORCE(NDOF,NLOAD),IELD(5,MAXELD),ELD(12,MAXELD)         ELE20320
      DIMENSION RINPUT(100),FEM(12,NLOAD),GFEM(12),TMP(6)               ELE20330
      DIMENSION DUCT(3,6),EXCR(6,6),P(6),V(12),VL(12),EFUB(12),UP(12)   ELE20340
      DIMENSION MYZX(6),ISEMLI(6),MATMLI(6),ROIELI(4),RIELI(4)          ELE20350
C                                                                       ELE20360
C*************************************************************************ELE20370
C VARIABLES:                                                            ELE20380
C-------------------------------------------------------------------------ELE20390
C---------- GLOBAL VARIABLES                                           ELE20400
C IOPT   - 1, INITIALIZE BEAM COLUMN ELEMENT                            ELE20410
C        - 2, CALCULATE GEOMETRIC STIFFNESS                             ELE20420
C        - 3, FORM STIFFNESS                                            ELE20430
C        - 4, CALCULATE INCREMENTAL FORCES                              ELE20440
C        - 5, CALCULATE TOTAL FORCES AND ENERGIES                       ELE20450
C IUNIT  - OUTPUT FILE UNIT # FOR PRINTING OUTPUT                       ELE20460
C FIRST  - FLAG, FIRST=.TRUE., PRINT HEADERS                            ELE20470
C PRINT  - FLAG, PRINT=.TRUE., PRINT DATA                               ELE20480
C NDOF   - # OF GLOBAL DOF                                              ELE20490
C NLOAD  - # OF LOAD COMBINATIONS                                       ELE20500
C MAXNOD - # ROW DIMENSION OF NODE ARRAY                                ELE20510
C NNODE  - # OF NODES                                                   ELE20520
C ID     - ARRAY OF EXTERNAL NODE NUMBERS                               ELE20530
C IDOF   - ARRAY OF DEGREES OF FREEDOM                                  ELE20540
C COORD  - ARRAY OF COORDINATES                                         ELE20550
C COSINE - ARRAY OF DIRECTION COSINES OF NODES                          ELE20560
C CONST  - ARRAY OF CONSTRAINT TRANSFORMATIONS                          ELE20570
C DISPL  - GLOBAL DISPLACEMENT MATRIX                                   ELE20580
C IZMAT  - LOCATION OF MATERIAL DATA IN GLOBAL STORAGE VECTOR           ELE20590
C NAME   - ELEMENT NAME                                                 ELE20600
C IELNO  - ELEMENT NUMBER                                               ELE20610
C RINPUT - INPUT DATA                                                   ELE20620
C STIFF  - OUTPUT ELEMENT STIFFNESS (UPPER TRIANGULAR FORM)             ELE20630
C LM     - OUTPUT LOCAL TO GLOBAL STIFFNESS MAPPING MATRIX              ELE20640
C---------- ELEMENT VARIABLES                                           ELE20650
C H,Q,AI,AJ,C,BI,BJ,S22,S26,S212,S33,S35,S311,D                         ELE20660
C        - INDIVIDUAL TERMS IN THE ELEMENT STIFFNESS (S) MATRIX         ELE20670
C PKG    - INPUT LOAD FOR FORMING GEOMETRIC STIFFNESS MATRIX ..         ELE20680
C R      - INCRIMENTAL DISPLACEMENTS                                    ELE20690
C RT     - TOTAL DISPLACEMENTS                                          ELE20700
C F      - INCRIMENTAL FORCES                                           ELE20710
C FT     - TOTAL FORCES                                                 ELE20720
C LM     - LOCAL TO GLOBAL STIFFNESS MAPPING MATRIX                     ELE20730
C CTS    - LOCAL TO GLOBAL ROTATION MATRIX START                        ELE20740
C CTE    - LOCAL TO GLOBAL ROTATION MATRIX END                          ELE20750
C BS     - LOCAL TO GLOBAL ROTATED ECCENTRICITY/CONSTRAINT  MATRIX STARTELE20760
C BE     - LOCAL TO GLOBAL ROTATED ECCENTRICITY/CONSTRAINT  MATRIX END  ELE20770
C STIFF  - ELEMENT STIFFNESS                                            ELE20780
C IELTYP - ELEMENT TYPE = 9                                             ELE20790
C NODEI  - EXTERNAL JOINT NUMBER, END I                                 ELE20800
C NODEJ  - EXTERNAL JOINT NUMBER, END J                                 ELE20810
C JOINT1 - INTERNAL JOINT NUMBER, END I                                 ELE20820
C JOINTJ - INTERNAL JOINT NUMBER, END J                                 ELE20830
C---------- TEMPROARY VARIABLES                                         ELE20840
C VX     - VECTOR DEFINING THE ELEMENT X AXIS                           ELE20850
C VY     - VECTOR DEFINING THE ELEMENT Y AXIS                           ELE20860
C A      - GLOBAL TO LOCAL FORCE TRANSFORMATION MATRIX                  ELE20870
C S      - ELEMENT STIFFNESS AT LOCAL COORD, AT ENDS OF FLEXABLE PART   ELE20880
C SAT    - PRODUCT OF S*A'                                              ELE20890
C ASAT   - PRODUCT OF A*SAT                                             ELE20900
C IAPAG  - 1 : INDEX FOR PINO-SUAREZ COLUMN CONSIDERING LOCAL BUCKLING  ELE20910
C          2 : PINO-SUAREZ COLUMN A END LOCAL BUCKLING OCCURRED         ELE20920
C          3 : PINO-SUAREZ COLUMN B END LOCAL BUCKLING OCCURRED         ELE20930
C          4 : PINO-SUAREZ COLUMN A AND B ENDS LOCAL BUCKLING OCCURRED  ELE20940
C          5 : INDEX FOR CHECKING BOX COLUMN STRENGH CRITERIA           ELE20950
C          6 : BOX COLUMN A END REACH TO FULLY PLASTIC                  ELE20960
C          7 : BOX COLUMN B END REACH TO FULLY PLASTIC                  ELE20970
C          8 : BOX COLUMN A AND B ENDS REACH TO FULLY PLASTIC           ELE20980
C---- ---------------------------------------------------------------- ELE20990
C                                                                       ELE21000
C------- CHOOSE OPTION                                                  ELE21010
      IF (IOPT.LT.1 .OR.IOPT.GT.14) THEN                                ELE21020
        WRITE(108,*) 'INVALID OPTION IN ELE 09, IOPT=',IOPT             ELE21030
        STOP                                                            ELE21040
      ENDIF                                                             ELE21050
C                                                                       ELE21060
      GOTO (100,200,300,400,500,600,700,800,900,1000,1100,              ELE21070
     &                                  1200,1300,1400) IOPT            ELE21080
C                                                                       ELE21090
  100 CONTINUE                                                          ELE21100
C                                                                       ELE21110
C------ ZERO STRAIN ENERGY                                             ELE21120
      ENGDAT(1) = 0                                                     ELE21130
      ENGDAT(2) = 0                                                     ELE21140
C                                                                       ELE21150
C------ INTERPERTATE INPUT DATA                                        ELE21160
      IELTYP= 9                                                         ELE21170
      DO 102 I=1,6                                                      ELE21180
  102 MATMLI(I) = RINPUT( I)                                            ELE21190
      NODEI  = RINPUT( 7)                                               ELE21200
      NODEJ  = RINPUT( 8)                                               ELE21210
      VY(1)  = RINPUT( 9)                                               ELE21220
      VY(2)  = RINPUT(10)                                               ELE21230
      VY(3)  = RINPUT(11)                                               ELE21240
      XS     = RINPUT(12)                                               ELE21250
      XE     = RINPUT(13)                                               ELE21260
      PKG    = RINPUT(14)                                               ELE21270
C                                                                       ELE21280
      IAPAG =0                                                          ELE21290
      STBFAG=0                                                          ELE21300
C                                                                       ELE21310
C------ GET INTERNAL NODE # ASSOCIATED WITH EXTERNAL NODE #'S          ELE21320
      JOINTI=IQUICK(NODEI,ID,NNODE)                                     ELE21330
      JOINTJ=IQUICK(NODEJ,ID,NNODE)                                     ELE21340
C                                                                       ELE21350
C------ PRINT DATA FOR PLOTTING                                        ELE21360
      IF (BTEST(IBUG,2)) WRITE(7,10) NODEJ,NODEI,(COORD(JOINTJ,I),      ELE21370
     &                                            I=1,3),(COORD(JOINTI,J),J=1,3)ELE21380
   10 FORMAT (I5,1X,I5,1P,6G12.4)                                       ELE21390
C                                                                       ELE21400
C-----------------------------------------------------------------      ELE21410
C  GET GLOBAL TO LOCAL TRANSFORMATION MATRIX                            ELE21420
C-----------------------------------------------------------------      ELE21430
C------ DEFINE VECTOR X, LENGTH                                        ELE21440
      VX(1)=COORD(JOINTJ,1)-COORD(JOINTI,1)                             ELE21450
      VX(2)=COORD(JOINTJ,2)-COORD(JOINTI,2)                             ELE21460
      VX(3)=COORD(JOINTJ,3)-COORD(JOINTI,3)                             ELE21470
      LENGTH=SQRT(VX(1)**2+VX(2)**2+VX(3)**2)-XS-XE                     ELE21480
      IF (LENGTH.LE.0) THEN                                             ELE21490
        WRITE(108,101) IELNO,JOINTI,JOINTJ,LENGTH                       ELE21500
  101   FORMAT('1X,20('*'),'ER',' ROR - LENGTH IS LE 0',/               ELE21510
     &    5X,'IELNO=',I5,5X,'JOINTI=',I5,5X,'JOINTJ=',I5,              ELE21520
     &    5X,'LENGTH',1P,G15.6/                                         ELE21530
     &    5X,'REVISE INPUT..... ..... .....'/)                         ELE21540
        LENGTH=1.0                                                      ELE21550
      ENDIF                                                             ELE21560
C------ GET LOCAL TO GLOBAL TRANSFORMATION MATRICIES CT AND B           ELE21570
      ICS=JTCOS(JOINTI)                                                 ELE21580
      ICE=JTCOS(JOINTJ)                                                 ELE21590
C                                                                       ELE21600
      CALL ROTXYZ(VX,VY,COSINE(1,1,ICS),CTS, XS ,0.,0.,BS,              ELE21610
     &            CONST(JOINTI,1),MAXNOD)                               ELE21620
      CALL ROTXYZ(VX,VY,COSINE(1,1,ICE),CTE,-XE ,0.,0.,BE,              ELE21630
     &            CONST(JOINTJ,1),MAXNOD)                               ELE21640
C                                                                       ELE21650
C------ ZERO MATRICIES                                                 ELE21660
      DO 40 I=1,12                                                      ELE21670
        FMAX(I)=0                                                       ELE21680
        F (I)=0                                                         ELE21690
        FT(I)=0                                                         ELE21700
        R (I)=0                                                         ELE21710
        RT(I)=0                                                         ELE21720
        V (I)=0                                                         ELE21730
        VL(I)=0                                                         ELE21740
        DO 40 J=1,12                                                    ELE21750
   40     A(I,J)=0.                                                    ELE21760
C                                                                       ELE21770
C------ ASSEMBLE TRANSFORMATION MATRIX A                               ELE21780
      DO 50 I=1,3                                                       ELE21790
      DO 50 J=1,3                                                       ELE21800
        A(I  ,J  )=CTS(I,J)                                             ELE21810
        A(I+3,J+3)=CTS(I,J)                                             ELE21820
        A(I+6,J+6)=CTE(I,J)                                             ELE21830
        A(I+9,J+9)=CTE(I,J)                                             ELE21840
        A(I+3,J  )=BS(I,J)                                              ELE21850
   50   A(I+9,J+6)=BE(I,J)                                             ELE21860
C                                                                       ELE21870
C------ GET MATERIAL PROPERTIES                                        ELE21880
       FALSE=.FALSE.                                                    ELE21890
       LSTTYP=0                                                         ELE21900
       DO 56 I=1,6                                                      ELE21910
        KK=0                                                            ELE21920
        CALL MATLIB(2,LSTTYP,FALSE,EESE,EPSE,DUCT(1,I),EXCR(1,I),       ELE21930
     &    FT,F,RT,R,0.0,0.0,LELEM,LMAT,MATMLI(I),MATTYP,SE(ISEMLI(I)),   ELE21940
     &    IELNO,DAMAGE)                                                 ELE21950
        MYZX(I)=MATTYP                                                  ELE21960
        IF (I.LE.4) THEN                                                ELE21970
          IF( MATTYP .NE. 9 .AND  MATTYP .NE. 10 .AND  MATTYP  NE  11   ELE21980
     &      .AND  MATTYP .NE  6 ) KK=1                                  ELE21990
        ELSE IF(I EQ.5) THEN                                            ELE22000
          IF( MATTYP  NE. 10 ) KK=1                                     ELE22010
        ELSE                                                            ELE22020
          IF( MATTYP .NE. 10 .AND. MATTYP  NE. 13) KK=1                 ELE22030
        ENDIF                                                           ELE22040
        IF (KK EQ.1) WRITE(108,190) IELNO,MATTYP,NODEI,NODEJ            ELE22050
  190   FORMAT (' ELEMENT ',I6,'  INCOMPATIBLE MATERIAL PROP.#',I5,     ELE22060
     &      ' START JOINT=',I5,' END JOINT=',I5,/                       ELE22070
     &      ' THE ELEMENT IS REJECTED AND EP','ROR CHECKING CONTINUES') ELE22080
   56   CONTINUE                                                        ELE22090
C                                                                       ELE22100
C  .. CHECK INTERACTIVE OPTION                                          ELE22110
      IF (MYZX(1).EQ.10.AND.MYZX(2).EQ.10.AND.                          ELE22120
     &    MYZX(3).EQ.10.AND.MYZX(4).EQ.10.AND.MYZX(6).EQ.10) THEN       ELE22130
        IF (SE(ISEMLI(1)+4).EQ.2.AND.SE(ISEMLI(2)+4).EQ.2 AND.          ELE22140
     &      SE(ISEMLI(3)+4).EQ.2.AND.SE(ISEMLI(4)+4).EQ.2) IAPAG=1      ELE22150
        IF (SE(ISEMLI(1)+4).EQ.3.AND.SE(ISEMLI(2)+4).EQ.3 AND           ELE22160
     &      SE(ISEMLI(3)+4).EQ.3 AND.SE(ISEMLI(4)+4).EQ.3) IAPAG=5      ELE22170
      ENDIF                                                             ELE22180
C                                                                       ELE22190
      EESE=0                                                            ELE22200
      EPSE=0                                                            ELE22210
C                                                                       ELE22220
C-- ... .. NO END RELEASE FOR THIS ELEMENT                              ELE22230
      REL=0                                                             ELE22240
C                                                                       ELE22250
C------- . SET STIFFNESS COIEFFICIENTS.                                 ELE22260
C                                                                       ELE22270
C  FOR BENDING IN Y AXIS:  LL=0                                         ELE22280
C  FOR BENDING IN Z AXIS.  LL=2                                         ELE22290
      KK=0                                                              ELE22300
  110 DO 205 LL=0,2,2                                                   ELE22310
        L=0                                                             ELE22320
```

```
      IF (LL.EQ.2) LO=1                                                 ELE02330
      IF( (MYZX(LL+1) .EQ. 9 .AND. MYZX(LL+2) .EQ. 9) .OR.              ELE02340
     &    (MYZX(LL+1) .EQ.11 .AND. MYZX(LL+2) .EQ.11) .OR.              ELE02350
     &    (MYZX(LL+1) .EQ  6 .AND. MYZX(LL+2) .EQ. 6) ) THEN            ELE02360
         STY(LL+1)=SE(ISEMLI(LL+1))                                     ELE02370
         STY(LL+2)=SE(ISEMLI(LL+2))                                     ELE02380
         IF( MYZX(LL+1) .EQ. 9 ) THEN                                   ELE02390
            SA1=SE(ISEMLI(LL+1)+60)                                     ELE02400
            SA2=SE(ISEMLI(LL+2)+60)                                     ELE02410
         ELSE IF( MYZX(LL+1) .EQ.11) THEN                               ELE02420
            SA1=SE(ISEMLI(LL+1)+30)                                     ELE02430
            SA2=SE(ISEMLI(LL+2)+30)                                     ELE02440
         ELSE IF( MYZX(LL+1) .EQ. 6) THEN                               ELE02450
            SA1=SE(ISEMLI(LL+1)+47)/LENGTH                              ELE02460
            SA2=SE(ISEMLI(LL+2)+47)/LENGTH                              ELE02470
            STY(LL+1)=STY(LL+1)/LENGTH                                  ELE02480
            STY(LL+2)=STY(LL+2)/LENGTH                                  ELE02490
            EI=SE(ISEMLI(LL+1)+48)                                      ELE02500
            TERM1=(LENGTH**2)/(12*EI*EI)                                ELE02510
            TERM2=LENGTH/(3*EI)                                         ELE02520
         ENDIF                                                          ELE02530
         IF ( MYZX(LL+1) .NE  6) THEN                                   ELE02540
            TERM1=3./(SA1**2)                                           ELE02550
            TERM2=2./SA1                                                ELE02560
         ENDIF                                                          ELE02570
         FLX1=0.                                                        ELE02580
         IF ( SA1 .NE. 0.) FLX1=(1/STY(LL+1))-(1/SA1)                   ELE02590
         FLX2=0                                                         ELE02600
         IF ( SA2 .NE. 0 ) FLX2=(1/STY(LL+2))-(1/SA2)                   ELE02610
         DCOP=TERM1 + TERM2*(FLX1+FLX2) + FLX1*FLX2                     ELE02620
         DCOP=1/DCOP                                                    ELE02630
         S23(LL+8) =DCOP*(TERM2 + FLX2)                                 ELE02640
         S23(LO+12)=DCOP*(TERM2 + FLX1)                                 ELE02650
         S23(LL+9)=DCOP*(0.5*TERM2)                                     ELE02660
      ELSE IF( MYZX(LL+1) .EQ  10 .AND. MYZX(LL+2)  EQ  10 ) THEN       ELE02670
         ELAS=SE(ISEMLI(LL+1)+4)                                        ELE02680
         SP  =SE(ISEMLI(LL+1)+5)                                        ELE02690
         EIYZ=SE(ISEMLI(LL+1)  )                                        ELE02700
         STY(LL+7)=SE(ISEMLI(LL+1)+3)                                   ELE02710
         STY(LL+8)=SE(ISEMLI(LL+2)+3)                                   ELE02720
         IF( STY(LL+7) .EQ. 1 .OR. STY(LL+8) .EQ. 1 ) STBFAG=1          ELE02730
         IF (KK0.EQ.0) THEN                                             ELE02740
          SE(ISEMLI(LL+1)+12)=SE(ISEMLI(LL+1)+1)*LENGTH/(6*EIYZ)        ELE02750
          SE(ISEMLI(LL+2)+12)=SE(ISEMLI(LL+2)+1)*LENGTH/(6*EIYZ)        ELE02760
         ENDIF                                                          ELE02770
C                                                                       ELE02780
         IF( ELAS .EQ  0 ) THEN                                         ELE02790
              S23(LL+8) =4*EIYZ/LENGTH                                  ELE02800
              S23(LO+12)=4*EIYZ/LENGTH                                  ELE02810
              S23(LL+9) =2*EIYZ/LENGTH                                  ELE02820
         ELSE                                                           ELE02830
              IF( STY(LL+7) .EQ. 0 .AND. STY(LL+8) .EQ  0 ) THEN        ELE02840
                   S23(LL+8) =4*EIYZ/LENGTH                             ELE02850
                   S23(LO+12)=4*EIYZ/LENGTH                             ELE02860
                   S23(LL+9) =2*EIYZ/LENGTH                             ELE02870
              ELSE IF( STY(LL+7) .EQ.1 .AND. STY(LL+8) .EQ. 0 ) THEN    ELE02880
                   S23(LL+8) =SP*(4*EIYZ/LENGTH)                        ELE02890
                   S23(LO+12)=SP*(4*EIYZ/LENGTH)+(1-SP)*(3*EIYZ/LENGTH) ELE02900
                   S23(LL+9) =SP*(2*EIYZ/LENGTH)                        ELE02910
              ELSE IF( STY(LL+7) .EQ. 0 .AND. STY(LL+8) .EQ. 1 ) THEN   ELE02920
                   S23(LL+8) =SP*(4*EIYZ/LENGTH)+(1-SP)*(3*EIYZ/LENGTH) ELE02930
                   S23(LO+12)=SP*(4*EIYZ/LENGTH)                        ELE02940
                   S23(LL+9) =SP*(2*EIYZ/LENGTH)                        ELE02950
              ELSE IF( STY(LL+7) .EQ. 1 .AND. STY(LL+8) .EQ. 1 ) THEN   ELE02960
                   S23(LL+8) =SP*(4*EIYZ/LENGTH)                        ELE02970
                   S23(LO+12)=SP*(4*EIYZ/LENGTH)                        ELE02980
                   S23(LL+9) =SP*(2*EIYZ/LENGTH)                        ELE02990
              ENDIF                                                     ELE03000
         ENDIF                                                          ELE03010
      ELSE                                                              ELE03020
       IF (LL.EQ.0) THEN                                                ELE03030
        WRITE(108,69) MYZX(LL+1),MYZX(LL+2),IELNO                       ELE03040
69    FORMAT(1X,'** INVALID MATERIAL TYPE MYA=',I2,' MYB=',I2,          ELE03050
     &          ' FOR ELEMENT NO. ',I5,' **')                           ELE03060
       ELSE                                                             ELE03070
        WRITE(108,79) MYZX(LL+1),MYZX(LL+2),IELNO                       ELE03080
79    FORMAT(1X,'** INVALID MATERIAL TYPE MZA=',I2,' MZB=',I2,          ELE03090
     &          ' FOR ELEMENT NO. ',I5,' **')                           ELE03100
       ENDIF                                                            ELE03110
      ENDIF                                                             ELE03120
205   CONTINUE                                                          ELE03130
C                                                                       ELE03140
C  FOR AXIAL IN X AXIS:    LL-6                                         ELE03150
C  FOR TORSION IN X AXIS:  LL-5                                         ELE03160
      DO 210 LL=5,6                                                     ELE03170
       LO=0                                                             ELE03180
       IF (LL.EQ.5) LO=7                                                ELE03190
      IF( MYZX(LL) .EQ. 10 ) THEN                                       ELE03200
         ELAS=SE(ISEMLI(LL)+4)                                          ELE03210
         SP  =SE(ISEMLI(LL)+5)                                          ELE03220
         EAGJ=SE(ISEMLI(LL)  )                                          ELE03230
         STY(LL)=SE(ISEMLI(LL)+3)                                       ELE03240
         IF( STY(LL) .EQ. 1) STBFAG=1                                   ELE03250
         IF (KK0.EQ.0) SE(ISEMLI(LL)+12)=SE(ISEMLI(LL)+1)*LENGTH/EAGJ   ELE03260
         IF( ELAS .EQ  0 ) THEN                                         ELE03270
              S23(LO)=EAGJ/LENGTH                                       ELE03280
         ELSE                                                           ELE03290
              IF( STY(LL) .EQ. 0 ) THEN                                 ELE03300
                 S23(LO)=EAGJ/LENGTH                                    ELE03310
              ELSE                                                      ELE03320
                 S23(LO)=SP*(EAGJ/LENGTH)                               ELE03330
              ENDIF                                                     ELE03340
         ENDIF                                                          ELE03350
       ELSE IF( MYZX(LL) .EQ. 13 ) THEN                                 ELE03360
         S23(LO)=SE(ISEMLI(LL))                                         ELE03370
       ELSE                                                             ELE03380
        IF (LL.EQ.6) THEN                                               ELE03390
         WRITE(108,89) MYZX(LL),IELNO                                   ELE03400
89    FORMAT(1X,'** INVALID MATERIAL TYPE MFXA=',I2,                    ELE03410
     &          ' FOR ELEMENT NO. ',I5,' **')                           ELE03420
        ELSE                                                            ELE03430
         WRITE(108,99) MYZX(LL),IELNO                                   ELE03440
99    FORMAT(1X,'** INVALID MATERIAL TYPE MXA=',I2,                     ELE03450
     &          ' FOR ELEMENT NO. ',I5,' **')                           ELE03460
        ENDIF                                                           ELE03470
       ENDIF                                                            ELE03480
210   CONTINUE                                                          ELE03490
C                                                                       ELE03500
      S23(1)=(S23(10)+2*S23(11)+S23(13))/(LENGTH**2)                    ELE03510
      S23(2)=(S23(10)+S23(11))/LENGTH                                   ELE03520
      S23(3)=(S23(13)-S23(11))/LENGTH                                   ELE03530
      S23(4)=(S23(8)+2*S23(9)-S23(12))/(LENGTH**2)                      ELE03540
      S23(5)=(S23(8)-S23(9))/LENGTH                                     ELE03550
      S23(6)=(S23(12)+S23(9))/LENGTH                                    ELE03560
C                                                                       ELE03570
C-------------- ------------ ... ------------------------------------- ELE03580
C  ASSEMBLE LOCAL STIFFNESS MATRIX                                      ELE03590
C ------------------------------------------------------------------    ELE03600
      DO 72 I=1,12                                                      ELE03610
      DO 72 J=1,12                                                      ELE03620
72    S(I,J)=0.                                                         ELE03630
      S( 1, 1) = S23(0)                                                 ELE03640
      S( 1, 7) =-S23(0)                                                 ELE03650
      S( 2, 2) = S23(1)                                                 ELE03660
      S( 2, 6) = S23(2)                                                 ELE03670
      S( 2, 8) =-S23(1)                                                 ELE03680
      S( 2,12) = S23(3)                                                 ELE03690
      S( 3, 3) = S23(4)                                                 ELE03700
      S( 3, 5) =-S23(5)                                                 ELE03710
      S( 3, 9) =-S23(4)                                                 ELE03720
      S( 3,11) =-S23(6)                                                 ELE03730
      S( 4, 4) = S23(7)                                                 ELE03740
      S( 4,10) =-S23(7)                                                 ELE03750
      S( 5, 5) = S23(8)                                                 ELE03760
      S( 5, 9) = S23(5)                                                 ELE03770
      S( 5,11) = S23(9)                                                 ELE03780
      S( 6, 6) = S23(10)                                                ELE03790
      S( 6, 8) =-S23(2)                                                 ELE03800
      S( 6,12) = S23(11)                                                ELE03810
      S( 7, 7) = S23(0)                                                 ELE03820
      S( 8, 8) = S23(1)                                                 ELE03830
      S( 8,12) =-S23(3)                                                 ELE03840
      S( 9, 9) = S23(4)                                                 ELE03850
      S( 9,11) = S23(6)                                                 ELE03860
      S(10,10) = S23(7)                                                 ELE03870
      S(11,11) = S23(12)                                                ELE03880
      S(12,12) = S23(13)                                                ELE03890
      DO 78 I=1,12                                                      ELE03900
      DO 78 J=I+1,12                                                    ELE03910
78    S( J, I) = S( I, J)                                               ELE03920
C                                                                       ELE03930
C----- CALCULATE SAT                                                    ELE03940
      CALL MULTM(3,12,SAT,S,A)                                          ELE03950
C--- CALCULATE ASAT                                                    ELE03960
      CALL MULTM(0,12,ASAT,A,SAT)                                       ELE03970
C                                                                       ELE03980
C----- DEGREES OF FREEDOM                                              ELE03990
C      REMOVE DOF'S THAT ARE CONSTRAINED TO THE SAME DOF...             ELE04000
      DO 75 I=1,6                                                       ELE04010
        IF ( IDOF(JOINTI,I).NE.IDOF(JOINTJ,I) ) THEN                    ELE04020
           LMT(I  )=IDOF(JOINTI,I)                                      ELE04030
           LMT(I+6)=IDOF(JOINTJ,I)                                      ELE04040
        ELSE                                                            ELE04050
           LMT(I  )= 0                                                  ELE04060
           LMT(I+6)= 0                                                  ELE04070
        ENDIF                                                           ELE04080
 75   CONTINUE                                                          ELE04090
C                                                                       ELE04100
C----- CONVERT TO HALF STORAGE MODE  BY COLUMNS                         ELE04110
C                                                                       ELE04120
      L=0                                                               ELE04130
      IELDOF=0                                                          ELE04140
      DO 90 I=1,12                                                      ELE04150
        IF (LMT(I).EQ.0                       ) GO TO 90                ELE04160
        IELDOF=IELDOF+1                                                 ELE04170
        LM(IELDOF)=LMT(I)                                               ELE04180
        DO 80 J=I,12                                                    ELE04190
          IF (LMT(J).EQ.0                     ) GO TO 80                ELE04200
          L=L+1                                                         ELE04210
          STIFF(L)=ASAT(I,J)                                            ELE04220
 80     CONTINUE                                                        ELE04230
 90   CONTINUE                                                          ELE04240
C  ** IOPT=3, KK0=1, RETURN **                                          ELE04250
      IF (KK0.EQ.1) RETURN                                              ELE04260
C-----------------------------------------------------------------      ELE04270
C  ASSEMBLE GEOMETRIC STIFFNESS MATRIX FOR A UNIT LOAD...               ELE04280
C-----------------------------------------------------------------      ELE04290
      LKG=L                                                             ELE04300
      IF ( PKG.EQ.0 .AND. KGDATA(1).NE.2 ) THEN                         ELE04310
         AXIAL=.FALSE.                                                  ELE04320
         REL=IBCLR(REL,6)                                               ELE04330
         GO TO 180                                                      ELE04340
      ENDIF                                                             ELE04350
C                                                                       ELE04360
C--------- DETERMINE IF AXIAL DEFORMATION IS CONSIDERED                 ELE04370
C          BIT #6 OF REL IS TRUE IF AXIAL DEFORMATION IS CONSIDERED     ELE04380
      DO 999 I=1,3                                                      ELE04390
         CS=CTS(I,1)                                                    ELE04400
         CE=CTE(I,1)                                                    ELE04410
         IF (ABS(CS) LE. .0001 .AND. ABS(CE).LE. .0001 ) GO TO 999      ELE04420
         IS=IDOF(JOINTI,I)                                              ELE04430
         IE=IDOF(JOINTJ,I)                                              ELE04440
         IF (IS.EQ.IE) GO TO 999                                        ELE04450
           REL=IBSET(REL,6)                                             ELE04460
 999     CONTINUE                                                       ELE04470
C                                                                       ELE04480
         AXIAL=BTEST(REL,6)                                             ELE04490
C                                                                       ELE04500
      IF (KGDATA(1).NE.0 .AND. KGDATA(2) NE.0 .AND. BTEST(REL,6)) THEN  ELE04510
C                                                                       ELE04520
C--------- DETERMINE THE AXIAL LOAD TO BE USED FOR KG..                 ELE04530
         IF (KGDATA(1).EQ 1) THEN                                       ELE04540
              PGEOM =  PKG                                              ELE04550
         ELSE IF (KGDATA(1).EQ.2) THEN                                  ELE04560
              PGEOM =  0.50*(F(1)-F(7))                                 ELE04570
         ENDIF                                                          ELE04580
C                                                                       ELE04590
C--ASSEMBLE LOCAL GEOMETRIC STIFFNESS MATRIX FOR A UNIT LOAD...         ELE04600
C          ZERO LOCAL KG MATRIX                                         ELE04610
           DO 998 I=1,12                                                ELE04620
           DO 998 J=1,12                                                ELE04630
 998         S(I,J)=0.                                                  ELE04640
C                                                                       ELE04650
C--------- LUMPED MASS FORM OF GEOMETRIC STIFFNESS                      ELE04660
         IF (KGDATA(2).EQ.1) THEN                                       ELE04670
              F1=1/LENGTH                                               ELE04680
              S( 2, 2)= F1                                              ELE04690
              S( 2, 8)=-F1                                              ELE04700
              S( 8, 8)= F1                                              ELE04710
              S( 3, 3)= F1                                              ELE04720
              S( 3, 9)=-F1                                              ELE04730
              S( 9, 9)= F1                                              ELE04740
C                                                                       ELE04750
C--------- CONSISTENT MASS FORM OF GEOMETRIC STIFFNESS                  ELE04760
         ELSE IF (KGDATA(2) EQ.2) THEN                                  ELE04770
C                                                                       ELE04780
C- ------- SET CONSISTENT MASS GEOMETRIC STIFFNESS COIEPFICIENTS.       ELE04790
           GAZ=  2 * LENGTH / 15.                                       ELE04800
           GBZ=  GAZ                                                    ELE04810
           GCZ=  LENGTH / 30.                                           ELE04820
           GDZ=  0.10                                                   ELE04830
           GEZ=  0.10                                                   ELE04840
           GFZ=  1.2 / LENGTH                                           ELE04850
           GAY=  2 * LENGTH / 15                                        ELE04860
           GBY=  GAY                                                    ELE04870
           GCY=  LENGTH / 30                                            ELE04880
           GDY=  0.10                                                   ELE04890
           GEY=  0.10                                                   ELE04900
           GFY= 1.2 / LENGTH                                            ELE04910
                S( 2, 2)=  GFZ                                          ELE04920
                S( 2, 6)=  GDZ                                          ELE04930
                S( 2, 8)= -GFZ                                          ELE04940
                S( 2,12)=  GEZ                                          ELE04950
                S( 6, 6)=  GAZ                                          ELE04960
                S( 6, 8)= -GDZ                                          ELE04970
                S( 6,12)=  GCZ                                          ELE04980
                S( 8, 8)=  GFZ                                          ELE04990
                S( 8,12)= -GEZ                                          ELE05000
                S(12,12)=  GBZ                                          ELE05010
                S( 3, 3)=  GFY                                          ELE05020
                S( 3, 5)= -GDY                                          ELE05030
                S( 3, 9)= -GFY                                          ELE05040
                S( 3,11)= -GEY                                          ELE05050
                S( 5, 5)=  GAY                                          ELE05060
                S( 5, 9)=  GDY                                          ELE05070
                S( 5,11)= -GCY                                          ELE05080
                S( 9, 9)=  GFY                                          ELE05090
                S( 9,11)=  GEY                                          ELE05100
                S(11,11)=  GBY                                          ELE05110
         ENDIF                                                          ELE05120
         DO 70 I=1,12                                                   ELE05130
         DO 70 J=I+1,12                                                 ELE05140
70       S( J, I) = S( I, J)                                            ELE05150
C                                                                       ELE05160
C --- CALCULATE SAT                                                     ELE05170
      CALL MULTM(3,12,SAT,S,A)                                          ELE05180
C--- CALCULATE ASAT                                                    ELE05190
      CALL MULTM(0,12,ASAT,A,SAT)                                       ELE05200
C                                                                       ELE05210
C  - CONVERT TO HALF STORAGE MODE  BY COLUMNS                           ELE05220
           KGDOF=0                                                      ELE05230
           DO 1090 I=1,12                                               ELE05240
```

```
            IF (LMT(I) EQ 0                          ) GO TO 1090          ELE05250
            KGDOF=KGDOF+1                                                  ELE05260
            LMXG(KGDOF)=LMT(I)                                             ELE05270
            DO 1080 J=1,2,-1                                               ELE05280
              IF (LMT(J).EQ.0                        ) GO TO 1080          ELE05290
              L=L-1                                                        ELE05300
              STIFF(L)=ASAT(I,J)                                           ELE05310
 1080         CONTINUE                                                     ELE05320
 1090         CONTINUE                                                     ELE05330
      ENDIF                                                               ELE05340
C                                                                         ELE05350
C- --- RELEASE UNUSED STORAGE                                            ELE05360
  180 IREL=156-L                                                          ELE05380
C                                                                         ELE05380
C     PRINT COLUMN DATA                                                   ELE05390
      IF (FIRST) WRITE(108,192)                                           ELE05400
  191 WRITE(108,193) NAME,IELNO,NODE1,NODEJ,LENGTH,                       ELE05410
     &               VY(1),VY(2),VY(3),XS,-XE,PKG                         ELE05420
      WRITE(108,194) MATMLI(6),MATMLI(5),MATMLI(1),MATMLI(2),             ELE05430
     &               MATMLI(3),MATMLI(4)                                  ELE05440
C                                                                         ELE05450
  192 FORMAT(/' ELEMENT 09,  IESDBEAM ELEMENT'//                          ELE05460
     &  T22,'#   START   END',3X,'LENGTH',3X,11('-'),'  Y-AXIS  ',        ELE05470
     &  12('-'),'  START DIST    END DIST     PKG')                       ELE05480
  193 FORMAT(1X,A15,2I6,1X,I6,1X,1P,G10.4,0P,                             ELE05490
     &       F9.5,' I',5F,F9.5,' J',F9.5,' K',1P,6E,3G12.4)               ELE05500
  194 FORMAT(16X,'MATERIAL # ',(PX  )'-',I2,1X,', (MX  )'-',I2,1X,        ELE05510
     &   ',(MY-A) ',I2,1X,', (MY-B)-',I2,1X,',(MZ-A)-',I2,1X,            ELE05520
     &   ', (MZ-B)-',I2 )                                                 ELE05530
      RETURN                                                              ELE05540
C------------------------ DETERMINE GEOMETRIC STIFF ---------------------ELE05550
  200 CONTINUE                                                            ELE05560
C  XG IS DETERMINED FOR A UNIT LOAD WITH IOPT=1,                          ELE05570
C  THE ACTUAL LOAD (PGEOM) IS DETERMINED HERE.                            ELE05580
C  KG IS MULTIPLIED BY PGEOM WHEN THE TOTAL STIFFNESS IS ASSEMBLED        ELE05590
C- NEGATIVE PGEOM IS COMPRESSION....                                      ELE05600
      AXIAL-BTEST(REL,6)                                                  ELE05610
C                                                                         ELE05620
C---------- DETERMINE THE AXIAL LOAD TO BE USED FOR XG...                 ELE05630
      IF (AXIAL) THEN                                                     ELE05640
         IF (XGDATA(1).EQ.1) THEN                                         ELE05650
            PGEOM =  PKG                                                  ELE05660
         ELSE IF (XGDATA(1).EQ.2) THEN                                    ELE05670
            PGEOM =  0.50*(F(1)-F(7))                                     ELE05680
         ENDIF                                                            ELE05690
      ENDIF                                                               ELE05700
C                                                                         ELE05710
      RETURN                                                              ELE05720
C------------------------ DETERMINE STIFFNESS ---------------------------ELE05730
  300 CONTINUE                                                            ELE05740
C                                                                         ELE05750
C     STY(1),STY(2),STY(3),STY(4),STY(5),STY(6),STY(7),STY(8),            ELE05760
C     STY(9), AND STY(10) CONTAIN STIFFNESSES, WHICH WERE DETERMINED IN   ELE05770
C     THE LAST CALL TO MA_TLIB                                            ELE05780
C                                                                         ELE05790
      IF STY(1) - SE(ISEMLI(1))    FOR MAT09,MAT11,MAT06                  ELE05800
C        STY(2) - SE(ISEMLI(2))    FOR MAT09,MAT11,MAT06                  ELE05810
C        STY(3) - SE(ISEMLI(3))    FOR MAT09,MAT11,MAT06                  ELE05820
C        STY(4) - SE(ISEMLI(4))    FOR MAT09,MAT11,MAT06                  ELE05830
C        STY(5) - SE(ISEMLI(5)*3)  FOR MAT10                              ELE05840
C        STY(6) - SE(ISEMLI(6)*3)  FOR MAT10                              ELE05850
C     OR                                                                  ELE05860
C     IF STY(7) - SE(ISEMLI(1)*3)  FOR MAT10                              ELE05870
C        STY(8) - SE(ISEMLI(2)*3)  FOR MAT10                              ELE05880
C        STY(9) - SE(ISEMLI(3)*3)  FOR MAT10                              ELE05890
C        STY(10) - SE(ISEMLI(4)*3) FOR MAT10                              ELE05900
C        STY(5) - SE(ISEMLI(5)*3)  FOR MAT10                              ELE05910
C        STY(6) - SE(ISEMLI(6)*3)  FOR MAT10                              ELE05920
C     THEN,                                                               ELE05930
C     THE STIFFNESS HAS NOT CHANGED, RETURN                               ELE05940
C                                                                         ELE05950
      DO 330 L=1,3,2                                                      ELE05960
       L1=L-1                                                             ELE05970
       IF ( MYZX(L) .NE. MYZX(L1) ) GOTO 330                              ELE05980
       IF ( MYZX(L) .EQ. 9 .OR. MYZX(L1) .EQ. 11 ) THEN                   ELE05990
        IF (STY(L) .NE. SE(ISEMLI(L)) .OR.                                ELE06000
     &      STY(L1).NE. SE(ISEMLI(L1))) GO TO 3 _                         ELE06010
       ELSE IF( MYZX(L) .EQ. 6 ) THEN                                     ELE06020
        IF (STY(L) .NE. (SE(ISEMLI(L))/LENGTH) .OR                        ELE06030
     &      STY(L1).NE. (SE(ISEMLI(L1))/LENGTH) ) GO TO 3                 ELE06040
       ELSE IF( MYZX(L) .EQ. 10 ) THEN                                    ELE06050
        IF (STY(L-6) .NE. SE(ISEMLI(L)*3) .OR.                            ELE06060
     &      STY(L1+6) NE. SE(ISEMLI(L1)*3) ) GOTO 3                       ELE06070
       ENDIF                                                              ELE06080
  330  CONTINUE                                                           ELE06090
       IF(STY(5) .NE. SE(ISEMLI(5)*3) ) GO TO 3                           ELE06100
       IF( MYZX(6) .EQ. 10 ) THEN                                         ELE06110
         IF(STY(6) .NE. SE(ISEMLI(6)*3) ) GO TO 3                         ELE06120
       ELSE IF( MYZX(6) .EQ. 13 ) THEN                                    ELE06130
         IF( S23(0) .NE. SE(ISEMLI(6)) ) GO TO 3                          ELE06140
       ENDIF                                                              ELE06150
C                                                                         ELE06160
      RETURN                                                              ELE06170
C ----- CONSTRUCT THE STIFFNESS MATRIX                                    ELE06180
    3 KKD-1                                                               ELE06190
      GOTO 110                                                            ELE06200
C------------------------ DETERMINE INCREMENTAL FORCES ------------------ELE06210
  400 CONTINUE                                                            ELE06220
      IF (PRINT AND.FIRST) WRITE(108,491) STEPID                          ELE06230
C                                                                         ELE06240
C------- CALL BMLOA TO DETERMINE THE FIXED END FORCES                     ELE06250
      IF (MAXELD GT.0) THEN                                               ELE06260
         CALL BMLOA(IELNO,FEMFLG,LENGTH,REL,MAXELD,NELD,NLOAD,            ELE06270
     &              FEM,IELD,ELD)                                         ELE06280
      ELSE                                                                ELE06290
         FEMFLG=.FALSE.                                                   ELE06300
      ENDIF                                                               ELE06310
C                                                                         ELE06320
C------ LOOP FOR EACH LOAD                                               ELE06330
      DO 430 LOAD=1,NLOAD                                                 ELE06340
C                                                                         ELE06350
C------ TRANSFORM FRAME DISPL INTO LOCAL DISPL                            ELE06360
       DO 410 I=1,3                                                       ELE06370
        R(I  )=0                                                          ELE06380
        R(I+3)=0                                                          ELE06390
        R(I+6)=0                                                          ELE06400
        R(I+9)=0                                                          ELE06410
        DO 410 J=1,3                                                      ELE06420
           D1=DISPL(IDOF(JOINTI,J  ),LOAD)                                ELE06430
           D2=DISPL(IDOF(JOINTI,J+3),LOAD)                                ELE06440
           D3=DISPL(IDOF(JOINTJ,J  ),LOAD)                                ELE06450
           D4=DISPL(IDOF(JOINTJ,J+3),LOAD)                                ELE06460
        R(I  )=R(I  )+CTS(J,I)*D1 - BS(J,I)*D2                            ELE06470
        R(I+3)=R(I+3)+CTS(J,I)*D2                                         ELE06480
        R(I+6)=R(I+6)+CTE(J,I)*D3 - BE(J,I)*D4                            ELE06490
  410   R(I+9)=R(I+9)+CTE(J,I)*D4                                         ELE06500
C                                                                         ELE06510
C-----   INCREMENTAL FORCES                                               ELE06520
       F( 1)= S23(0)*(R( 1)-R( 7))                                        ELE06530
       F( 2)- S23(1)*(R( 2)-R( 8))+S23(2)*R( 6)+S23(3)*R(12)              ELE06540
       F( 3)= S23(4)*(R( 3)-R( 9))-S23(5)*R( 5)-S23(6)*R(11)              ELE06550
       F( 4)- S23(7)*(R( 4)-R(10))                                        ELE06560
       F( 5)=-S23(5)*(R( 3)-R( 9))-S23(8)*R( 5)+S23(9)*R(11)              ELE06570
       F( 6)- S23(2)*(R( 2)-R( 8))+S23(10)*R( 6)+S23(11)*R(12)            ELE06580
       F( 7)=-S23(0)*(R( 1)-R( 7))                                        ELE06590
       F( 8)=-S23(1)*(R( 2) R( 8))-S23(2)*R( 6)-S23(3)*R(12)              ELE06600
       F( 9)-S23(4)*(R( 3)-R( 9))+S23(5)*R( 5)+S23(6)*R(11)               ELE06610
       F(10)=-S23(7)*(R( 4)-R(10))                                        ELE06620
       F(11)=-S23(6)*(R( 3)-R( 9))+ S23(9)*R( 5)-S23(12)*R(11)            ELE06630
       F(12)- S23(3)*(R( 2) R( 8))+ S23(11)*R( 6)-S23(13)*R(12)           ELE06640
C                                                                         ELE06650
C-----  RELATIVE ROTATION FOR BENDINGS                                    ELE06660
        R(5)-R(5)-(R(3)-R(9))/LENGTH                                      ELE06670
        R(6)-R(6)-(R(8)-R(2))/LENGTH                                      ELE06680
        R(11)-R(11)-(R(3)-R(9))/LENGTH                                    ELE06690
        R(12)-R(12)-(R(8)-R(2))/LENGTH                                    ELE06700
C                                                                         ELE06710
C------ SUBTRACT OF FIXED END FORCES                                      ELE06720
      IF (FEMFLG) THEN                                                    ELE06730
        DO 420 I=1,12                                                     ELE06740
          IF (I.EQ.1.OR.I.EQ.4.OR.I.EQ.7 OR.I.EQ.10) GOTO 420             ELE06750
          F(I)=F(I)-FEM(I,LOAD)                                           ELE06760
  420   CONTINUE                                                          ELE06770
      ENDIF                                                               ELE06780
C                                                                         ELE06790
C------- DETERMINE MAXIMUN VALUES FOR MULTIPLE LOAD CASES                 ELE06800
      IF (NLOAD.GT.1) THEN                                                ELE06810
        DO 425 I=1,12                                                     ELE06820
          IF ( ABS(F(I)).LE.ABS(FMAX(I+12-12+I)) ) GOTO 425               ELE06830
          DO 426 J=0,11                                                   ELE06840
  426       FMAX(I+12-11+J)=F(J+1)                                        ELE06850
  425   CONTINUE                                                          ELE06860
      ENDIF                                                               ELE06870
C                                                                         ELE06880
C------- PRINT FORCES AND DEFORMATIONS                                    ELE06890
      IF (PRINT) WRITE(108,493) IELNO,LOAD,NODE1,(R(J),J=1,6),            ELE06900
     &                          NODEJ,(R(J),J=7,12)                     ELE06910
      IF (PRINT) WRITE(108,492) IELNO,LOAD,NODE1,(F(J),J=1,6),            ELE06920
     &                          NODEJ,(F(J),J=7,12),STBFAG              ELE06930
  430 CONTINUE                                                            ELE06940
C                                                                         ELE06950
C ------THIS IS FOR SOLOSA USE ONLY                                       ELE06960
      IF( ICOMN .EQ. 1 .OR. ICOMN .EQ. 2 ) THEN                           ELE06970
        DO 431 I=1,12                                                     ELE06980
  431     RSAP(I)=F(I)                                                    ELE06990
      ENDIF                                                               ELE07000
C                                                                         ELE07010
C  -----CHECK STIFFNESS                                                   ELE07020
C       THIS IS PERFORMED ONLY FOR INELASTIC ANALYSIS                     ELE07030
        P(1)=FT( 5) + F( 5)                                               ELE07040
        P(3)=FT(11)+  F(11)                                               ELE07050
        P(3)=FT( 6) + F( 6)                                               ELE07060
        P(4)=FT(12)+  F(12)                                               ELE07070
        P(5)=FT(10)+  F(10)                                               ELE07080
        P(6)=FT( 7)+  F( 7)                                               ELE07090
      IF(.NOT  ELSTIC) THEN                                               ELE07100
C                                                                         ELE07110
C.....CHECK INTERACTIVE STRENGTH CRITERIA                                 ELE07120
        PY  = SE(ISEMLI(6)+11)                                            ELE07130
        TMPY= SE(ISEMLI(1)+11)                                            ELE07140
        TMP2= SE(ISEMLI(3)+11)                                            ELE07150
        DO 432 I=1,4                                                      ELE07160
  432   TMP(I)=SE(ISEMLI(I)+1)                                            ELE07170
        IF ( IAPAG .EQ. 1 ) THEN                                          ELE07180
          CALL ITERS(P(6),P(1),P(2),TMPY,PY,P(3),P(4),TMPZ,IAPAG)         ELE07190
          IF( IAPAG .EQ. 2 .OR. IAPAG .EQ  3 ) THEN                       ELE07200
            K11=IAPAG-1                                                   ELE07210
            K12=IAPAG+1                                                   ELE07220
            SE(ISEMLI(K11)+1)=ABS(P(K11))*SE(ISEMLI(K11)*8)               ELE07230
            SE(ISEMLI(K12)+1)=ABS(P(K12))*SE(ISEMLI(K12)*8)               ELE07240
            SE(ISEMLI(K11)+2)=-SE(ISEMLI(K11)+1)                          ELE07250
            SE(ISEMLI(K12)+2)=-SE(ISEMLI(K12)+1)                          ELE07260
          ELSE IF( IAPAG .EQ. 4 ) THEN                                    ELE07270
            DO 433 I=1,4                                                  ELE07280
              SE(ISEMLI(I)+1)=TMP(I)*SE(ISEMLI(I)*8)                      ELE07290
              SE(ISEMLI(I)+2)=-SE(ISEMLI(I)+1)                            ELE07300
  433         CONTINUE                                                    ELE07310
          ENDIF                                                           ELE07320
C                                                                         ELE07330
C... ASSIGN AXIAL RESIDUAL STRENGTH                                       ELE07340
C                                                                         ELE07350
        IF (SE(ISEMLI(6)+1).EQ.-SE(ISEMLI(6)+2).AND.IAPAG.NE.1) THEN      ELE07360
          SE(ISEMLI(6)+1)=SE(ISEMLI(6)+1)                                 ELE07370
          SE(ISEMLI(6)+2)=-ABS(P(6))*SE(ISEMLI(6)*8)                      ELE07380
          WRITE(108,705) IELNO,JSTEP,IAPAG                                ELE07390
  705     FORMAT(1X,'* ELEMENT ',I5,' LOCAL BUCKLING OCCURRED ',          ELE07400
     &           'AT STEP ',I5,' IAPAG=',I5,' *')                         ELE07410
        ENDIF                                                             ELE07420
        ELSE IF( IAPAG .EQ  2 .OR. IAPAG .EQ  3  ) THEN                   ELE07430
          IF (IAPAG.EQ.2) THEN                                            ELE07440
            K11=2                                                         ELE07450
            K12=4                                                         ELE07460
          ELSE                                                            ELE07470
            K11=1                                                         ELE07480
            K12=3                                                         ELE07490
          ENDIF                                                           ELE07500
          CALL ITERS(P(6),P(1),P(2),TMPY,PY,P(3),P(4),TMPZ,IAPAG)         ELE07510
          IF ( IAPAG .EQ. 4 ) THEN                                        ELE07520
            SE(ISEMLI(K11)+1)=ABS(P(K11))*SE(ISEMLI(K11)*8)               ELE07530
            SE(ISEMLI(K12)-1)=ABS(P(K12))*SE(ISEMLI(K12)*8)               ELE07540
            SE(ISEMLI(K11)+2)=-SE(ISEMLI(K11)-1)                          ELE07550
            SE(ISEMLI(K12)+2)=-SE(ISEMLI(K12)+1)                          ELE07560
            WRITE(108,705) IELNO,JSTEP,IAPAG                              ELE07570
          ENDIF                                                           ELE07580
        ELSE IF( IAPAG .EQ  5 ) THEN                                      ELE07590
          CALL ITERS2(P(6),P(1),P(2),TMPY,PY,P(3),P(4),TMPZ,IAPAG)        ELE07600
          IF( IAPAG .GT  5 ) THEN                                         ELE07610
            WRITE(108,706) IELNO,JSTEP,IAPAG                              ELE07620
  706       FORMAT(1X,'* ELEMENT ',I5, ' MODEL OVERSTRENGTH ',            ELE07630
     &           'AT STEP ',I5,' IAPAG=',I5,' *')                         ELE07640
          ENDIF                                                           ELE07650
        ENDIF                                                             ELE07660
C                                                                         ELE07670
C     ----- TRANSFER PERMENANT HYSTERESIS DATA TO TEMPERARY STORAGE      ELE07680
  377  ISE2=ISE/2                                                         ELE07690
       DO 522 I=1,ISE2                                                    ELE07700
  522   SE(I+ISE2)=SE(I)                                                  ELE07710
C                                                                         ELE07720
C---- CALCULATE INELASTIC ROTATION                                       ELE07730
       DO 310 I=1,2                                                       ELE07740
         I1=I+1                                                           ELE07750
         IF (MYZX(I).NE.MYZX(I1)) GOTO 310                                ELE07760
         IF (MYZX(I).NE.6.OR.MYZX(I).NE.9.OR.MYZX(I) NE.11) GOTO 310      ELE07770
         STY(I )=SE(ISEMLI(I ))                                           ELE07780
         STY(I1)=SE(ISEMLI(I1))                                           ELE07790
         IF( MYZX(I)   EQ. 9 ) THEN                                       ELE07800
           SA1=SE(ISEMLI(I )*60)                                          ELE07810
           SA2=SE(ISEMLI(I1)*60)                                          ELE07820
         ELSE IF( MYZX(I) .EQ 11 ) THEN                                   ELE07830
           SA1=SE(ISEMLI(I )*30)                                          ELE07840
           SA2=SE(ISEMLI(I1)*30)                                          ELE07850
         ELSE IF( MYZX(I) .EQ. 6 ) THEN                                   ELE07860
           SA1=SE(ISEMLI(I )*47)/LENGTH                                   ELE07870
           SA2=SE(ISEMLI(I1)*47)/LENGTH                                   ELE07880
           STY(I )=STY(I )/LENGTH                                         ELE07890
           STY(I1)=STY(I1)/LENGTH                                         ELE07900
         ENDIF                                                            ELE07910
         FLX1=0                                                           ELE07920
         IF ( SA1 .NE  0 ) FLX1=(1/STY(I ))-(1/SA1)                       ELE07930
         FLX2=0                                                           ELE07940
         IF ( SA2 .NE. 0 ) FLX2=(1/STY(I1))-(1/SA2)                       ELE07950
         RIELI(I )=ROIELI(I ) + FLX1*F(4-1)                               ELE07960
         RIELI(I1)=ROIELI(I1) + FLX2*F(10+I)                              ELE07970
  310    CONTINUE                                                         ELE07980
C                                                                         ELE07990
C      SET UP TEMTERARY - TOTAL LOADS,DISP'S                              ELE08000
C---  BENDING IN A END Y DIRECTION I-1                                    ELE08010
C---- BENDING IN B END Y DIRECTION I-2                                    ELE08020
C---- BENDING IN A END Z DIRECTION I-3                                    ELE08030
C---- BENDING IN B END Z DIRECTION I-4                                    ELE08040
C---- TORSION IN A END X DIRECTION I-5                                    ELE08050
C- -- AXIAL IN A END X DIRECTION   I-6                                    ELE08060
       DO 525 I=1,6                                                       ELE08070
         IF (I.LE.4) THEN                                                 ELE08080
           K11=I*(I-1)                                                    ELE08090
           IF (I LE.2) K11=6*I-1                                          ELE08100
           PL=FT(K11)                                                     ELE08110
           D =RT(K11)*R(K11)                                              ELE08120
           DL=RT(K11)                                                     ELE08130
           IF( MYZX(I)  EQ. 9 ) THEN                                      ELE08140
             D=RIELI(I) + P(I)/SE(ISEMLI(I)*60)                           ELE08150
             DL=ROIELI(I) + PL/SE(ISEMLI(I)*60)                           ELE08160
```

```
        ELSE IF( MYZX(I) .EQ. 11) THEN                                  ELE08170
          D-RIELI(I) * P(I)/SE(ISEMLI(I)+30)                            ELE08180
          DL-ROIELI(I) * PL/SE(ISEMLI(I)+30)                            ELE08190
        ELSE IF( MYZX(I) .EQ.  6) THEN                                  ELE08200
          IF( SE(ISEMLI(I)+47) .NE. 0 ) THEN                            ELE08210
            D=(RIELI(I)/LENGTH) * P(I)/SE(ISEMLI(I)+47)                 ELE08220
            DL=(ROIELI(I)/LENGTH) * PL/SE(ISEMLI(I)+47)                 ELE08230
          ENDIF                                                         ELE08240
        ENDIF                                                           ELE08250
        ELSE                                                            ELE08260
          K11=10                                                        ELE08270
          IF (I EQ 6) K11=7                                             ELE08280
          PL=PT(K11)                                                    ELE08290
          D =RT(K11)*R(K11)-RT(K11-6)-R(K11-6)                          ELE08300
          DL=RT(K11)-RT(K11-6)                                          ELE08310
        ENDIF                                                           ELE08320
C                                                                       ELE08330
        CALL MATLIB(3,LSTTYP,FALSE,EESE,EPSE,DUCT(1,I),EXCR(1,I),       ELE08340
     &              P(I),PL,D,DL,0 0,0.0,LELEM,LMAT,MATMLI(I),MATTYP,   ELE08350
     &              SE(ISE2+ISEMLI(I)),IELNO,DAMAGE)                    ELE08360
C                                                                       ELE08370
        IF (I.LE.4) THEN                                                ELE08380
         IF ( MATTYP .EQ. 9 ) THEN                                      ELE08390
          IF (SE(ISE2+ISEMLI(I)) .NE.SE(ISEMLI(I)) ) NEWK=.TRUE.        ELE08400
          ROIELI(I)=D-P(I)/SE(ISEMLI(I)+60)                             ELE08410
         ELSE IF( MATTYP .EQ.11 ) THEN                                  ELE08420
          IF (SE(ISE2+ISEMLI(I)) .NE.SE(ISEMLI(I)) ) NEWK=.TRUE         ELE08430
          ROIELI(I)=D-P(I)/SE(ISEMLI(I)+30)                             ELE08440
         ELSE IF( MATTYP .EQ. 6 ) THEN                                  ELE08450
          IF (SE(ISE2+ISEMLI(I)) .NE.SE(ISEMLI(I)) ) NEWK=.TRUE         ELE08460
          IF (SE(ISEMLI(I)+47) .NE. 0 ) THEN                            ELE08470
           ROIELI(I)=(D-P(I)/SE(ISEMLI(I)+47))*LENGTH                   ELE08480
          ENDIF                                                         ELE08490
         ELSE IF( MATTYP .EQ. 10 ) THEN                                 ELE08500
          IF (SE(ISE2+ISEMLI(I)+3).NE.SE(ISEMLI(I)+3) ) NEWK=.TRUE      ELE08510
         ENDIF                                                          ELE08520
        ELSE IF(I EQ.5) THEN                                            ELE08530
         IF (SE(ISE2-ISEMLI(I)+3) .NE.SE(ISEMLI(I)+3) ) NEWK=.TRUE.     ELE08540
        ELSE                                                            ELE08550
         IF ( MYZX(6) .EQ. 10 ) THEN                                    ELE08560
          IF ( SE(ISE2+ISEMLI(I)+3) .NE. SE(ISEMLI(I)+3) ) NEWK=.TRUE.  ELE08570
         ELSE IF( MYZX(6) .EQ. 13 ) THEN                                ELE08580
          IF ( SE(ISE2+ISEMLI(I)) .NE. SE(ISEMLI(I)) ) NEWK=.TRUE.      ELE08590
         ENDIF                                                          ELE08600
        ENDIF                                                           ELE08610
525     CONTINUE                                                        ELE08620
        ENDIF                                                           ELE08630
C                                                                       ELE08640
C------ CALCULATE THE UNBALACED MEMBER FORCES ON THE JOINT              ELE08650
C                                                                       ELE08660
        UP( 5)= -( PT( 5)+P( 5)-P(1) )                                  ELE08670
        UP(11)= -( PT(11)+P(11)-P(2) )                                  ELE08680
        UP( 6)= -( PT( 6)+P( 6)-P(3) )                                  ELE08690
        UP(12)= -( PT(12)+P(12)-P(4) )                                  ELE08700
        UP(10)= -( PT(10)+P(10)-P(5) )                                  ELE08710
        UP( 7)= -( PT( 7)+P( 7)-P(6) )                                  ELE08720
        UP( 9)=(UP( 5)+UP(11))/LENGTH                                   ELE08730
        UP( 3)= -UP( 9)                                                 ELE08740
        UP( 2)=(UP( 6)+UP(12))/LENGTH                                   ELE08750
        UP( 8)= -UP( 2)                                                 ELE08760
        UP( 1)= -UP( 7)                                                 ELE08770
        UP( 4)= -UP(10)                                                 ELE08780
C                                                                       ELE08790
        DO 440 I=1,IELDOF                                               ELE08800
           K=LM(I)                                                      ELE08810
           EPUB(I)=0                                                    ELE08820
           DO 460 J=1,12                                                ELE08830
 460          EPUB(I)=EPUB(I) + A(I,J)*UP(J)                           ELE08840
           PUB(K)=PUB(K)+EPUB(I)                                        ELE08850
 440    CONTINUE                                                        ELE08860
C                                                                       ELE08870
C------ ADJUST P FOR UNBALANCED LOAD P                                  ELE08880
        DO 445 I=1,12                                                   ELE08890
 445    P(I)=P(I)+UP(I)                                                 ELE08900
C                                                                       ELE08910
  491 FORMAT (/'     IE3D BEAM FORCES...',5X/7X,A/,                     ELE08920
     & '    ELEMENT LOAD          NODE',7X,'AXIAL',11X,'FY',13X,'FZ',   ELE08930
     & 11X,'TORSION',10X,'MY',13X,'MZ   STEPAG')                        ELE08940
  492 FORMAT (/,I10,I5,2X,'FORCE',I6,1P,6G15.6,                         ELE08950
     &          /15X    ,2X,'FORCE',I6,   6G15.6,3X,I1)               ELE08960
  493 FORMAT (/,I10,I5,2X,'DISPL',I6,1P,6G15 6,                         ELE08970
     &          /15X    ,2X,'DISPL',I6,   6G15.6)                     ELE08980
  494 FORMAT (7X,A)                                                     ELE08990
      RETURN                                                            ELE09000
C------------------------ DETERMINE TOTAL FORCES ----------------------ELE09010
  500 CONTINUE                                                          ELE09020
      IF (PRINT.AND.FIRST) WRITE(108,491) STEPID                        ELE09030
      IF(ICOMN .NE. 1 .AND. ICOMN .NE. 2 ) THEN                         ELE09040
C                                                                       ELE09050
C------ TOTAL FORCES AND DISPLACEMENTS                                  ELE09060
      DO 510 I=1,12                                                     ELE09070
         FT(I)=FT(I)+F(I)                                               ELE09080
         RT(I)=RT(I)+R(I)                                               ELE09090
  510 CONTINUE                                                          ELE09100
C                                                                       ELE09110
      DO 512 I=1,12                                                     ELE09120
         IF ( ABS(PT(I)).GT.ABS(FMAX(I*12-12+I)) ) THEN                 ELE09130
            DO 511 J=0,11                                               ELE09140
  511          FMAX(I*12-11+J)=FT(J+1)                                  ELE09150
         ENDIF                                                          ELE09160
  512 CONTINUE                                                          ELE09170
      ELSE                                                              ELE09180
         DO 513 I=1,12                                                  ELE09190
  513    FT(I)=RSAP(I)                                                  ELE09200
      ENDIF                                                             ELE09210
C                                                                       ELE09220
      LOAD=1                                                            ELE09230
      IF (PRINT) WRITE(108,493) IELNO,LOAD,NODE1,(RT(J),J=1,6),         ELE09240
     &                          NODEJ,(RT(J),J=7,12)                    ELE09250
      IF (PRINT) WRITE(108,492) IELNO,LOAD,NODEI,(FT(J),J=1,6),         ELE09260
     &                          NODEJ,(FT(J),J=7,12),STEPAG             ELE09270
C                                                                       ELE09280
C     --- - TRANSFER TEMPERARY HYSTERESIS DATA TO PERMANENT STORAGE     ELE09290
      ISE2=ISE/2                                                        ELE09300
      IF(  .NOT. ELSTIC) THEN                                           ELE09310
      DO 415 I=1,ISE2                                                   ELE09320
  415    SE(I)=SE(I+ISE2)                                               ELE09330
      ENDIF                                                             ELE09340
C                                                                       ELE09350
      RETURN                                                            ELE09360
C                                                                       ELE09370
C------------------------ DETERMINE FIXED END FORCES, AND ADD TO FORCE-ELE09380
  600 IF (MAXELD.LE.0) RETURN                                           ELE09390
C                                                                       ELE09400
C-----    CALL BMLOA TO DETERMINE THE FIXED END FORCES                  ELE09410
      CALL BMLOA(IELNO,FEMFLG,LENGTH,REL,MAXELD,NELD,NLOAD,PEM,IELD,ELD)ELE09420
      FEMFLG=FEMFLG                                                     ELE09430
      IF ( NOT. FEMFLG) RETURN                                          ELE09440
C                                                                       ELE09450
      DO 620 LOAD=1,NLOAD                                               ELE09460
        DO 610 I=1,12                                                   ELE09470
        GFEM(I  )=0                                                     ELE09480
        GFEM(I-3)=0.                                                    ELE09490
        GFEM(I-6)=0                                                     ELE09500
        GFEM(I-9)=0.                                                    ELE09510
        DO 610 J=1,3                                                    ELE09520
          D1=PEM(J  ,LOAD)                                              ELE09530
          D2=PEM(J+3,LOAD)                                              ELE09540
          D3=PEM(J+6,LOAD)                                              ELE09550
          D4=PEM(J+9,LOAD)                                              ELE09560
        GFEM(I  )=GFEM(I  )+CTS(I,J)*D1                                 ELE09570
        GFEM(I-3)=GFEM(I-3)+ BS(I,J)*D1 + CTS(I,J)*D2                   ELE09580
        GFEM(I-6)=GFEM(I-6)+CTE(I,J)*D3                                 ELE09590
  610   GFEM(I-9)=GFEM(I-9)+ BE(I,J)*D3 + CTE(I,J)*D4                   ELE09600
C                                                                       ELE09610
      DO 620 J=1,6                                                      ELE09620
        JI=IDOF(JOINTI,J)                                               ELE09630
        JJ=IDOF(JOINTJ,J)                                               ELE09640
        FORCE(JI,LOAD)=FORCE(JI,LOAD)+GFEM(J  )                         ELE09650
  620   FORCE(JJ,LOAD)=FORCE(JJ,LOAD)+GFEM(J+6)                         ELE09660
C                                                                       ELE09670
      RETURN                                                            ELE09680
C                                                                       ELE09690
C-------------------- WRITE MEMBER FORCES TO OUTPUT FILE --------------ELE09700
  700 CONTINUE                                                          ELE09710
C                                                                       ELE09720
      WRITE (IUNIT,710) NODEI,(FT(J),J=1,6),NODEJ,(FT(J),J=7,12),STEPAG ELE09730
      WRITE (IUNIT,710) NODEI,(RT(J),J=1,6),NODEJ,(RT(J),J=7,12)       ELE09740
  710 FORMAT (I6,1P,6G12.5/,I6,1P,6G12.5,I1)                            ELE09750
C                                                                       ELE09760
      RETURN                                                            ELE09770
C                                                                       ELE09780
C-------------------- READ AND PRINT MEMBER FORCES FROM OUTPUT FILE ----ELE09790
  800 CONTINUE                                                          ELE09800
      BACKSPACE(IUNIT)                                                  ELE09810
      READ (IUNIT,*  ) ITYPE,IELNO,IWRITE,TO,DT                        ELE09820
      WRITE(108,805)                                                    ELE09830
      ISTEP=-IWRITE                                                     ELE09840
  810 READ (IUNIT,815,END=830) NODEI,(FT(J),J=1,6),NODEJ,               ELE09850
     &                         (FT(J),J=7,12),STEPAG                  ELE09860
      READ (IUNIT,815,END=830) NODEI,(RT(J),J=1,6),NODEJ,(RT(J),J=7,12) ELE09870
  815 FORMAT (I6,6G12.5/,I6,6G12.5,I1)                                  ELE09880
C                                                                       ELE09890
        ISTEP=ISTEP-IWRITE                                              ELE09900
        T=TO+DT*ISTEP                                                   ELE09910
        IF (MOD(ISTEP,INC).EQ.0) THEN                                   ELE09920
          WRITE(108,820) ISTEP,T,NODEI,(FT(J),J=1,6),                   ELE09930
     &                   NODEJ,(FT(J),J=7,12),STEPAG                  ELE09940
          WRITE(108,920) (RT(J),J=1,12)                                 ELE09950
        ENDIF                                                           ELE09960
        GO TO 810                                                       ELE09970
C                                                                       ELE09980
  805 FORMAT (/'     IE3D BEAM FORCES...',/,                            ELE09990
     &        '    STEP    TIME        NODE',7X,'AXIAL',11X,'FY',       ELE10000
     &        13X,'FZ',11X,'TORSION',10X,'MY',13X,'MZ   STEPAG')       ELE10010
  820 FORMAT (/,I10,1P,G15 5,0P,I6,1P,6G15.6,/25X,I6,6G15.6,3X,I1)      ELE10020
  920 FORMAT ( 25X,'DISP  ',1P,6G15.6,/31X,6G15.6)                      ELE10030
C                                                                       ELE10040
  830 RETURN                                                            ELE10050
C                                                                       ELE10060
C-------------------- DETERMINE THE LENGTH OF THE MATERIAL DATA         ELE10070
  900 CONTINUE                                                          ELE10080
      ISE=0                                                             ELE10090
C---- LOOP FOR EACH MATERIAL -------                                    ELE10100
      DO 910 I=1,6                                                      ELE10110
        ISEMLI(I)=ISE+1                                                 ELE10120
        MATI  = RINPUT( I)                                              ELE10130
        CALL MATLIB(0,LSTTYP,FALSE,EESE,EPSE,DUCT,EXCR,                 ELE10140
     &              PT,F,RT,R,0.,0.,LELEM,LMAT,MATI,MATTYP,SE,IELNO,DAMAGE) ELE10150
  910   ISE=ISE+LELEM                                                   ELE10160
C                                                                       ELE10170
C----- DOUBLE THE STORAGE, TO ALLOW FOR TEMPOARY VALUES                 ELE10180
       ISE=ISE*2                                                        ELE10190
C                                                                       ELE10200
      RETURN                                                            ELE10210
C                                                                       ELE10220
C-------------------- DETERMINE THE TOTAL STRAIN ENERGY                ELE10230
 1000 CONTINUE                                                          ELE10240
      EESE=0                                                            ELE10250
      EPSE=0                                                            ELE10260
C                                                                       ELE10270
C     /* ENERGY FORMULATION IS NOT AVAILABLE FOR ELE09 ELEMENT */       ELE10280
        DO 1010 I=1,12                                                  ELE10290
 1010     EESE=EESE + (FT(I)+P(I))*(R(I)+RT(I))/2.                      ELE10300
C                                                                       ELE10310
      ENGDAT(1) = EESE                                                  ELE10320
      ENGDAT(2) = 0                                                     ELE10330
C                                                                       ELE10340
      RETURN                                                            ELE10350
C                                                                       ELE10360
C-------------------- DUCTILITIES(BASED ON DEFINITION ONE ONLY          ELE10370
 1100 CONTINUE                                                          ELE10380
C     - - NOTE ELE09 WAS LAST CALLED TO CALC INCIRMENTAL DISPL,         ELE10390
C              THE ENERGIES ASSOCIATED WITH INCR DISPLT ARE STORED IN   ELE10400
C              THE ADDRESSES: ISEMLI(1)+ISE/2                           ELE10410
C                             ISEMLI(2)+ISE/2                           ELE10420
C                             ISEMLI(3)+ISE/2                           ELE10430
C                             ISEMLI(4)+ISE/2                           ELE10440
C                             ISEMLI(5)+ISE/2                           ELE10450
C                             ISEMLI(6)+ISE/2                           ELE10460
C ........  BENDING ENERGY, DUCTILITY AND EXCURSION .....              ELE10470
      ISE2=ISE/2                                                        ELE10480
C---- BENDING IN A END Y DIRECTION I=1                                  ELE10490
C---- BENDING IN B END Y DIRECTION I=2                                  ELE10500
C---- BENDING IN A END Z DIRECTION I=3                                  ELE10510
C---- BENDING IN B END Z DIRECTION I=4                                  ELE10520
C---- TORSION IN A END X DIRECTION I=5                                  ELE10530
C---- AXIAL IN A END X DIRECTION   I=6                                  ELE10540
        DO 1105 I=1,6                                                   ELE10550
         CALL MATLIB(4,LSTTYP,FALSE,EESE,EPSE,DUCT(1,I),EXCR(1,I),      ELE10560
     &          PT,F,RT,R,0.0,0.0,LELEM,LMAT,MATMLI(I),MATTYP,          ELE10570
     &          SE(ISE2+ISEMLI(I)),IELNO,DAMAGE)                        ELE10580
 1105   CONTINUE                                                        ELE10590
        DO 1109 I=1,6                                                   ELE10600
          IF( ABS(DUCT(1,I))  LT. 1.) DUCT(1,I)=0.                      ELE10610
 1109   CONTINUE                                                        ELE10620
C                                                                       ELE10630
        IF( DUCFAG .EQ. 0 ) THEN                                        ELE10640
        IF(  ABS(DUCT(1,1))  .GE. 1 .OR. ABS(DUCT(1,2))  GE  1         ELE10650
     &  .OR  ABS(DUCT(1,3))  .GE. 1 .OR. ABS(DUCT(1,4))  GE. 1         ELE10660
     &  .OR  ABS(DUCT(1,5))  .GE. 1 .OR. ABS(DUCT(1,6))  GE  1 ) THEN  ELE10670
            DUCFAG=ABS(DUCT(1,1))                                       ELE10680
            IF( DUCFAG .LT. ABS(DUCT(1,2)) ) DUCFAG=ABS(DUCT(1,2))      ELE10690
            IF( DUCFAG .LT. ABS(DUCT(1,3)) ) DUCFAG=ABS(DUCT(1,3))      ELE10700
            IF( DUCFAG .LT. ABS(DUCT(1,4)) ) DUCFAG=ABS(DUCT(1,4))      ELE10710
            IF( DUCFAG .LT  ABS(DUCT(1,5)) ) DUCFAG=ABS(DUCT(1,5))      ELE10720
            IF( DUCFAG .LT. ABS(DUCT(1,6)) ) DUCFAG=ABS(DUCT(1,6))      ELE10730
        ENDIF                                                           ELE10740
        ENDIF                                                           ELE10750
C                                                                       ELE10760
       IF (PRINT.AND.FIRST) WRITE(108,1191)                             ELE10770
      IF (PRINT) WRITE(108,1192)IELNO,(DUCT(1,J),J=1,6)                 ELE10780
 1192 FORMAT (6X,I5,1X,6F11 5)                                          ELE10790
 1191 FORMAT (//'     IE3DBEAM  DUCTILITY(BASED ON DEFINITION 1):'/6X,  ELE10800
     &  ' ELEM#',' :MYA:      :MYB:      :MZA:',                       ELE10810
     &           '  :MZB:      :MX :      :FX :')                      ELE10820
      RETURN                                                            ELE10830
C                                                                       ELE10840
C                                                                       ELE10850
C-------------------- PRINT MAXIMUM ELEMENT LOADS                       ELE10860
C                                                                       ELE10870
 1200 CONTINUE                                                          ELE10880
      LOAD2=0                                                           ELE10890
      IF (FIRST) WRITE(108,494)                                         ELE10900
     &           ' COLUMN ( LOAD) REPRESENTS THE DOF WHICH'            ELE10910
     &          //' HAS MAXIMUM VALUE'                                 ELE10920
      IF (FIRST) WRITE(108,491) ' MAXIMUM VALUES FOR ALL STEPS  '       ELE10930
     &          //' NOTE  MAXIMUM VALUES WITH THE OTHER DOFS FORCES'   ELE10940
     &          //' ARE PRINT OUT AT THE SAME TIME'                    ELE10950
      DO 309 I=0,11                                                     ELE10960
           LOAD2=I+1                                                    ELE10970
           IF( FMAX(I*12+1-I)  EQ  0  ) GO TO 309                       ELE10980
           WRITE(108,492) IELNO,LOAD2,NODEI,(FMAX(I*12+J),J=1,6),       ELE10990
     &                    NODEJ,(FMAX(I*12+J),J=7,12),STEPAG            ELE11000
  309 CONTINUE                                                          ELE11010
C                                                                       ELE11020
      RETURN                                                            ELE11030
C                                                                       ELE11040
C                                                                       ELE11050
C-------------------- RESET ELEMENT FORCES                              ELE11060
C                                                                       ELE11070
 1300 CONTINUE                                                          ELE11080
```

```
C----- ZERO MATRICIES                                                    ELE11090
      DO 1340 I=1,12                                                    ELE11100
        FMAX(I+12-11)=0                                                 ELE11110
        F (I)=0                                                         ELE11120
        FT(I)=0                                                         ELE11130
        R (I)=0                                                         ELE11140
 1340   RT(I)=0                                                         ELE11150
      DO 1308 I=1,6                                                     ELE11160
 1308   IF (MYZX(I).EQ.10) TMP(I)=SE(ISEMLI(I)+12)                      ELE11170
      DO 1310 I=1,ISE                                                   ELE11180
        SE(I)=0                                                         ELE11190
 1310 CONTINUE                                                          ELE11200
      DO 1312 I=1,6                                                     ELE11210
 1312   IF (MYZX(I).EQ.10) SE(ISEMLI(I)+12)=TMP(I)                      ELE11220
C                                                                       ELE11230
      WRITE(108,1330) IELNO                                             ELE11240
 1330 FORMAT (' IE3D BEAM  ...  :ELEMENT #',I6,                         ELE11250
     & '  INTERNAL FORCES AND HYSTERESIS MODELS ARE RESET TO ZERO')     ELE11260
        FALSE= FALSE.                                                   ELE11270
        LSTTYP=0                                                        ELE11280
        DO 1335 I=1,6                                                   ELE11290
        CALL MATLIB(2,LSTTYP,FALSE,EESE,EPSE,DUCT,EXCR,                 ELE11300
     &      PT,F,RT,R,0.0,0.0,LELEM,LMAT,MATMLI(I),MATTYP,SE(ISEMLI(I)), ELE11310
     &      IELNO,DAMAGE)                                               ELE11320
 1335   CONTINUE                                                        ELE11330
C                                                                       ELE11340
      RETURN                                                            ELE11350
C                                                                       ELE11360
C-------------------- DAMAGE INDEX                                       ELE11370
C                                                                       ELE11380
 1400 DAMAGE = 0.                                                       ELE11390
C                                                                       ELE11400
      RETURN                                                            ELE11410
      END                                                               ELE11420
      SUBROUTINE MULTM(LL,N,A,B,C)                                      MULT0010
      DIMENSION A(N,N),B(N,N),C(N,N)                                    MULT0020
      DO 20 I=1,N                                                       MULT0030
        A(I,J)=0.                                                       MULT0040
        DO 20 K=1,N                                                     MULT0050
          IF (LL.EQ.0) THEN                                             MULT0060
            A(I,J)=A(I,J)+B(I,K)*C(K,J)                                 MULT0070
          ELSE IF(LL.EQ.1) THEN                                         MULT0080
            A(I,J)=A(I,J)+B(K,I)*C(K,J)                                 MULT0090
          ELSE                                                          MULT0100
            A(I,J)=A(I,J)+B(I,K)*C(J,K)                                 MULT0110
          ENDIF                                                         MULT0120
 20   CONTINUE                                                          MULT0130
      RETURN                                                            MULT0140
      END                                                               MULT0150
C                                                                       ITER0010
C     ------------------------------------------------------------      ITER0020
C     THIS SUBROUTINE IS FOR JUDGING BOX SECTION STRENGTH CRITERIA      ITER0030
C     BASED ON W.F.C'S PAPER                                            ITER0040
C                                                                       ITER0050
      SUBROUTINE ITER5(PI,TMXI,TMXJ,TMPX,PY,TMYI,TMYJ,TMPY,IAFAG)      ITER0060
C     NOTE:                                                             ITER0070
C      PI    : CURRENT AXIAL LOAD                                       ITER0080
C      PY    : AXIAL YIELDING LOAD CAPACITY                             ITER0090
C      TMXI  : CURRENT MOMENT IN X DIRECTION AT I END                   ITER0100
C      TMXJ  : CURRENT MOMENT IN X DIRECTION AT J END                   ITER0110
C      TMYI  : CURRENT MOMENT IN Y DIRECTION AT I END                   ITER0120
C      TMYJ  : CURRENT MOMENT IN Y DIRECTION AT J END                   ITER0130
C      TMPX  : FULLY PLASTIC MOMENT IN X DIRECTION                      ITER0140
C      TMPY  : FULLY PLASTIC MOMENT IN Y DIRECTION                      ITER0150
C      IAFAG : FLAG FOR JUDGING FULLY PLASTIC SECTION                   ITER0160
C              1: MEMBER TWO ENDS STILL ARE NOT FULLY PLASTIC           ITER0170
C              2: MEMBER A END IS IN FULLY PLASTIC                      ITER0180
C              3: MEMBER B END IS IN FULLY PLASTIC                      ITER0190
C              4: MEMBER A AND B ENDS ARE IN FULLY PLASTIC              ITER0200
C     ------------------------------------------------------------      ITER0210
C                                                                       ITER0220
      IF( PI .GE. 0 ) RETURN                                            ITER0230
      SP=ABS(PI/PY)                                                     ITER0240
      IF( SP .EQ. 0 ) THEN                                              ITER0250
        RETURN                                                          ITER0260
      ELSE IF( SP .LT. 1 .AND. SP .GT  0 ) THEN                         ITER0270
        A1=1.7 /(SP/LOG(SP))                                            ITER0280
      ELSE IF( SP .GE. 1 ) THEN                                         ITER0290
        IAFAG=4                                                         ITER0300
        RETURN                                                          ITER0310
      ENDIF                                                             ITER0320
      TMPCXI=1.2*(1-SP)*TMPX                                            ITER0330
      IF(ABS(TMPCXI) .GT. ABS(TMPX)) TMPCXI=TMPX                        ITER0340
      TMPCXJ=TMPCXI                                                     ITER0350
      TMPCYI=1.2*(1-SP)*TMPY                                            ITER0360
      IF(ABS(TMPCYI) .GT  ABS(TMPY)) TMPCYI=TMPY                        ITER0370
      TMPCYJ=TMPCYI                                                     ITER0380
      RMXI=ABS(TMXI)                                                    ITER0390
      RMXJ=ABS(TMXJ)                                                    ITER0400
      RMYI=ABS(TMYI)                                                    ITER0410
      RMYJ=ABS(TMYJ)                                                    ITER0420
C                                                                       ITER0430
      IF( IAFAG .EQ  1 ) THEN                                           ITER0440
        IF( (RMXI/TMPCXI) .GE. 1 .OR. (RMYI/TMPCYI) .GE  1 ) THEN       ITER0450
            YIELD1=1                                                    ITER0460
        ELSE IF((RMXI/TMPCXI) .LT. 1 .AND. (RMYI/TMPCYI) .LT. 1 ) THEN  ITER0470
            YIELD1=(RMXI/TMPCXI)**A1 + (RMYI/TMPCYI)**A1                ITER0480
        ENDIF                                                           ITER0490
        IF( (RMXJ/TMPCXJ) .GE. 1 .OR. (RMYJ/TMPCYJ) .GE. 1 ) THEN       ITER0500
            YIELD2=1                                                    ITER0510
        ELSE IF((RMXJ/TMPCXJ) .LT. 1 .AND. (RMYJ/TMPCYJ) .LT. 1 ) THEN  ITER0520
            YIELD2=(RMXJ/TMPCXJ)**A1 + (RMYJ/TMPCYJ)**A1                ITER0530
        ENDIF                                                           ITER0540
C                                                                       ITER0550
        IF( YIELD1 .GE. 1 .AND. YIELD2 .LT. 1 ) THEN                    ITER0560
          IAFAG=2                                                       ITER0570
        ELSE IF( YIELD1 .LT. 1 .AND. YIELD2 .GE. 1 ) THEN               ITER0580
          IAFAG=3                                                       ITER0590
        ELSE IF( YIELD1 .GE  1 .AND. YIELD2  GE. 1 ) THEN               ITER0600
          IAFAG=4                                                       ITER0610
        ENDIF                                                           ITER0620
      ELSE IF( IAFAG .EQ. 2 ) THEN                                      ITER0630
        IF( (RMXJ/TMPCXJ)  GE. 1 .OR  (RMYJ/TMPCYJ) .GE  1 ) THEN       ITER0640
          YIELD2=1                                                      ITER0650
        ELSE IF((RMXJ/TMPCXJ) .LT  1 .AND. (RMYJ/TMPCYJ) .LT. 1 ) THEN  ITER0660
          YIELD2=(RMXJ/TMPCXJ)**A1 + (RMYJ/TMPCYJ)**A1                  ITER0670
        ENDIF                                                           ITER0680
        IF( YIELD2  GE. 1) THEN                                         ITER0690
          IAFAG=4                                                       ITER0700
        ENDIF                                                           ITER0710
      ELSE IF( IAFAG  EQ  3 ) THEN                                      ITER0720
        IF( (RMXI/TMPCXI)  GE  1 .OR. (RMYI/TMPCYI)  .GE. 1 ) THEN      ITER0730
          YIELD1=1                                                      ITER0740
        ELSE IF((RMXI/TMPCXI) .LT. 1 .AND  (RMYI/TMPCYI) .LT. 1 ) THEN  ITER0750
          YIELD1=(RMXI/TMPCXI)**A1 + (RMYI/TMPCYI)**A1                  ITER0760
        ENDIF                                                           ITER0770
        IF( YIELD1 .GE  1) THEN                                         ITER0780
          IAFAG=4                                                       ITER0790
        ENDIF                                                           ITER0800
      ENDIF                                                             ITER0810
C                                                                       ITER0820
      RETURN                                                            ITER0830
      END                                                               ITER0840
C                                                                       ITER0010
C     THIS SUBROUTINE IS FOR JUDGING BOX SECTION STRENGTH CRITERIA      ITER0020
C     BASED ON W F.C'S PAPER                                            ITER0030
C                                                                       ITER0040
      SUBROUTINE ITER52(PI,TMXI,TMXJ,TMPX,PY,TMYI,TMYJ,TMPY,IAFAG )    ITER0050
C     ------------------------------------------------------------      ITER0060
C     NOTE:                                                             ITER0070
C      PI    : CURRENT AXIAL LOAD                                       ITER0080
C      PY    : AXIAL YIELDING LOAD CAPACITY                             ITER0090
C      TMXI  : CURRENT MOMENT IN X DIRECTION AT I END                   ITER0100
C      TMXJ  : CURRENT MOMENT IN X DIRECTION AT J END                   ITER0110
C      TMYI  : CURRENT MOMENT IN Y DIRECTION AT I END                   ITER0120
C      TMYJ  : CURRENT MOMENT IN Y DIRECTION AT J END                   ITER0130
C      TMPX  : FULLY PLASTIC MOMENT IN X DIRECTION                      ITER0140
C      TMPY  : FULLY PLASTIC MOMENT IN Y DIRECTION                      ITER0150
C      IAFAG : FLAG FOR JUDGING FULLY PLASTIC SECTION                   ITER0160
C              5. MEMBER TWO ENDS STILL ARE NOT FULLY PLASTIC           ITER0170
C              6: MEMBER A END IS IN FULLY PLASTIC                      ITER0180
C              7: MEMBER B END IS IN FULLY PLASTIC                      ITER0190
C              8: MEMBER A AND B ENDS ARE IN FULLY PLASTIC              ITER0200
C     ------------------------------------------------------------      ITER0210
C                                                                       ITER0220
      IF( PI .GE. 0 ) RETURN                                            ITER0230
      SP=ABS(PI/PY)                                                     ITER0240
      IF( SP .EQ. 0 ) THEN                                              ITER0250
        RETURN                                                          ITER0260
      ELSE IF( SP .LT. 1 .AND  SP .GT. 0 ) THEN                         ITER0270
        A1=1.7 /(SP/LOG(SP))                                            ITER0280
      ELSE IF( SP .GE  1 ) THEN                                         ITER0290
        IAFAG=4                                                         ITER0300
        RETURN                                                          ITER0310
      ENDIF                                                             ITER0320
      TMPCXI=1.2*(1-SP)*TMPX                                            ITER0330
      IF(ABS(TMPCXI) .GT  ABS(TMPX)) TMPCXI=TMPX                        ITER0340
      TMPCXJ=TMPCXI                                                     ITER0350
      TMPCYI=1.2*(1-SP)*TMPY                                            ITER0360
      IF(ABS(TMPCYI) .GT  ABS(TMPY)) TMPCYI=TMPY                        ITER0370
      TMPCYJ=TMPCYI                                                     ITER0380
      RMXI=ABS(TMXI)                                                    ITER0390
      RMXJ=ABS(TMXJ)                                                    ITER0400
      RMYI=ABS(TMYI)                                                    ITER0410
      RMYJ=ABS(TMYJ)                                                    ITER0420
C                                                                       ITER0430
      IF( IAFAG .EQ  1 ) THEN                                           ITER0440
        IF( (RMXI/TMPCXI) .GE. 1 .OR. (RMYI/TMPCYI) .GE  1 ) THEN       ITER0450
            YIELD1=1                                                    ITER0460
        ELSE IF((RMXI/TMPCXI) .LT. 1 .AND. (RMYI/TMPCYI) .LT. 1 ) THEN  ITER0470
            YIELD1=(RMXI/TMPCXI)**A1 + (RMYI/TMPCYI)**A1                ITER0480
        ENDIF                                                           ITER0490
        IF( (RMXJ/TMPCXJ) .GE. 1 .OR. (RMYJ/TMPCYJ) .GE. 1 ) THEN       ITER0500
            YIELD2=1                                                    ITER0510
        ELSE IF((RMXJ/TMPCXJ) .LT. 1 .AND. (RMYJ/TMPCYJ) .LT. 1 ) THEN  ITER0520
            YIELD2=(RMXJ/TMPCXJ)**A1 + (RMYJ/TMPCYJ)**A1                ITER0530
        ENDIF                                                           ITER0540
C                                                                       ITER0550
        IF( YIELD1 .GE. 1 .AND. YIELD2 .LT. 1 ) THEN                    ITER0560
          IAFAG=6                                                       ITER0570
        ELSE IF( YIELD1 .LT. 1  AND. YIELD2 .GE. 1 ) THEN               ITER0580
          IAFAG=7                                                       ITER0590
        ELSE IF( YIELD1 .GE. 1  AND. YIELD2 .GE. 1 ) THEN               ITER0600
          IAFAG=8                                                       ITER0610
        ENDIF                                                           ITER0620
C                                                                       ITER0630
      ENDIF                                                             ITER0640
C                                                                       ITER0650
      RETURN                                                            ITER0660
      END                                                               ITER0670
C-----------------------------------------------------------------------*ELEL0010
      SUBROUTINE ELELIB                                                 ELEL0020
     &     (IOPT,LSTTYP,PRINT,IREL,HEAD,NAME,EESE,EPSE,DAMAGE,DUCFAG,   ELEL0030
     &      IELNO ,IELDOP,KGDOP,PGEOM,RINPUT,AXIAL,IDISP,ILOAD,MSTOR)   ELEL0040
C                                                                       ELEL0050
      LOGICAL KSAME,ERR,FIRST,PRINT,HEAD,AXIAL                          ELEL0060
      CHARACTER*80 NAME                                                 ELEL0070
      DIMENSION RINPUT(100)                                             ELEL0080
$INCLUDE.'ZCOMN'                                                        ELEL0090
C                                                                       ELEL0100
      IW    = IZ                                                        ELEL0110
      IC    = NZ(IZELE-IELNO)                                           ELEL0120
      IZELNO= IELNO                                                     ELEL0130
      IELTYP = NZ(IC)                                                   ELEL0140
      IF (LSTTYP.EQ.IELTYP) THEN                                        ELEL0150
          FIRST= .FALSE.                                                ELEL0160
      ELSE                                                              ELEL0170
          FIRST= TRUE.                                                  ELEL0180
          IF (HEAD) WRITE(108,305) TITLE(1),TITLE(2)                    ELEL0190
  305     FORMAT ('1 STRUCTURE....: ',A,/'  SOLUTION ... : ',A,/)       ELEL0200
      ENDIF                                                             ELEL0210
C                                                                       ELEL0220
      MSTOR=0                                                           ELEL0230
C                                                                       ELEL0240
      IF (IELTYP.EQ.1)  THEN                                            ELEL0250
       IZLM=IC+64                                                       ELEL0260
       IZKE=IC+281+NZ(IC+133)                                           ELEL0270
       IZLMKG=IC+121                                                    ELEL0280
       IF( MAXELD .LE. 0 ) MAXELD=1                                     ELEL0290
       IF (IOPT.NE.3) CALL ELE01                                        ELEL0300
     &  (IOPT      ,FIRST     ,PRINT     ,NZ(IZID)   ,NZ(IZIDOF) ,     ELEL0310
     &   Z(IZCORD) ,Z(IZCOS ) ,NZ(IZJCOS) ,Z(IZCNST) ,Z(IDISP)  ,      ELEL0320
     &   Z(ILOAD)  ,IREL      ,NZ(IZKGDT+1),PGEOM     ,NZ(IZIELD) ,     ELEL0330
     &   Z(IZELD)  ,NAME      ,AXIAL     ,IELNO      ,RINPUT     ,      ELEL0340
     &  Z(IZPUB)   ,EESE      ,EPSE      ,DAMAGE     ,NZ(IC    ) ,      ELEL0350
     &  NZ(IC+  1) , Z(IC+  2)  , Z(IC+ 16)  , Z(IC+ 28) , Z(IC+ 40) ,  ELEL0360
     &   Z(IC+ 52) ,NZ(IZLM )  , Z(IC+ 76)  , Z(IC+ 85) , Z(IC+ 94) ,  ELEL0370
     &   Z(IC+103) , Z(IC+112) ,NZ(IC+113)  ,NZ(IC+114) ,NZ(IC+115) ,  ELEL0380
     &  NZ(IC+116) ,NZ(IC+117) , Z(IC+118)  ,NZ(IC+119) ,NZ(IC+120) ,  ELEL0390
     &  NZ(IZLMKG) ,NZ(IC+133) , Z(IC+134)  , Z(IC+136) ,NZ(IC+280) ,  ELEL0400
     &   Z(IC+281) , Z(IZKE )  ,  Z(IW    ) )                         ELEL0410
C                                                                       ELEL0420
       MSTOR =NZ(IC+133)                                                ELEL0430
       IELDOP=NZ(IC+1)                                                  ELEL0440
       IZLMKG=IC+121                                                    ELEL0450
       IZKEKG=IZKE+ NZ(IC+120)                                          ELEL0460
       KGDOP =NZ(IC+119)                                                ELEL0470
C                                                                       ELEL0480
      ELSE IF (IELTYP.EQ.2)  THEN                                       ELEL0490
       AXIAL=.FALSE.                                                    ELEL0500
       IZKE=IC+55+NZ(IC+12)                                             ELEL0510
       IZLM=IC+40                                                       ELEL0520
        CALL ELE02                                                      ELEL0530
     &  (IOPT       ,FIRST     ,PRINT     ,NZ(IZID)   ,NZ(IZIDOF) ,     ELEL0540
     &   Z(IZCORD)  ,Z(IZCOS)  ,NZ(IZJCOS) ,Z(IZCNST)  ,Z(IDISP )  ,     ELEL0550
     &   Z(IZDVEL)  ,Z(IZPUB)  ,IREL       ,NAME       ,IELNO      ,     ELEL0560
     &   RINPUT     ,EESE      ,EPSE      ,DAMAGE     ,DUCFAG     ,     ELEL0570
     &   NZ(IC   ) ,NZ(IC+  1) ,NZ(IC+  2) ,NZ(IC+  3) ,NZ(IC+  4) ,     ELEL0580
     &   NZ(IC+  5) , Z(IC+  6)  , Z(IC+  7)  , Z(IC+  8)  , Z(IC+  9) ,  ELEL0590
     &    Z(IC+ 10) , Z(IC+ 11) ,NZ(IC+ 12)  , Z(IC+ 13)  , Z(IC+ 14) ,  ELEL0600
     &   NZ(IC+ 15) , Z(IC+ 16) ,NZ(IC+ 40) ,NZ(IC+ 52) , Z(IC+ 53) ,     ELEL0610
     &    Z(IC+ 55) , Z(IZKE ) )                                        ELEL0620
       MSTOR =NZ(IC+12)                                                 ELEL0630
       IELDOP=NZ(IC+1)                                                  ELEL0640
       IZLMKG=0                                                         ELEL0650
       IZKEKG=0                                                         ELEL0660
       KGDOP =0                                                         ELEL0670
C                                                                       ELEL0680
      ELSE IF (IELTYP.EQ.3)  THEN                                       ELEL0690
       IZKE=IC+167+NZ(IC+5)                                             ELEL0700
        CALL ELE03                                                      ELEL0710
     &  (IOPT       ,FIRST     ,PRINT     ,PGEOM      ,NAME       ,     ELEL0720
     &   NZ(IZID)   ,NZ(IZIDOF) ,Z(IZCORD) ,Z(IZCOS)  ,NZ(IZJCOS) ,     ELEL0730
     &   Z(IZCNST)  ,Z(IDISP )  ,Z(IZDVEL) ,Z(IZPUB)  ,IREL      ,     ELEL0740
     &   NZ(IZKGDT+1),IELNO     ,RINPUT    ,AXIAL      ,EESE       ,     ELEL0750
     &   EPSE       ,DAMAGE     ,DUCFAG    ,NZ(IC    ) ,NZ(IC+  1) ,     ELEL0760
     &   NZ(IC+  2) ,NZ(IC+  5) ,NZ(IC+  6) ,NZ(IC+ 10) , Z(IC+ 14) ,     ELEL0770
     &    Z(IC+ 17) , Z(IC+ 20)  , Z(IC+ 23) , Z(IC+ 26) , Z(IC+ 29) ,     ELEL0780
     &    Z(IC+ 32) ,NZ(IC+ 33) ,NZ(IC+ 34) ,NZ(IC+ 58) , Z(IC+ 82) ,     ELEL0790
     &    Z(IC+154) , Z(IC+155) , Z(IC+156) ,NZ(IC+159) , Z(IC+160) ,     ELEL0800
     &    Z(IC+161) , Z(IC+167) , Z(IZKE ) )                            ELEL0810
       MSTOR =NZ(IC+5)                                                  ELEL0820
       IELDOP=NZ(IC+1)                                                  ELEL0830
       IZLM=IC+34                                                       ELEL0840
       IZLMKG=IC+58                                                     ELEL0850
       IZKEKG=IC+167+NZ(IC+5) + NZ(IC+33) 1                             ELEL0860
       KGDOP =NZ(IC+159)                                                ELEL0870
C                                                                       ELEL0880
      ELSE IF (IELTYP EQ 8)  THEN                                       ELEL0890
       AXIAL=.FALSE.                                                    ELEL0900
       IZKE=IC+55+NZ(IC+12)                                             ELEL0910
```

```
      IZLM-IC+40                                                        ELEL0920
       CALL ELE06                                                       ELEL0930
     &  (IOPT       ,FIRST      ,PRINT      ,NAME       ,IELNO      ,   ELEL0940
     &   NZ(IZID)   ,NZ(IZIDOF) ,Z(IZCORD) ,Z(IZCOS)   ,NZ(IZJCOS) ,   ELEL0950
     &   Z(IZCNST)  ,Z(IDISP )  ,Z(IZDVEL) ,Z(IZFUB)   ,IREL,          ELEL0960
     &   RINPUT     ,EESE       ,EPSE       ,DAMAGE     ,DUCFAG     ,   ELEL0970
     &   NZ(IC    ) ,NZ(IC+  1) ,NZ(IC+  2) ,NZ(IC+  3) ,NZ(IC+  4) ,   ELEL0980
     &   NZ(IC+  5) , Z(IC+  6) , Z(IC+  7) , Z(IC+  8) , Z(IC+  9) ,   ELEL0990
     &    Z(IC+ 10) , Z(IC+ 11) ,NZ(IC+ 12) , Z(IC+ 13) , Z(IC+ 14) ,   ELEL1000
     &   NZ(IC+ 15) , Z(IC+ 16) ,NZ(IC+ 40) ,NZ(IC+ 52) , Z(IC+ 53) ,   ELEL1010
     &    Z(IC+ 55) , Z(IZKE  ) )                                       ELEL1020
       MSTOR =NZ(IC+12)                                                 ELEL1030
       IELDOF=NZ(IC+1)                                                  ELEL1040
       IZLMKG=0                                                         ELEL1050
       IZKEKG=0                                                         ELEL1060
       KGDOF =0                                                         ELEL1070
C                                                                       ELEL1080
      ELSE IF (IELTYP.EQ.9)  THEN                                       ELEL1090
       IZLM=IC+73                                                       ELEL1100
       IZKE=IC+458+NZ(IC+141)                                           ELEL1110
       IZLMKG=IC+129                                                    ELEL1120
       IF( MAXELD .LE. 0 ) MAXELD=1                                     ELEL1130
       CALL ELE09                                                       ELEL1140
     &  (IOPT       ,FIRST      ,PRINT      ,NAME       ,AXIAL      ,   ELEL1150
     &   NZ(IZID)   ,NZ(IZIDOF) , Z(IZCORD) , Z(IZCOS ) ,NZ(IZJCOS),   ELEL1160
     &    Z(IZCNST) , Z(IDISP)  , Z(ILOAD)  ,IREL       ,             ,   ELEL1170
     &   NZ(IZKGDT+1) ,PGEOM     ,NZ(IZIELD) ,Z(IZELD)   ,IELNO      ,   ELEL1180
     &   RINPUT     , Z(IZFUB)  ,EESE       ,EPSE       ,DAMAGE     ,   ELEL1190
     &   DUCFAG     ,NZ(IC)     ,NZ(IC+  1) ,NZ(IC+2)   , Z(IC+6)  ,   ELEL1200
     &   NZ(IC+18)  , Z(IC+24)  , Z(IC+25)  , Z(IC+37)  , Z(IC+49) ,   ELEL1210
     &    Z(IC+61)  ,NZ(IZLM  ) , Z(IC+ 85) , Z(IC+ 94) , Z(IC+103),   ELEL1220
     &    Z(IC+112) ,NZ(IC+121) ,NZ(IC+122) ,NZ(IC+123) ,NZ(IC+124),   ELEL1230
     &   NZ(IC+125) , Z(IC+126) ,NZ(IC+127) ,NZ(IC+128) ,NZ(IZLMKG),   ELEL1240
     &   NZ(IC+141) , Z(IC+142) , Z(IC+144) ,NZ(IC+288) ,NZ(IC+289),   ELEL1250
     &   NZ(IC+290) , Z(IC-296) , Z(IC+310) , Z(IC+314),               ELEL1260
     &    Z(IC+658) , Z(IZKE)   , Z(IW)     )                         ELEL1270
C                                                                       ELEL1280
       MSTOR =NZ(IC+141)                                                ELEL1290
       IELDOF=NZ(IC+1)                                                  ELEL1300
       IZLMKG=IC+129                                                    ELEL1310
       IZKEKG=IZKE+ NZ(IC+128)                                          ELEL1320
       KGDOF =NZ(IC+127)                                                ELEL1330
C                                                                       ELEL1340
      ELSE IF (IELTYP.EQ.12)  THEN                                      ELEL1350
       AXIAL=.FALSE.                                                    ELEL1360
       IZLM=IC+50                                                       ELEL1370
       IZKE=IC+540+NZ(IC+104)                                           ELEL1380
       IF( MAXELD .LE. 0 ) MAXELD=1                                     ELEL1390
       CALL ELE12                                                       ELEL1400
     &  (IOPT       ,FIRST      ,PRINT      ,IREL       ,NAME       ,   ELEL1410
     &   NZ(IZID)   ,NZ(IZIDOF) ,Z(IZCORD) , Z(IZCOS ) ,NZ(IZJCOS) ,   ELEL1420
     &    Z(IZCNST) , Z(IDISP)  , RINPUT    , Z(IZFUB)  ,EESE       ,   ELEL1430
     &    EPSE      ,DAMAGE     ,NZ(IC)     ,NZ(IC+  1) ,Z(IC+  2)  ,   ELEL1440
     &    Z(IC+ 14) , Z(IC+ 26) , Z(IC+ 36) ,NZ(IZLM  ) , Z(IC+ 62) ,   ELEL1450
     &    Z(IC+ 71) , Z(IC+ 80) , Z(IC+ 89) ,NZ(IC+ 98) ,NZ(IC+ 99) ,   ELEL1460
     &   NZ(IC+100) ,NZ(IC+101) ,NZ(IC+102) , Z(IC+103) ,IELNO       ,   ELEL1470
     &   NZ(IC+104) , Z(IC+105) , Z(IC+107) , Z(IC+251) ,Z(IC+395)  ,   ELEL1480
     &   NZ(IC+539) , Z(IC+540) , Z(IZKE  ) )                           ELEL1490
C                                                                       ELEL1500
       NZ(IC+62)=1                                                      ELEL1510
       MSTOR =NZ(IC+104)                                                ELEL1520
       IELDOF=NZ(IC+1)                                                  ELEL1530
       IZLMKG=0                                                         ELEL1540
       IZKEKG=0                                                         ELEL1550
       KGDOF =0                                                         ELEL1560
C                                                                       ELEL1570
      ELSE IF (IELTYP.EQ.14)  THEN                                      ELEL1580
       IZKE=IC+167+NZ(IC+5)                                             ELEL1590
        CALL ELE14                                                      ELEL1600
     &  (IOPT       ,FIRST      ,PRINT      ,PGEOM      ,NAME       ,   ELEL1610
     &   NZ(IZID)   ,NZ(IZIDOF) ,Z(IZCORD) ,Z(IZCOS)   ,NZ(IZJCOS) ,   ELEL1620
     &   Z(IZCNST)  ,Z(IDISP )  ,Z(IZFUB)  ,IREL       ,NZ(IZKGDT+1),   ELEL1630
     &   IELNO      ,RINPUT     ,AXIAL      ,EESE       ,EPSE       ,   ELEL1640
     &   NZ(IC    ) ,NZ(IC+  1) ,NZ(IC+  2) ,NZ(IC+  5) ,NZ(IC+  6) ,   ELEL1650
     &   NZ(IC+ 10) , Z(IC+ 14) ,                                       ELEL1660
     &    Z(IC+ 17) , Z(IC+ 20) , Z(IC+ 23) , Z(IC+ 26) , Z(IC+ 29) ,   ELEL1670
     &    Z(IC+ 32) ,NZ(IC+ 33) ,NZ(IC+ 34) ,NZ(IC+ 58) , Z(IC+ 82) ,   ELEL1680
     &    Z(IC+154) , Z(IC+155) , Z(IC+156) ,DAMAGE     ,DUCFAG     ,   ELEL1690
     &   NZ(IC+159) , Z(IC+160) , Z(IC+161) , Z(IC+167) , Z(IZKE  ) )   ELEL1700
       MSTOR =NZ(IC+5)                                                  ELEL1710
       IELDOF=NZ(IC+1)                                                  ELEL1720
       IZLM=IC+34                                                       ELEL1730
       IZLMKG=IC+58                                                     ELEL1740
       IZKEKG=IC+167+NZ(IC+5) + NZ(IC+33)-1                             ELEL1750
       KGDOF =NZ(IC+159)                                                ELEL1760
      ELSE                                                              ELEL1770
         WRITE(108,10) IELNO                                            ELEL1780
         ERR=.TRUE.                                                     ELEL1790
   10    FORMAT(' ELEMENT ',I6,' IS NOT AVAILABLE' )                    ELEL1800
      ENDIF                                                             ELEL1810
      LSTTYP=IELTYP                                                     ELEL1820
      RETURN                                                            ELEL1830
      END                                                               ELEL1840
C                                                                       WMTR0010
C-----------------------------------------------------------------------WMTR0020
C                                                                       WMTR0030
      SUBROUTINE WMTRX2(A,IR,IC,NAME,IROW)                              WMTR0040
      DIMENSION A(IROW,IC)                                              WMTR0050
      CHARACTER*(*) NAME                                                WMTR0060
C                                                                       WMTR0070
      CALL GETINT(NAME,'DUMP=',IDUMP,0,.TRUE.,.FALSE.)                  WMTR0080
      WRITE(108,*) ' '                                                  WMTR0090
      WRITE(108,*) 'PRINTING MATRIX: ',NAME                             WMTR0100
      DO 50 ICG=1,IC,6                                                  WMTR0110
         IF (ICG.GT.1) WRITE(108,*) ' '                                 WMTR0120
         K=MIN((ICG+5),IC)                                              WMTR0130
         WRITE(108,10) (J,J=ICG,K)                                      WMTR0140
         DO 20 I=1,IR                                                   WMTR0150
           IF (IDUMP.NE.0) THEN                                         WMTR0160
             DO 15 J=ICG,K                                              WMTR0170
   15        IF (A(I,J).NE.0) WRITE(IDUMP,*) I,J,A(I,J),NAME            WMTR0180
             ENDIF                                                      WMTR0190
   20        WRITE(108,30)  I,(A(I,J),J=ICG,K)                         WMTR0200
   50 CONTINUE                                                          WMTR0210
   10 FORMAT(' COLUMN',T17,I5,5(15X,I5))                                WMTR0220
   30 FORMAT(' ROW',I4,6F20.6)                                          WMTR0230
      RETURN                                                            WMTR0240
      END                                                               WMTR0250
C-----------------------------------------------------------------------VCRO0010
C  V3 = V1 X V2,  VECTOR CROSS PRODUCT ...                              VCRO0020
C-----------------------------------------------------------------------VCRO0030
      SUBROUTINE  VCROSS(V3,V1,V2)                                      VCRO0040
      DIMENSION V1(3),V2(3),V3(3)                                       VCRO0050
C                                                                       VCRO0060
      V3(1) = V1(2)*V2(3) - V1(3)*V2(2)                                 VCRO0070
      V3(2) = -(V1(1)*V2(3) - V1(3)*V2(1) )                             VCRO0080
      V3(3) = V1(1)*V2(2) - V1(2)*V2(1)                                 VCRO0090
C                                                                       VCRO0100
      RETURN                                                            VCRO0110
      END                                                               VCRO0120
C                                                                       ELE00010
C-----------------------------------------------------------------------ELE00020
C                                                                       ELE00030
      SUBROUTINE ELE03                                                  ELE00040
     &  (IOPT       ,FIRST      ,PRINT      ,PGEOM      ,NAME       ,   ELE00050
     &   ID         ,IDOF       ,COORD      ,COSINE     ,JTCOS      ,   ELE00060
     &   CONST      ,DISPL      ,VELOC      ,FUB        ,IREL       ,   ELE00070
     &   KGDATA     ,IELNO      ,RINPUT     ,AXIAL      ,ESE        ,   ELE00080
     &   PSE        ,DAMAGE     ,DUCFAG     ,IELTYP     ,IELDOF     ,   ELE00090
     &   ISEB       ,ISE        ,JK         ,MATB       ,R          ,   ELE00100
     &   RT         ,F          ,FT         ,V          ,VT         ,   ELE00110
     &   PKG        ,LXG        ,LM         ,LMXG       ,A          ,   ELE00120
     &   SS         ,SA         ,KGDOF      ,LENGTH     ,           ,   ELE00130
     &   PMAX       ,SE         ,STIFF      )                         ELE00140
C                                                                       ELE00150
      IMPLICIT REAL(A-H,O-Z)                                            ELE00160
C                                                                       ELE00170
      COMMON /RSA/ICOMN,RSAF(12)                                        ELE00180
      REAL ICOMN,LENGTH                                                 ELE00190
      LOGICAL FIRST,PRINT,FALSE,AXIAL                                   ELE00200
      CHARACTER*80 NAME                                                 ELE00210
$INCLUDE:'ZCOMN3'                                                       ELE00220
      DIMENSION IDOF(MAXNOD,6),COORD(MAXNOD,6),COSINE(3,3,NCOS),AC(6,3) ELE00230
      DIMENSION CONST(MAXNOD,6),ID(MAXNOD),JTCOS(MAXNOD),JK(4),LMXG(24) ELE00240
      DIMENSION SAT(3,24),STIFF(600),KGDATA(3),SAT2(6,24),S(6,6),LM(24) ELE00250
      DIMENSION CT(3,3),BS(3,3),VX(3),VY(3),VXB(3),VXT(3),VZT(3),VZB(3) ELE00260
      DIMENSION AT(24,6),ISEB(3),MATB(3),ESEB(3),PSEB(3),DAMB(3)        ELE00270
      DIMENSION SE(ISE),AA(4,6,3),RINPUT(100),DUCT(3,3),EXCR(6,3)       ELE00280
      DIMENSION F(3),R(3),V(3),FT(3),RT(3),VT(3),P(3),VELOC(NDOF)       ELE00290
      DIMENSION DISPL(NDOF,NLOAD),FUB(NDOF),EFUB(24),PMAX(6),A(24,3)    ELE00300
C                                                                       ELE00310
C-----------------------------------------------------------------------ELE00320
C VARIABLES:                                                            ELE00330
C------------ ------- ------------------------------ - ----------------ELE00340
C---------- GLOBAL VARIABLES                                            ELE00350
C IOPT    - 1, INITIALIZE BEAM COLUMN ELEMENT                           ELE00360
C         - 2, CALCULATE GEOMETRIC STIFFNESS                            ELE00370
C         - 3, FORM STIFFNESS                                           ELE00380
C         - 4, CALCULATE INCREMENTAL FORCES                             ELE00390
C         - 5, CALCULATE TOTAL FORCES AND ENERGIES                      ELE00400
C IUNIT   - OUTPUT FILE UNIT # FOR PRINTING OUTPUT                      ELE00410
C FIRST   - FLAG, FIRST=.TRUE., PRINT HEADERS                           ELE00420
C PRINT   - FLAG, PRINT=.TRUE., PRINT DATA                              ELE00430
C NDOF    - # OF GLOBAL DOF                                             ELE00440
C NLOAD   - # OF LOAD COMBINATIONS                                      ELE00450
C MAXNOD  - # ROW DIMENSION OF NODE ARRAY                               ELE00460
C NNODE   - # OF NODES                                                  ELE00470
C ID      - ARRAY OF EXTERNAL NODE NUMBERS                              ELE00480
C IDOF    - ARRAY OF DEGREES OF FREEDOM                                 ELE00490
C COORD   - ARRAY OF COORDINATES                                        ELE00500
C COSINE  - ARRAY OF DIRECTION COSINES OF NODES                         ELE00510
C CONST   - ARRAY OF CONSTRAINT TRANSFORMATIONS                         ELE00520
C DISPL   - GLOBAL DISPLACEMENT MATRIX                                  ELE00530
C Z       - REAL GLOBAL STORTAG VECTOR                                  ELE00540
C NZ      - INTEGER GLOBAL STORTAG VECTOR                               ELE00550
C IZMAT   - LOCATION OF MATERIAL DATA IN GLOBAL STORAGE VECTOR          ELE00560
C NAME    - ELEMENT NAME                                                ELE00570
C IELNO   - ELEMENT NUMBER                                              ELE00580
C RINPUT  - INPUT DATA                                                  ELE00590
C STIFF   - OUTPUT ELEMENT STIFFNESS (UPPER TRIANGULAR FORM)            ELE00600
C LM      - OUTPUT LOCAL TO GLOBAL STIFFNESS MAPPING MATRIX             ELE00610
C---------- ELEMENT VARIABLES                                           ELE00620
C H,Q,AI,AJ,C,BI,BJ,S22,S26,S212,S33,S35,S311,D                         ELE00630
C         - INDIVIDUAL TERMS IN THE ELEMENT STIFFNESS (S) MATRIX        ELE00640
C PKG     - INPUT LOAD FOR FORMING GEOMETRIC STIFFNESS MATRIX...        ELE00650
C R       - INCRIMENTAL DISPLACEMENTS                                   ELE00660
C RT      - TOTAL DISPLACEMENTS                                         ELE00670
C F       - INCRIMENTAL FORCES                                          ELE00680
C FT      - TOTAL FORCES                                                ELE00690
C LM      - LOCAL TO GLOBAL STIFFNESS MAPPING MATRIX                    ELE00700
C CTS     - LOCAL TO GLOBAL ROTATION MATRIX START                       ELE00710
C CTE     - LOCAL TO GLOBAL ROTATION MATRIX END                         ELE00720
C BS      - LOCAL TO GLOBAL ROTATED ECCENTRICITY/CONSTRAINT  MATRIX STARTELE00730
C BE      - LOCAL TO GLOBAL ROTATED ECCENTRICITY/CONSTRAINT  MATRIX END  ELE00740
C STIFF   - ELEMENT STIFFNESS                                           ELE00750
C IELTYP  - ELEMENT TYPE = 10                                           ELE00760
C NODEI   - EXTERNAL JOINT NUMBER, END I                                ELE00770
C NODEJ   - EXTERNAL JOINT NUMBER, END J                                ELE00780
C JOINTI  - INTERNAL JOINT NUMBER, END I                                ELE00790
C JOINTJ  - INTERNAL JOINT NUMBER, END J                                ELE00800
C---------- TEMPORARY VARIABLES                                         ELE00810
C VX      - VECTOR DEFINING THE ELEMENT X AXIS                          ELE00820
C VY      - VECTOR DEFINING THE ELEMENT Y AXIS                          ELE00830
C COSLOC  - LOCAL COSINE MATRIX                                         ELE00840
C A       - GLOBAL TO LOCAL FORCE TRANSFORMATION MATRIX                 ELE00850
C S       - ELEMENT STIFFNESS AT LOCAL COORD, AT ENDS OF FLEXABLE PART  ELE00860
C SAT     - PRODUCT OF S*A'                                             ELE00870
C ASAT    - PRODUCT OF A*SAT                                            ELE00880
C-----------------------------------------------------------------------ELE00890
C                                                                       ELE00900
C                                                                       ELE00910
C------- CHOOSE OPTION                                                  ELE00920
       IF (IOPT.LT.1 .OR.IOPT.GT.14) THEN                               ELE00930
          WRITE(108,*) 'INVALID OPTION IN ELE-02, IOPT=',IOPT           ELE00940
          STOP                                                          ELE00950
       ENDIF                                                            ELE00960
C                                                                       ELE00970
      GO TO (100,200,300,400,500,600,700,800,900,1000,1100,             ELE00980
     &                                        1200,1300,1400) IOPT      ELE00990
C                                                                       ELE01000
  100 CONTINUE                                                          ELE01010
C                                                                       ELE01020
C----- INTERPERTATE INPUT DATA                                          ELE01030
       IELTYP= 3                                                        ELE01040
       MATB(1)= RINPUT( 1)                                              ELE01050
       MATB(2)= RINPUT( 2)                                              ELE01060
       MATB(3)= RINPUT( 3)                                              ELE01070
       NODE1 = RINPUT( 4)                                               ELE01080
       NODE2 = RINPUT( 5)                                               ELE01090
       NODE3 = RINPUT( 6)                                               ELE01100
       NODE4 = RINPUT( 7)                                               ELE01110
       ALPIN = RINPUT( 8)                                               ELE01120
       PXG   = RINPUT( 9)                                               ELE01130
C                                                                       ELE01140
C----- GET INTERNAL NODE # ASSOCIATED WITH EXTERNAL NODE #'S            ELE01150
       JK(1)=IQUICK(NODE1,ID,NNODE)                                     ELE01160
       JK(2)=IQUICK(NODE2,ID,NNODE)                                     ELE01170
       JK(3)=IQUICK(NODE3,ID,NNODE)                                     ELE01180
       JK(4)=IQUICK(NODE4,ID,NNODE)                                     ELE01190
C                                                                       ELE01200
C-----------------------------------------------------------------      ELE01210
C  GET GLOBAL TO LOCAL TRANSFORMATION MATRIX                            ELE01220
C------------------------------------------------------ ------ ------   ELE01230
       DO 105 J=1,3                                                     ELE01240
C----- DEFINE VECTOR X, AT BOTTOM OF THE WALL                           ELE01250
       VXB(J)=COORD(JK(4),J)-COORD(JK(3),J)                             ELE01260
C                                                                       ELE01270
C----- DEFINE VECTOR Y, AT TOP OF THE WALL                              ELE01280
       VXT(J)=COORD(JK(1),J)-COORD(JK(2),J)                             ELE01290
C                                                                       ELE01300
C --- CALCULATE Y VECTOR ...                                           ELE01310
       VY(J)=(COORD(JK(1),J)+COORD(JK(2),J)-                            ELE01320
     &        COORD(JK(3),J)-COORD(JK(4),J))/2.                         ELE01330
  105  CONTINUE                                                         ELE01340
C                                                                       ELE01350
C----- CALCULATE AVERAGE LENGTH,ALPHA AND BETA...                       ELE01360
       LENGTH=SQRT(VY(1)**2+VY(2)**2+VY(3)**2)                          ELE01370
       IF (LENGTH.LE.0) THEN                                            ELE01380
          WRITE(108,101) IELNO,LENGTH                                   ELE01390
  101     FORMAT(1X,20('*'),'ZP','ROR - LENGTH IS LE 0',/               ELE01400
     &          5X,'IELNO=',I5,5X,5X,'LENGTH',1P,G15.6/                 ELE01410
     &          5X,'REVISE INPUT ...  ....  ...  .  ..    '/)           ELE01420
          LENGTH=1.0                                                    ELE01430
       ENDIF                                                            ELE01440
C                                                                       ELE01450
       ALPHA=ALPIN*LENGTH                                               ELE01460
       BETA = LENGTH-ALPHA                                              ELE01470
C                                                                       ELE01480
C----- NORMALIZE Y VECTOR ...                                           ELE01490
       VY(1)=VY(1)/LENGTH                                               ELE01500
       VY(2)=VY(2)/LENGTH                                               ELE01510
       VY(3)=VY(3)/LENGTH                                               ELE01520
C                                                                       ELE01530
C----- CALCULATE THE WIDTH ...                                          ELE01540
       CALL VCROSS(VZT,VY,VXT)                                          ELE01550
       CALL VCROSS(VZB,VY,VXB)                                          ELE01560
       WT = SQRT(VZT(1)**2 + VZT(2)**2 + VZT(3)**2)                     ELE01570
       WB = SQRT(VZB(1)**2 + VZB(2)**2 + VZB(3)**2)                     ELE01580
       W  = (WT+WB)/2.0                                                 ELE01590
C                                                                       ELE01600
C----- CHECK FOR TWIST                                                  ELE01610
       TWIST=ACOS(VCOS(VZT,VZB))*57.2958                                ELE01620
```

```
      IF (ABS(TWIST).GT. 5.0) WRITE(108,102) TWIST                      ELE01630
  102 FORMAT(5X,10('*'),' WARNING THE WALL IS TWISTED-',1P,G12.4,       ELE01640
     &       ' DEGREES, REVISE INPUT AND RERUN')                        ELE01650
C                                                                       ELE01660
C---- CALCULATE VX, AND NORMALIZE                                       ELE01670
      CALL VCROSS(VX,VY,VZB)                                            ELE01680
      WX=SQRT(VX(1)**2+VX(2)**2+VX(3)**2)                               ELE01690
      VX(1)=VX(1)/WX                                                    ELE01700
      VX(2)=VX(2)/WX                                                    ELE01710
      VX(3)=VX(3)/WX                                                    ELE01720
C                                                                       ELE01730
C----- CHECK FOR TWIST                                                  ELE01740
C                                                                       ELE01750
C------ GET LOCAL TO GLOBAL TRANSFORMATION MATRICIES                    ELE01760
C--- JOINT #1 -- #4                                                     ELE01770
      DO 30 L=1,4                                                       ELE01780
         CALL ROTNYZ(VX,VY,COSINE(1,1,JTCOS(JK(L))),CT,                 ELE01790
     &              0 ,0.,0 ,BS,CONST(JK(L),1),MAXNOD)                  ELE01800
      DO 20 I=1,3                                                       ELE01810
      DO 20 J=1,3                                                       ELE01820
        AA(L,I  ,J)=CT(I,J)                                             ELE01830
   20   AA(L,I+3,J)=BS(I,J)                                             ELE01840
30    CONTINUE                                                          ELE01850
C                                                                       ELE01860
C---- ZERO MATRICIES                                                    ELE01870
      DO 25 I=1,3                                                       ELE01880
        FMAX(I  )= 0                                                    ELE01890
        FMAX(I+3)= 0                                                    ELE01900
        F (I)=0                                                         ELE01910
        FT(I)=0                                                         ELE01920
        R (I)=0                                                         ELE01930
        RT(I)=0                                                         ELE01940
        V (I)=0                                                         ELE01950
   25   VT(I)=0                                                         ELE01960
      DO 27 I=1,24                                                      ELE01970
         LM  (I)=0                                                      ELE01980
         LMKG(I)=0                                                      ELE01990
         DO 26 J=1,3                                                    ELE02000
   26      A(I,J)=0.00                                                  ELE02010
         DO 27 K=1,6                                                    ELE02020
   27      AT(I,K)=0.00                                                 ELE02030
C                                                                       ELE02040
C------ ASSEMBLE TRANSFORMATION MATRIX A = A5 X A4 X A3                 ELE02050
C  NOTE. AT IS DIMMENSIONED AS A 24X6 MATRIX, BUT ONLY THE FIRST IELDOF ELE02060
C        ROW'S CONTAIN INFORMATION. THE BALANCE OF THE MATRIX IS         ELE02070
C        UNDEFINED.  THE GLOBAL DOF COORESPONDING TO THE ROW'S OF A      ELE02080
C        ARE STORED IN VECTOR LM   AGAIN, ONLY THE FIRST IELDOF ROW'S    ELE02090
C        OF LM ARE DEFINED.                                             ELE02100
      IELDOF=0                                                          ELE02110
      DO 50 I=1,4                                                       ELE02120
      DO 50 J=1,6                                                       ELE02130
C                                                                       ELE02140
C       REMOVE DOF'S THAT ARE CONSTRAINED TO THE SAME DOF..             ELE02150
        IF ( (IDOF(JK(1),J).EQ.IDOF(JK(4),J))  .OR.                     ELE02160
     &       (IDOF(JK(1),J).EQ.IDOF(JK(3),J))  .OR.                     ELE02170
     &       (IDOF(JK(2),J).EQ.IDOF(JK(4),J))  .OR.                     ELE02180
     &       (IDOF(JK(2),J).EQ.IDOF(JK(3),J)) ) GO TO 50                ELE02190
C                                                                       ELE02200
        IELDOF=IELDOF+1                                                 ELE02210
        LMTI=IDOF(JK(I),J)                                              ELE02220
        K0=I-1                                                          ELE02230
        IF (I.EQ.4) K0=I+2                                              ELE02240
        AT(IELDOF,K0)=AA(I,J,2)                                         ELE02250
        IF (I.EQ.1.OR.I.EQ.4) AT(IELDOF,K0-1)=AA(I,J,1)                 ELE02260
C                                                                       ELE02270
        LM(IELDOF)=LMTI                                                 ELE02280
        IF((AT(IELDOF,1).EQ.0).AND.(AT(IELDOF,2).EQ.0)  .AND.           ELE02290
     &     (AT(IELDOF,3).EQ.0).AND.(AT(IELDOF,4).EQ.0)  .AND.           ELE02300
     &     (AT(IELDOF,5).EQ.0).AND.(AT(IELDOF,6).EQ.0)) THEN            ELE02310
          IELDOF=IELDOF-1                                               ELE02320
        ELSE IF (IELDOF.GT.1) THEN                                      ELE02330
          DO 49  II=1,IELDOF-1                                          ELE02340
            IF (LM(II).EQ.LM(IELDOF)) THEN                              ELE02350
              DO 48 JJ=1,6                                              ELE02360
                AT(II,JJ)=AT(II,JJ)+AT(IELDOF,JJ)                       ELE02370
   48           AT(IELDOF,JJ)=0                                         ELE02380
              IELDOF=IELDOF-1                                           ELE02390
              GO TO 50                                                  ELE02400
            ENDIF                                                       ELE02410
   49     CONTINUE                                                      ELE02420
        ENDIF                                                           ELE02430
C                                                                       ELE02440
   50   CONTINUE                                                        ELE02450
C                                                                       ELE02460
C------ A2A1 FOR STIFFNESS ONLY ..                                      ELE02470
      AC(1,1)=  0.                                                      ELE02480
      AC(2,1)= -1./W                                                    ELE02490
      AC(3,1)=  1./W                                                    ELE02500
      AC(4,1)= -1./W                                                    ELE02510
      AC(5,1)=  0.                                                      ELE02520
      AC(6,1)=  1./W                                                    ELE02530
      AC(1,2)=  1.                                                      ELE02540
      AC(2,2)=  ALPHA/W                                                 ELE02550
      AC(3,2)= -ALPHA/W                                                 ELE02560
      AC(4,2)= -BETA /W                                                 ELE02570
      AC(5,2)= -1.                                                      ELE02580
      AC(6,2)=  BETA /W                                                 ELE02590
      AC(1,3)=  0.                                                      ELE02600
      AC(2,3)=  1./2.                                                   ELE02610
      AC(3,3)=  1./2                                                    ELE02620
      AC(4,3)= -1./2                                                    ELE02630
      AC(5,3)=  0.                                                      ELE02640
      AC(6,3)= -1./2.                                                   ELE02650
      DO 55 I=1,IELDOF                                                  ELE02660
      DO 55 J=1,3                                                       ELE02670
        A(I,J)=0.                                                       ELE02680
        DO 55 K=1,6                                                     ELE02690
   55     A(I,J)=A(I,J)+AT(I,K)*AC(K,J)                                ELE02700
C                                                                       ELE02710
C                                                                       ELE02720
C------ GET SPRING STIFFNESS                                            ELE02730
      FALSE=.FALSE.                                                     ELE02740
      LSTTYP=0                                                          ELE02750
      DO 56 I=1,3                                                       ELE02760
        CALL MATLIB(2,LSTTYP,FALSE,ESEB(I),PSEB(I),DUCT(1,I),EXCR(1,I),ELE02770
     &              FT,F,RT,S,0 ,0.,LELEM,LMAT,MATB(1),MATTYP,          ELE02780
     &              SE(ISEB(I)),IELNO,DAMAGE)                           ELE02790
56    CONTINUE                                                          ELE02800
C                                                                       ELE02810
      ESE=0.                                                            ELE02820
      PSE=0                                                             ELE02830
C                                                                       ELE02840
C------- SET UP THE LOCAL STIFFNESS MATRIX                              ELE02850
      SB= SE(ISEB(1))                                                   ELE02860
      SS= SE(ISEB(2))                                                   ELE02870
      SA= SE(ISEB(3))                                                   ELE02880
C                                                                       ELE02890
C------ CALCULATE SAT                                                   ELE02900
      SBL=SB/LENGTH                                                     ELE02910
      SSL=SS/LENGTH                                                     ELE02920
      SAL=SA/LENGTH                                                     ELE02930
      DO 60 J=1,IELDOF                                                  ELE02940
        SAT(1,J)=SBL*A(J,1)                                             ELE02950
        SAT(2,J)=SSL*A(J,2)                                             ELE02960
   60   SAT(3,J)=SAL*A(J,3)                                             ELE02970
C                                                                       ELE02980
C ---- CALCULATE ASAT AND CONVERT TO HALF STORAGE MODE  BY COLUMNS       ELE02990
C                                                                       ELE03000
C          1  3  6 10 15 21 28 36 45 55 66 78    300   NUMBER INDICATES ELE03010
C             2  5  9 14 20 27 35 44 54 65 77          LOCATION OF DATA  ELE03020
C                4  8 13 19 26 34 43 53 64 76          IN ARRAY STIFF    ELE03030
C                   7 12 18 25 33 42 52 63 75                            ELE03040
C                  11 17 24 32 41 51 62 74                               ELE03050
C                     16 23 31 40 50 61 73                               ELE03060
C                        22 30 39 49 60 72                               ELE03070
C                           29 38 48 59 71                               ELE03080
C                                 37 47 58 70                           ELE03090
C                                    46 57 69                           ELE03100
C                                       56 68                           ELE03110
C                                          67                           ELE03120
C                                                                       ELE03130
C                                                                       ELE03140
C                                                                       ELE03150
      L=0                                                               ELE03160
      DO 70 J=1,IELDOF                                                  ELE03170
        DO 70 I=J,1,-1                                                  ELE03180
        L=L+1                                                           ELE03190
        STIFF(L)=0                                                      ELE03200
        DO 70 K=1,3                                                     ELE03210
   70     STIFF(L)=STIFF(L)+A(I,K)*SAT(K,J)                            ELE03220
C                                                                       ELE03230
C-----------------------------------------------------------------      ELE03240
C  ASSEMBLE GEOMETRIC STIFFNESS MATRIX FOR A UNIT LOAD.                 ELE03250
C-----------------------------------------------------------------      ELE03260
      LKG=L+1                                                           ELE03270
      IF ( PKG.EQ.0 .AND. KGDATA(1).NE.2 ) GO TO 180                    ELE03280
      IF (KGDATA(1).EQ.0 .OR. KGDATA(2).EQ 0 ) GO TO 180                ELE03290
      AXIAL=.TRUE.                                                      ELE03300
C                                                                       ELE03310
C---------- DETERMINE THE AXIAL LOAD TO BE USED FOR KG. .               ELE03320
        IF (KGDATA(1).EQ.1) THEN                                        ELE03330
              PGEOM =  PKG                                              ELE03340
        ELSE IF (KGDATA(1).EQ.2) THEN                                   ELE03350
              PGEOM = -F(3)                                             ELE03360
        ENDIF                                                           ELE03370
C                                                                       ELE03380
C----- ASSEMBLE TRANSFORMATION MATRIX A                                 ELE03390
C  NOTE: AT IS DIMMENSIGNED AS A 12X6 MATRIX, BUT ONLY THE FIRST IELDOF ELE03400
C        ROW'S CONTAIN INFORMATION. THE BALANCE OF THE MATRIX IS         ELE03410
C        UNDEFINED.  THE GLOBAL DOF COORESPONDING TO THE ROW'S OF A      ELE03420
C        ARE STORED IN VECTOR LM.  AGAIN, ONLY THE FIRST IELDOF ROW'S    ELE03430
C        OF LM ARE DEFINED.                                             ELE03440
      DO 140 I=1,24                                                     ELE03450
      DO 140 K=1,6                                                      ELE03460
  140   AT(I,K)=0.00                                                    ELE03470
      KGDOF=0                                                           ELE03480
      DO 145 I=1,4                                                      ELE03490
      DO 145 J=1,6                                                      ELE03500
        KGDOF =KGDOF +1                                                 ELE03510
        LMTI=IDOF(JK(I),J)                                              ELE03520
        K0=I-1                                                          ELE03530
        IF (I.EQ.4) K0=I+2                                              ELE03540
        AT(KGDOF,K0)=AA(I,J,3)                                          ELE03550
        IF (I.EQ.1.OR.I.EQ.4) AT(KGDOF,K0-1)=AA(I,J,1)                  ELE03560
        LMKG(KGDOF)=LMTI                                                ELE03570
        IF((AT(KGDOF,1).EQ.0).AND.(AT(KGDOF,2).EQ.0).AND.               ELE03580
     &       (AT(KGDOF,3).EQ.0).AND.(AT(KGDOF,4).EQ.0).AND.             ELE03590
     &       (AT(KGDOF,5).EQ.0).AND.(AT(KGDOF,6).EQ.0)) THEN            ELE03600
          KGDOF=KGDOF-1                                                 ELE03610
        ELSE IF (KGDOF.GT.1) THEN                                       ELE03620
          DO 149  II=1,KGDOF-1                                          ELE03630
            IF (LMKG(II).EQ.LMKG(KGDOF)) THEN                           ELE03640
              DO 148 JJ=1,6                                             ELE03650
                AT(II,JJ)=AT(II,JJ)+AT(KGDOF,JJ)                       ELE03660
  148           AT(KGDOF,JJ)=0                                          ELE03670
              KGDOF=KGDOF-1                                             ELE03680
              GO TO 145                                                 ELE03690
            ENDIF                                                       ELE03700
  149     CONTINUE                                                      ELE03710
        ENDIF                                                           ELE03720
  145   CONTINUE                                                        ELE03730
C                                                                       ELE03740
C------ LOCAL GEOMETRIC STIFFNESS MATRIX                                ELE03750
      DO 150 I2=1,12                                                    ELE03760
      DO 150 JI=1,12                                                    ELE03770
  150   S(I2,JI)=0.0                                                    ELE03780
        S(1,1)=  1.0/LENGTH                                             ELE03790
        S(2,2)=  0.5/LENGTH                                             ELE03800
        S(3,3)=  0.5/LENGTH                                             ELE03810
        S(4,4)=  0.5/LENGTH                                             ELE03820
        S(5,5)=  1.0/LENGTH                                             ELE03830
        S(6,6)=  0.5/LENGTH                                             ELE03840
        S(1,5)= -1.0/LENGTH                                             ELE03850
        S(2,6)= -0.5/LENGTH                                             ELE03860
        S(3,4)= -0.5/LENGTH                                             ELE03870
        S(5,1)= -1.0/LENGTH                                             ELE03880
        S(6,2)= -0.5/LENGTH                                             ELE03890
        S(4,3)= -0.5/LENGTH                                             ELE03900
C                                                                       ELE03910
C------ CALCULATE SAT                                                   ELE03920
      DO 160 I=1,6                                                      ELE03930
        DO 160 J=1,KGDOF                                                ELE03940
        SAT2(I,J)=0                                                     ELE03950
        DO 160 K=1,6                                                    ELE03960
  160     SAT2(I,J)=SAT2(I,J)+S(I,K)*AT(J,K)                            ELE03970
C                                                                       ELE03980
C------ CALCULATE ASAT AND CONVERT TO HALF STORAGE MODE  BY COLUMNS     ELE03990
      DO 170 J=1,KGDOF                                                  ELE04000
        DO 170 I=J,1,-1                                                 ELE04010
        L=L+1                                                           ELE04020
        STIFF(L)=0                                                      ELE04030
        DO 170 K=1,6                                                    ELE04040
  170     STIFF(L)=STIFF(L)+AT(I,K)*SAT2(K,J)                           ELE04050
C                                                                       ELE04060
C------ PRINT GEOMETRIC STIFF DATA FOR DEB_UGGING                       ELE04070
C                                                                       ELE04080
C------ RELEASE UNUSED STORAGE                                          ELE04090
  180 IREL=600-L                                                        ELE04100
C                                                                       ELE04110
C     PRINT COLUMN DATA                                                 ELE04120
      IF (FIRST) WRITE(108,192)                                         ELE04130
  191 WRITE(108,193) NAME,IELNO,(MATB(I),I=1,3),(ID(JK(I)),I=1,6),      ELE04140
     &                LENGTH,W,ALPIN,PKG                                ELE04150
C                                                                       ELE04160
  192 FORMAT(//' ELEMENT 03, R/C SHEAR WALL ELEMENT'// T17,             ELE04170
     &    ' ELEM',' BEND',' SHEAR',' AXIAL',' JOINT',' JOINT',          ELE04180
     &    ' JOINT',' JOINT',    LENGTH   ',    WIDTH   ',              ELE04190
     &    '    ALPHA   ',    PKG    '/,T17,                            ELE04200
     &    '     #',' MATL',' MATL',' MATL',    #1',     #2',           ELE04210
     &    '     #3',     #4'/)                                         ELE04220
C                                                                       ELE04230
  193 FORMAT(1X,A15,8I6,2X,1P,6G12.4)                                   ELE04240
      IOPT=IOPT                                                         ELE04250
      RETURN                                                            ELE04260
C------------------------- DETERMINE GEOMETRIC STIFF ------------------ELE04270
  200 CONTINUE                                                          ELE04280
C- KG IS DETERMINED FOR A UNIT LOAD WITH IOPT=1,                        ELE04290
C- THE ACTUAL LOAD (PGEOM) IS DETERMINED HERE.                          ELE04300
C- KG IS MULTIPLIED BY PGEOM WHEN THE TOTAL STIFFNESS IS ASSEMBLED      ELE04310
C- NEGATIVE PGEOM IS COMPRESSION...                                     ELE04320
C                                                                       ELE04330
      AXIAL=.TRUE.                                                      ELE04340
C------ --- DETERMINE THE AXIAL LOAD TO BE USED FOR KG ..               ELE04350
      IF (KGDATA(1) EQ.1) THEN                                          ELE04360
              PGEOM =  PKG                                              ELE04370
      ELSE IF (KGDATA(1).EQ.2) THEN                                     ELE04380
              PGEOM = -F(3)                                             ELE04390
      ENDIF                                                             ELE04400
C                                                                       ELE04410
      RETURN                                                            ELE04420
C                                                                       ELE04430
C------------------------- DETERMINE STIFFNESS ------------------------ELE04440
  300 CONTINUE                                                          ELE04450
C                                                                       ELE04460
C------ SB, SS AND SA CONTAIN THE STIFFNESSES, WHICH WERE DETERMINED IN ELE04470
C       THE LAST CALL TO MA_TLIB                                        ELE04480
C       IF SB=SE(ISEB) AND SS=SE(ISES) AND SA=SE(ISEA)                  ELE04490
C       THEN THE STIFFNESS HAS NOT CHANGED, RETURN                      ELE04500
C                                                                       ELE04510
      IF (SB.EQ.SE(ISEB(1)).AND.SS.EQ.SE(ISEB(2)).AND.SA                ELE04520
     &    .EQ.SE(ISEB(3))) RETURN                                       ELE04530
C                                                                       ELE04540
```

```
C----- - SET UP THE LOCAL STIFFNESS MATRIX                                ELE04550
      SB= SE(ISEB(1))                                                     ELE04560
      SS= SE(ISEB(2))                                                     ELE04570
      SA= SE(ISEB(3))                                                     ELE04580
      SBL= SE(ISEB(1))/LENGTH                                             ELE04590
      SSL= SE(ISEB(2))/LENGTH                                             ELE04600
      SAL= SE(ISEB(3))/LENGTH                                             ELE04610
C                                                                         ELE04620
C----- CALCULATE SAT                                                      ELE04630
      DO 360 J=1,IELDOF                                                   ELE04640
        SAT(1,J)=SBL*A(J,1)                                               ELE04650
        SAT(2,J)=SSL*A(J,2)                                               ELE04660
  360   SAT(3,J)=SAL*A(J,3)                                               ELE04670
C                                                                         ELE04680
C----- CALCULATE ASAT, STORE IN UPPER TRIANGULAR MATRIX                   ELE04690
      L=0                                                                 ELE04700
      DO 370 J=1,IELDOF                                                   ELE04710
        DO 370 I=J,1,-1                                                   ELE04720
        L=L+1                                                             ELE04730
        STIFF(L)=0                                                        ELE04740
        DO 370 K=1,3                                                      ELE04750
  370     STIFF(L)=STIFF(L)+A(I,K)*SAT(K,J)                               ELE04760
C                                                                         ELE04770
      RETURN                                                              ELE04780
C------------------------ DETERMINE INCRIMENTAL FORCES ----------------- ELE04790
  400 CONTINUE                                                            ELE04800
      IF (PRINT.AND.FIRST) WRITE(108,691) STEPID                          ELE04810
C                                                                         ELE04820
C------- SET UP THE COMPONENT STIFFNESSES                                 ELE04830
      SB= SE(ISEB(1))                                                     ELE04840
      SS= SE(ISEB(2))                                                     ELE04850
      SA= SE(ISEB(3))                                                     ELE04860
C                                                                         ELE04870
C---- - LOOP FOR EACH LOAD                                               ELE04880
      DO 630 LOAD=1,NLOAD                                                 ELE04890
C                                                                         ELE04900
C------ TRANSFORM FRAME DISPL INTO LOCAL DISPL                            ELE04910
      DO 410 I=1,3                                                        ELE04920
        R(I)=0                                                            ELE04930
        V(I)=0                                                            ELE04940
        DO 410 J=1,IELDOF                                                 ELE04950
          K=LM(J)                                                         ELE04960
          R(I)=R(I)+A(J,I)*DISPL(K,LOAD)                                  ELE04970
  410     V(I)=V(I)+A(J,I)*VELOC(K)                                       ELE04980
C                                                                         ELE04990
C------ INCRIMENTAL FORCES                                                ELE05000
     .F(1)=SB*R(1)/LENGTH                                                 ELE05010
      F(2)=SS*R(2)/LENGTH                                                 ELE05020
      F(3)=SA*R(3)/LENGTH                                                 ELE05030
C                                                                         ELE05040
C------ CHECK STIFFNESS AND CALCULATE UNBALANCED FORCES                   ELE05050
C       THIS IS PREFORMED ONLY FOR INELASTIC ANALYSIS                     ELE05060
      P(1) =FT(1)+F(1)                                                    ELE05070
      P(2) =FT(2)+F(2)                                                    ELE05080
      P(3) =FT(3)+F(3)                                                    ELE05090
      IF (.NOT. ELSTIC) THEN                                              ELE05100
C  - TRANSFER PERMANENT HYSTERESIS DATA TO TEMPROARY STORAGE...           ELE05110
        ISE2=ISE/2                                                        ELE05120
        DO 415 I=1,ISE2                                                   ELE05130
  415     SE(I+ISE2)=SE(I)                                                ELE05140
        DO 420 I=1,3                                                      ELE05150
C  -- BENDING ----------------                                           ELE05160
C  -- SHEAR ----------------                                             ELE05170
C  - AXIAL ----------------                                             ELE05180
C  -- SET UP TEMPORARY - TOTAL LOADS, DISPL'S AND VELOC                  ELE05190
        PL=FT(I)                                                          ELE05200
        D =(RT(I)+R(I))/LENGTH                                            ELE05210
        DL=RT(I) /LENGTH                                                  ELE05220
        VC=VT(I)+V(I)                                                     ELE05230
        VL=VT(I)                                                          ELE05240
C       -- CALL HYST MODEL TO GET NEW STIFFNESS AND                       ELE05250
C          THE TOTAL FORCE P AT DISPL D.                                  ELE05260
        CALL MATLIB(3,LSTTYP,FALSE,ESEB(I),PSEB(I),DUCT(1,I),EXCR(1,I), ELE05270
     &              P(I),PL,D,DL,VC,VL,LELEM,LMAT,MATB(I),MATTYP,       ELE05280
     &              SE(ISE2+ISEB(I)),IELNO,DAMAGE)                      ELE05290
        IF (SE(ISE2+ISEB(I)).NE.SE(ISEB(I)) NEWK=.TRUE.                   ELE05300
  420   CONTINUE                                                          ELE05310
      ENDIF                                                               ELE05320
C                                                                         ELE05330
      IF (ELSTIC) THEN                                                    ELE05340
        ESE=0.                                                            ELE05350
        DO 425 I=1,3                                                      ELE05360
          ESEB(I)=0.50*(FT(I)+F(I))*(RT(I)+R(I))                          ELE05370
  425     ESE=ESE+ESEB(I)                                                 ELE05380
          PSE=0 0                                                         ELE05390
      ELSE                                                                ELE05400
        ESE=(ESEB(1)+ESEB(2)+ESEB(3))*LENGTH                              ELE05410
        PSE=(PSEB(1)+PSEB(2)+PSEB(3))*LENGTH                              ELE05420
      ENDIF                                                               ELE05430
C                                                                         ELE05440
C------ SAVE MAXIMUM LOADS                                                ELE05450
      IF (NLOAD.GT.1) THEN                                                ELE05460
        DO 389 I=1,3                                                      ELE05470
          IF (ABS(F(I)).GT.ABS(FMAX(I)) ) THEN                            ELE05480
            FMAX(I  )= F(I)                                               ELE05490
            FMAX(I+3)= R(I)                                               ELE05500
          ENDIF                                                           ELE05510
  389   CONTINUE                                                          ELE05520
      ENDIF                                                               ELE05530
C                                                                         ELE05540
C----- PRINT DATA                                                         ELE05550
      IF (PRINT) WRITE(108,392) IELNO,LOAD,ID(JK(1)),ID(JK(2)),           ELE05560
     &                          ID(JK(3)),ID(JK(4)),(F(I),R(I),I=1,3)     ELE05570
  392 FORMAT (I10,I5,2X,4I5,1P,6G15 6)                                    ELE05580
C                                                                         ELE05590
C------ CALCULATE THE UNBALANCED MEMBER FORCE ON A JOINT.                 ELE05600
      IF (LOAD.EQ.1) THEN                                                 ELE05610
            UBM = -( F(1)*FT(1)-P(1) )                                    ELE05620
            UBS = -( F(2)*FT(2)-P(2) )                                    ELE05630
            UBA = -( F(3)*FT(3)-P(3) )                                    ELE05640
            DO 440 I=1,IELDOF                                             ELE05650
              K=LM(I)                                                     ELE05660
              EFUB(I)= A(I,1)*UBM + A(I,2)*UBS + A(I,3)*UBA               ELE05670
  440         FUB(K)= FUB(K) + EFUB(I)                                    ELE05680
      ENDIF                                                               ELE05690
      F(1)=P(1)-FT(1)                                                     ELE05700
      F(2)=P(2)-FT(2)                                                     ELE05710
      F(3)=P(3)-FT(3)                                                     ELE05720
  430 CONTINUE                                                            ELE05730
C                                                                         ELE05740
C------ THIS IS FOR SOLOSA ONLY                                           ELE05750
      IF( ICOMN .EQ. 1  OR. ICOMN .EQ  2 ) THEN                           ELE05760
        DO 431 I=1,3                                                      ELE05770
  431   RSAP(I)=F(I)                                                      ELE05780
      ENDIF                                                               ELE05790
C                                                                         ELE05800
  491 FORMAT (/'    R/C SHEAR WALL FORCES... ',5X,A/                      ELE05810
     &   '   ELEMENT LOAD',T17,' JNT-1',' JNT-2',' JNT-3',' JNT-4',       ELE05820
     &   '    MOMENT      ','    ROTATION     ','     SHEAR      ',      ELE05830
     & ' SHEAR DISPL   ','     AXIAL     ',' AXIAL DISPL   '/)           ELE05840
C                                                                         ELE05850
      IOPT=IOPT                                                           ELE05860
      RETURN                                                              ELE05870
C------------------------ DETERMINE TOTAL FORCES, ENERGIES ECT --------- ELE05880
  500 CONTINUE                                                            ELE05890
      IF( ICOMN .NE. 1 .AND  ICOMN .NE. 2 ) THEN                          ELE05900
        IF (PRINT.AND.FIRST) WRITE(108,491) STEPID                        ELE05910
C                                                                         ELE05920
C------ TOTAL FORCES, DISPLACEMENTS AND VELOCITIES                        ELE05930
       DO 510 I=1,3                                                       ELE05940
        FT(I) = FT(I) + F(I)                                              ELE05950
        RT(I) = RT(I) + R(I)                                              ELE05960
        VT(I) = VT(I) + V(I)                                              ELE05970
        IF ( ABS(FT(I)).GT.ABS(FMAX(I)) ) THEN                            ELE05980
         FMAX(I  )=FT(I)                                                  ELE05990
         FMAX(I+3)=RT(I)                                                  ELE06000
        ENDIF                                                             ELE06010
  510   CONTINUE                                                          ELE06020
C                                                                         ELE06030
C------ PRINT DATA                                                        ELE06040
      IF (PRINT) WRITE(108,492) IELNO,LOAD,ID(JK(1)),ID(JK(2)),           ELE06050
     &                          ID(JK(3)),ID(JK(4)),(FT(I),RT(I),I=1,3)   ELE06060
  492 FORMAT (I10,I5,2X,4I5,1P,6G15.6)                                    ELE06070
      ELSE                                                                ELE06080
       IF( PRINT .AND. FIRST) WRITE(108,497) STEPID                       ELE06090
       DO 513 I=1,3                                                       ELE06100
  513   FT(I)=RSAP(I)                                                     ELE06110
        IF (PRINT) WRITE(108,492) IELNO,LOAD,ID(JK(1)),ID(JK(2)),         ELE06120
     &                            ID(JK(3)),ID(JK(4)),(FT(I),I=1,3)       ELE06130
      ENDIF                                                               ELE06140
  497 FORMAT (/'    R/C SHEAR WALL FORCES...',5X,A/                       ELE06150
     &   '   ELEMENT LOAD',T17,' JNT-1',' JNT-2',' JNT-3',' JNT-4',       ELE06160
     &   '    MOMENT      ','    SHEAR      ','     AXIAL      '/)       ELE06170
C                                                                         ELE06180
C------ TRANSFER TEMPROARY HYSTERESIS DATA TO PERMANENT STORAGE. .        ELE06190
      ISE2=ISE/2                                                          ELE06200
      IF (.NOT.ELSTIC) THEN                                               ELE06210
        DO 520 I=1,ISE2                                                   ELE06220
  520     SE(I)=SE(I+ISE2)                                                ELE06230
      ENDIF                                                               ELE06240
C                                                                         ELE06250
      RETURN                                                              ELE06260
C                                                                         ELE06270
C------------------------ DETERMINE FIXED END FORCES, AND ADD TO FORCE- ELE06280
  600 RETURN                                                              ELE06290
C                                                                         ELE06300
C------- FIXED END FORCES ARE NOT AVAILABLE FOR R/C SHEAR WALL            ELE06310
C                                                                         ELE06320
C------------------- WRITE MEMBER FORCES TO OUTPUT FILE ---------------- ELE06330
  700 CONTINUE                                                            ELE06340
C                                                                         ELE06350
C. ..... BENDING ENERGY, DUCTILITY AND EXCURSION .....                    ELE06360
      IF (ELSTIC) THEN                                                    ELE06370
       ESE=0                                                              ELE06380
       DO 704 I=1,3                                                       ELE06390
        ESEB(I)=0.50*(FT(I)+F(I))*(RT(I)+R(I))                            ELE06400
704     ESE=ESE+ESEB(I)                                                   ELE06410
        PSE=0.0                                                           ELE06420
      ELSE                                                                ELE06430
      DO 705 I=1,3                                                        ELE06440
       CALL MATLIB(4,LSTTYP,FALSE,ESEB(I),PSEB(I),DUCT(1,I),EXCR(1,I),    ELE06450
     &             FT,F,RT,R,0.,0. ,LELEM,LMAT,MATB(I),MATTYP,           ELE06460
     &             SE(ISEB(I)),IELNO,DAMAGE)                             ELE06470
705   CONTINUE                                                            ELE06480
C.......... SHEAR ENERGY, DUCTILITY AND EXCURSION .....                   ELE06490
C.......... AXIAL ENERGY, DUCTILITY AND EXCURSION .....                   ELE06500
C                                                                         ELE06510
        ESE=(ESEB(1)+ESEB(2)+ESEB(3))*LENGTH                              ELE06520
        PSE=(PSEB(1)+PSEB(2)+PSEB(3))*LENGTH                              ELE06530
      ENDIF                                                               ELE06540
C                                                                         ELE06550
                                                                          ELE06560
      IF (PRINT) WRITE (IUNIT,710) (ID(JK(I)),I=1,4)                      ELE06570
      WRITE (IUNIT,711) (FT(I),RT(I),I=1,3),ESE,PSE                       ELE06580
  710 FORMAT (4I5,1P,6E10.3/(8E10.3) )                                    ELE06590
  711 FORMAT (1P,8E10.3)                                                  ELE06600
C                                                                         ELE06610
      RETURN                                                              ELE06620
C                                                                         ELE06630
C------------------- READ AND PRINT MEMBER FORCES FROM OUTPUT FILE ----- ELE06640
  800 CONTINUE                                                            ELE06650
      SX =0                                                               ELE06660
      SXX=0                                                               ELE06670
      SY =0                                                               ELE06680
      SXY=0                                                               ELE06690
      SN =0                                                               ELE06700
      BACKSPACE(IUNIT)                                                    ELE06710
      READ (IUNIT,*  ) ITYPE,IELNO,IWRITE,TO,DT                          ELE06720
      READ (IUNIT,710) NODE1,NODE2,NODE3,NODE4                            ELE06730
      ISTEP=-IWRITE                                                       ELE06740
      AVEMV=0                                                             ELE06750
      ICOUNT=0                                                            ELE06760
C                                                                         ELE06770
  810 READ (IUNIT,815,END=830) (F(I),R(I),I=1,3),ESE,PSE                  ELE06780
  815 FORMAT (8E10.3)                                                     ELE06790
        ISTEP=ISTEP-IWRITE                                                ELE06800
        T=TO+DT*ISTEP                                                     ELE06810
        IF (ISTEP.EQ.0) WRITE(108,805) NODE1,NODE2,NODE3,NODE4            ELE06820
        IF (MOD(ISTEP,INC).EQ.0)                                          ELE06830
     &  WRITE(108,820) ISTEP,T,(F(I),R(I),I=1,3),ESE,PSE                  ELE06840
        SX  = SX  + F(2)                                                  ELE06850
        SXX = SXX + F(2)*F(2)                                             ELE06860
        SY  = SY  + F(1)                                                  ELE06870
        SXY = SXY + F(1)*F(2)                                             ELE06880
        SN  = SN  + 1                                                     ELE06890
        IF (F(2).NE.0) THEN                                               ELE06900
             RATMV= F(1)/F(2)                                             ELE06910
             AVEMV=AVEMV+RATMV                                            ELE06920
             ICOUNT=ICOUNT+1                                              ELE06930
        ENDIF                                                             ELE06940
        GO TO 810                                                         ELE06950
C                                                                         ELE06960
  830 CONTINUE                                                            ELE06970
      AVEMV=AVEMV/ MAX(ICOUNT,1)                                          ELE06980
      WRITE(108,831) AVEMV,ICOUNT                                         ELE06990
      IF (SY.NE 0 ) THEN                                                  ELE07000
        RM=SY/SX                                                          ELE07010
        ICOUNT=SN                                                         ELE07020
        WRITE(108,833) RM,ICOUNT                                          ELE07030
      ENDIF                                                               ELE07040
      DM = SN*SXX - SX*SX                                                 ELE07050
      DB = SN*SX  - SXX*SN                                                ELE07060
      IF (DM.NE.0 .AND  DB.NE 0) THEN                                     ELE07070
        RB = ( SXY*SX  - SY*SXX ) / DB                                    ELE07080
        RM = ( SN*SXY - SX*SY  ) / DM                                     ELE07090
        ICOUNT=SN                                                         ELE07100
        WRITE(108,832) RM,RB,ICOUNT                                       ELE07110
      ENDIF                                                               ELE07120
  831 FORMAT(/T10,'AVERAGE MOMENT/SHEAR RATIO=',1P,G12.4,5X,0P,I6,        ELE07130
     & ' POINTS WERE USED, ',' POINTS WITH V=0 ARE EXCLUDED')             ELE07140
  832 FORMAT( T10,'REGRESSION MOMENT-SHEAR EQUN:   ',                     ELE07150
     & ' M =',1P,G12.4,  '* V +', G12.4,5X,0P,I6,' POINTS WERE USED')     ELE07160
  833 FORMAT( T10,'REGRESSION MOMENT-SHEAR EQUN:   ',                     ELE07170
     & ' M =', 1P,G12.4,  '* V  ', 12X  ,5X,0P,I6,' POINTS WERE USED')    ELE07180
  805 FORMAT (/'    R/C SHEAR WALL FORCES...',/                           ELE07190
     & 5X,'NODE1:',I6,5X,'NODE2:',I6,5X,'NODE3:',I6,5X,'NODE4:',I6,//     ELE07200
     & '  STEP      TIME     ','     MOMENT     ','    ROTATION    ',    ELE07210
     & '       SHEAR     ','  SHEAR DISPL.  ',                           ELE07220
     & '     AXIAL     ','  AXIAL DISPL.  ',                             ELE07230
     & '   ESE       ','     PSE      '/)                                ELE07240
  820 FORMAT (I6,1P,G12.4,6G14.5,2G12.4)                                  ELE07250
      RETURN                                                              ELE07260
C                                                                         ELE07270
C------------------- DETERMINE THE LENGTH OF THE MATERIAL DATA           ELE07280
  900 CONTINUE                                                            ELE07290
      ISE=0                                                               ELE07300
C-- ---- LOOP FOR EACH MATERIAL ---- --                                   ELE07310
      DO 910 I=1,3                                                        ELE07320
        ISEB(I)=ISE+1                                                     ELE07330
        MAT1 = RINPUT(I)                                                  ELE07340
        CALL MATLIB(10,LSTTYP,FALSE,ESE,PSE,DUCT(1,2),EXCR(1,2),          ELE07350
     &   FT,F,RT,R,0 ,0 ,LELEM,LMAT,MAT1,MATTYP,SE,IELNO,DAMAGE)          ELE07360
  910   ISE=ISE + LELEM                                                   ELE07370
C                                                                         ELE07380
C------- DOUBLE THE STORAGE, TO ALLOW FOR TEMPROARY VALUES .              ELE07390
        ISE=ISE*2                                                         ELE07400
C                                                                         ELE07410
      RETURN                                                              ELE07420
C                                                                         ELE07430
C------------------- DETERMINE THE STRAIN ENERGY                         ELE07440
 1000 CONTINUE                                                            ELE07450
C-------------------- DUCTILITIEIS AND EXCURSION RATIOS                  ELE07460
```

```
 1100 CONTINUE                                                          ELE07470
C       ... NOTE ELE03 WAS LAST CALLED TO CALC INCIRMENTAL DISPL,       ELE07480
C               THE ENERGIES ASSOCIATED WITH INCR DISPL ARE STORED IN   ELE07490
C               THE ADDRESSES- ISE8+ISE/2                               ELE07500
C                              ISE8+ISE/2                               ELE07510
C                              ISEA+ISE/2                               ELE07520
C..... ..... BENDING ENERGY, DUCTILITY AND EXCURSION .....             ELE07530
      ISE2=ISE/2                                                        ELE07540
      DO 1105 I=1,3                                                     ELE07550
        CALL MATLIB(4,LSTTYP,FALSE,ESEB(I),PSEB(I),DUCT(1,I),EXCR(1,I), ELE07560
     &              FT,F,RT,R,0.,0.,LELEM,LMAT,MATB(I),MATTYP,          ELE07570
     &              SE(ISEB(I)+ISE2),IELNO,DAMAGE)                      ELE07580
 1105 CONTINUE                                                          ELE07590
C. . ..... SHEAR ENERGY, DUCTILITY AND EXCURSION ... .                  ELE07600
C .  ... .. AXIAL ENERGY, DUCTILITY AND EXCURSION .....                 ELE07610
C                                                                       ELE07620
        IF ( DUCPAG .EQ. 0 .AND. ( ABS(DUCT(1,1)) .GE. 1  OR.           ELE07630
     &       ABS(DUCT(1,2)) .GE. 1 .OR. ABS(DUCT(1,3)) .GE. 1 )) THEN   ELE07640
             DUCPAG=ABS(DUCT(1,1))                                      ELE07650
             IF( DUCPAG .LT. ABS(DUCT(1,2)) ) DUCPAG=ABS(DUCT(1,2))     ELE07660
             IF( DUCPAG .LT. ABS(DUCT(1,3)) ) DUCPAG=ABS(DUCT(1,3))     ELE07670
        ENDIF                                                           ELE07680
C                                                                       ELE07690
      IF (ELSTIC) THEN                                                  ELE07700
       ESE=0.                                                           ELE07710
       DO 1106 I=1,3                                                    ELE07720
        ESEB(I)=0.50*(FT(I)+F(I))*(RT(I)+R(I))                          ELE07730
 1106   ESE=ESE+ESEB(I)                                                 ELE07740
        PSE=0.0                                                         ELE07750
      ELSE                                                              ELE07760
        ESE=(ESEB(1)+ESEB(2)+ESEB(3))*LENGTH                            ELE07770
        PSE=(PSEB(1)+PSEB(2)+PSEB(3))*LENGTH                            ELE07780
      ENDIF                                                             ELE07790
C                                                                       ELE07800
      IF (PRINT.AND.FIRST) WRITE(108,1191)                              ELE07810
      IF (PRINT) WRITE(108,1192) IELNO,((DUCT(I,J),EXCR(I,J),I=1,3),    ELE07820
     $                                  J=1,3)                          ELE07830
 1192 FORMAT (1X,I5,18F7.2)                                             ELE07840
 1191 FORMAT (/'    R/C SHEAR WALL DUCTILITY AND EXCURSION RATIOS'/6X,   ELE07850
     &  ' |-------- BENDING COMPONENT ------------|',                    ELE07860
     &  ' |-- --- -- SHEAR COMPONENT ------------|',                    ELE07870
     &  ' |---------- AXIAL COMPONENT ------------|'/,' ELEM#',         ELE07880
     &  ' |-  DEFN 1--- --- DEFN 2--- --- DEFN 3--|',                   ELE07890
     &  ' |-- DEFN 1--- --- DEFN 2--- --- DEFN 3--|',                   ELE07900
     &  ' |-- DEFN 1 -- --- DEFN 2--- --- DEFN 3--|'/6X,                ELE07910
     &  '  DUCT.  EXCR   DUCT.  EXCR.  DUCT.  EXCR.',                   ELE07920
     &  '  DUCT.  EXCR.  DUCT.  EXCR.  DUCT.  EXCR.',                   ELE07930
     &  '  DUCT.  EXCR.  DUCT.  EXCR   DUCT.  EXCR.')                   ELE07940
      RETURN                                                            ELE07950
C                                                                       ELE07960
C------------------------ WRITE MAXIMUM AND MINIMUM VALUES              ELE07970
 1200 CONTINUE                                                          ELE07980
      IF (FIRST) WRITE(108,491)                                         ELE07990
     &           'MAXIMUM LOADS AND DISPL AT MAXIMUM LOADS '            ELE08000
     &         //' NOTE- MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'   ELE08010
      WRITE(108,492) IELNO,LOAD,ID(JK(1)),ID(JK(2)),ID(JK(3)),          ELE08020
     &               ID(JK(4)),(PMAX(I),PMAX(3+I),I=1,3)                ELE08030
C                                                                       ELE08040
      RETURN                                                            ELE08050
C                                                                       ELE08060
C------------------------ RESET INTERNAL FORCES                         ELE08070
 1300 CONTINUE                                                          ELE08080
C                                                                       ELE08090
C. --- ZERO MATRICIES                                                  ELE08100
      DO 1310 I=1,ISE                                                   ELE08110
 1310   SE(I)=0                                                         ELE08120
      DO 1325 I=1,3                                                     ELE08130
         PMAX(I)=0                                                      ELE08140
         PMAX(I+3)=0                                                    ELE08150
         F (I)=0                                                        ELE08160
         FT(I)=0                                                        ELE08170
         R (I)=0                                                        ELE08180
         RT(I)=0                                                        ELE08190
         V (I)=0                                                        ELE08200
 1325    VT(I)=0                                                        ELE08210
      WRITE(108,1330) IELNO                                             ELE08220
 1330 FORMAT (' SHEAR WALL..... ELEMENT #',I6,                          ELE08230
     &        ' INTERNAL FORCES AND HYSTERESIS MODELS ARE RESET TO ZERO') ELE08240
      DO 1305 I=1,3                                                     ELE08250
        CALL MATLIB(2,LSTTYP,FALSE,ESEB(I),PSEB(I),DUCT(1,I),EXCR(1,I),  ELE08260
     &              FT,F,RT,R,0.,0.,LELEM,LMAT,MATB(I),MATTYP,          ELE08270
     &              SE(ISEB(I)),IELNO,DAMAGE)                           ELE08280
 1305 CONTINUE                                                          ELE08290
C                                                                       ELE08300
      RETURN                                                            ELE08310
C------------------- DAMAGE INDEX                                       ELE08320
 1400 IF (ELSTIC) RETURN                                                ELE08330
C. ... ... BENDING ENERGY, AND DAMAGE INDEX                             ELE08340
       DAMAGE=0                                                         ELE08350
       DO 1406 I=1,3                                                    ELE08360
        CALL MATLIB(5,LSTTYP,FALSE,ESEB(I),PSEB(I),DUCT(1,I),EXCR(1,I), ELE08370
     &              PMAX(I),0.,PMAX(I+3),0.,0.,LELEM,LMAT,MATB(I),     ELE08380
     &              MATTYP,SE(ISEB(I)),IELNO,DAMB(I))                   ELE08390
        DAMAGE=DAMAGE+DAMB(I)*(ESEB(I)+PSEB(I))                         ELE08400
 1406  CONTINUE                                                         ELE08410
C                                                                       ELE08420
      ESE=(ESEB(1)+ESEB(2)+ESEB(3))*LENGTH                              ELE08430
      PSE=(PSEB(1)+PSEB(2)+PSEB(3))*LENGTH                              ELE08440
      DAMAGE=DAMAGE*LENGTH                                              ELE08450
      IF (FIRST) WRITE(108,1420)                                        ELE08460
      WRITE(108,1410) IELNO,(DAMB(I),ESEB(I),PSEB(I),I=1,3)             ELE08470
 1410 FORMAT (1X,I5,1P,3(F14.5,2G14.5))                                 ELE08480
 1420 FORMAT (/6X,                                                      ELE08490
     &  ' |-------- BENDING COMPONENT ------------|',                    ELE08500
     &  ' |---------- SHEAR COMPONENT --- --- ----|',                    ELE08510
     &  ' |---------- AXIAL COMPONENT ------------|'/,' ELEM#',         ELE08520
     &  ' |-- DAMAGE -- ---- ESE ---- ---- PSE ---|',                   ELE08530
     &  ' |-- DAMAGE -- --- - ESE ---- ---- PSE ---|',                  ELE08540
     &  ' |- DAMAGE -- ---- ESE ---- ---- PSE ---|')                    ELE08550
      RETURN                                                            ELE08560
      END                                                               ELE08570
C                                                                       ELE08010
C-----------------------------------------------------------------------CELE00020
C                                                                       ELE00030
      SUBROUTINE ELE08                                                  ELE00040
     &  (IOPT      ,FIRST     ,PRINT     ,NAME      ,IELNO     ,        ELE00050
     &   ID        ,IDOF      ,COORD     ,COSINE    ,JTCOS     ,        ELE00060
     &   CONST     ,DISPL     ,VELOC     ,PUB       ,IREL      ,        ELE00070
     &   RINPUT    ,EPSE      ,EPSE      ,DAMAGE    ,DUCPAG    ,        ELE00080
     &   IELTYP    ,IELDOF    ,NODEI     ,NODEJ     ,JOINTI    ,        ELE00090
     &   JOINTJ    ,R         ,RT        ,F         ,FT        ,        ELE00100
     &   V         ,VT        ,ISE       ,H         ,LENGTH    ,        ELE00110
     &   MAT       ,A         ,LM        ,KTYPE     ,PMAX      ,        ELE00120
     &   SE        ,STIFF     )                                         ELE00130
C                                                                       ELE00140
      IMPLICIT REAL(A-H,O-Z)                                            ELE00150
C                                                                       ELE00160
      COMMON /RSA/ICOMN, RSAF(12)                                       ELE00170
      COMMON /IDSTEP/ISTEP                                              ELE00180
      REAL ICOMN,LENGTH                                                 ELE00190
      LOGICAL FIRST,PRINT,FALSE,BTEST                                   ELE00200
      CHARACTER*80 NAME                                                 ELE00210
      CHARACTER*9  TYPE(1)                                              ELE00220
$INCLUDE:'2COMM3'                                                       ELE00230
      DIMENSION  IDOF(MAXNOD,6),COORD(MAXNOD,6),COSINE(3,3,NCOS)       ELE00240
      DIMENSION CONST(MAXNOD,6),   ID(MAXNOD)  ,JTCOS(MAXNCD)          ELE00250
      DIMENSION SAT(2,12),STIFF(78),A(12,2),S(2,2),LM(12),PMAX(2)       ELE00260
      DIMENSION CTS( 3, 3),CTE( 3, 3),VX(3),VY(3),VELOC(NDOF)           ELE00270
      DIMENSION BS( 3, 3),BE( 3, 3),SE(ISE)                             ELE00280
      DIMENSION DISPL(NDOF,NLOAD),PUB(NDOF)                             ELE00290
      DIMENSION RINPUT(100),WORK(6),DUCT(3),EXCR(6)                     ELE00300
      DATA TYPE/'    AXIAL'/                                            ELE00310
C                                                                       ELE00320
C ---- ------------ ------------------------------------------------- ELE00330
C VARIABLES                                                             ELE00340
C ------ ------------ --- ------ ------ -- ------ ----- ------------- ELE00350
C---------- GLOBAL VARIABLES                                           ELE00360
C IOPT    - 1, INITIALIZE BEAM COLUMN ELEMENT                           ELE00370
C         - 2, CALCULATE GEOMETRIC STIFFNESS                            ELE00380
C         - 3, FORM STIFFNESS                                           ELE00390
C         - 4, CALCULATE INCREMENTAL FORCES                             ELE00400
C         - 5, CALCULATE TOTAL FORCES AND ENERGIES                      ELE00410
C IUNIT   - OUTPUT FILE UNIT # FOR PRINTING OUTPUT                      ELE00420
C FIRST   - FLAG, FIRST- TRUE., PRINT HEADERS                           ELE00430
C PRINT   - FLAG, PRINT-.TRUE., PRINT DATA                              ELE00440
C NDOF    - # OF GLOBAL DOF                                             ELE00450
C NLOAD   - # OF LOAD COMBINATIONS                                      ELE00460
C MAXNOD  - # ROW DIMENSION OF NODE ARRAY                               ELE00470
C NNODE   - # OF NODES                                                  ELE00480
C ID      - ARRAY OF EXTERNAL NODE NUMBERS                              ELE00490
C IDOF    - ARRAY OF DEGREES OF FREEDOM                                 ELE00500
C COORD   - ARRAY OF COORDINATES                                        ELE00510
C COSINE  - ARRAY OF DIRECTION COSINES OF NODES                         ELE00520
C CONST   - ARRAY OF CONSTRAINT TRANSFORMATIONS                         ELE00530
C DISPL   - GLOBAL DISPLACEMENT MATRIX                                  ELE00540
C Z       - REAL GLOBAL STORTAG VECTOR                                  ELE00550
C NZ      - INTEGER GLOBAL STORTAG VECTOR                               ELE00560
C IZMAT   - LOCATION OF MATERIAL DATA IN GLOBAL STORAGE VECTOR          ELE00570
C NAME    - ELEMENT NAME                                                ELE00580
C IELNO   - ELEMENT NUMBER                                              ELE00590
C RINPUT  - INPUT DATA                                                  ELE00600
C STIFF   - OUTPUT ELEMENT STIFFNESS (UPPER TRIANGULAR FORM)            ELE00610
C LM      - OUTPUT LOCAL TO GLOBAL STIFFNESS MAPPING MATRIX             ELE00620
C---------- ELEMENT VARIABLES                                          ELE00630
C H,Q,AI,AJ,C,BI,BJ,S22,S26,S212,S33,S35,S311,D                         ELE00640
C         - INDIVIDUAL TERMS IN THE ELEMENT STIFFNESS (S) MATRIX        ELE00650
C PKG     - INPUT LOAD FOR FORMING GEOMETRIC STIFFNESS MATRIX ..        ELE00660
C R       - INCRIMENTAL DISPLACEMENTS                                   ELE00670
C RT      - TOTAL DISPLACEMENTS                                         ELE00680
C F       - INCRIMENTAL FORCES                                          ELE00690
C FT      - TOTAL FORCES                                                ELE00700
C LM      - LOCAL TO GLOBAL STIFFNESS MAPPING MATRIX                    ELE00710
C CTS     - LOCAL TO GLOBAL ROTATION MATRIX START                       ELE00720
C CTE     - LOCAL TO GLOBAL ROTATION MATRIX END                         ELE00730
C BS      - LOCAL TO GLOBAL ROTATED ECCENTRICITY/CONSTRAINT  MATRIX START ELE00740
C BE      - LOCAL TO GLOBAL ROTATED ECCENTRICITY/CONSTRAINT  MATRIX END  ELE00750
C STIFF   - ELEMENT STIFFNESS                                           ELE00760
C IELTYP  - ELEMENT TYPE - 10                                           ELE00770
C NODEI   - EXTERNAL JOINT NUMBER, END I                                ELE00780
C NODEJ   - EXTERNAL JOINT NUMBER, END J                                ELE00790
C JOINTI  - INTERNAL JOINT NUMBER, END I                                ELE00800
C JOINTJ  - INTERNAL JOINT NUMBER, END J                                ELE00810
C---------- TEMPROARY VARIABLES                                         ELE00820
C VX      - VECTOR DEFINING THE ELEMENT X AXIS                          ELE00830
C VY      - VECTOR DEFINING THE ELEMENT Y AXIS                          ELE00840
C COSLOC  - LOCAL COSINE MATRIX                                         ELE00850
C A       - GLOBAL TO LOCAL FORCE TRANSFORMATION MATRIX                 ELE00860
C S       - ELEMENT STIFFNESS AT LOCAL COORD, AT ENDS OF FLEXABLE PART  ELE00870
C SAT     - PRODUCT OF S*A'                                             ELE00880
C ASAT    - PRODUCT OF A*SAT                                            ELE00890
C-----------------------------------------------------------------------ELE00900
C                                                                       ELE00910
C                                                                       ELE00920
C------- CHOOSE OPTION                                                  ELE00930
        IF (IOPT.LT.1 .OR. IOPT.GT 14) THEN                             ELE00940
          WRITE(108,*) 'INVALID OPTION IN ELE-08, IOPT=',IOPT           ELE00950
          STOP                                                          ELE00960
        ENDIF                                                           ELE00970
C                                                                       ELE00980
      GO TO (100,200,300,400,500,600,700,800,900,1000,                  ELE00990
     &                            1100,1200,1300,1400) IOPT             ELE01000
C                                                                       ELE01010
  100 CONTINUE                                                          ELE01020
C                                                                       ELE01030
C.. -- INTERPERTATE INPUT DATA                                          ELE01040
      IELTYP= 8                                                         ELE01050
      MAT   = RINPUT( 1)                                                ELE01060
      NODEI = RINPUT( 2)                                                ELE01070
      NODEJ = RINPUT( 3)                                                ELE01080
      KTYPE = RINPUT( 4)                                                ELE01090
      SLK   = RINPUT( 5)                                                ELE01100
      VY(1) = RINPUT( 6)                                                ELE01110
      VY(2) = RINPUT( 7)                                                ELE01120
      VY(3) = RINPUT( 8)                                                ELE01130
      XS    = RINPUT( 9)                                                ELE01140
      XE    = RINPUT(10)                                                ELE01150
                                                                        ELE01160
C                                                                       ELE01170
C------ GET INTERNAL NODE # ASSOCIATED WITH EXTERNAL NODE #'S           ELE01180
      JOINTI=IQUICK(NODEI,ID,NNODE)                                     ELE01190
      JOINTJ=IQUICK(NODEJ,ID,NNODE)                                     ELE01200
C                                                                       ELE01210
      IF (KTYPE.LT.1 .OR. KTYPE.GT.1) THEN                              ELE01220
        WRITE(108,102) IELNO,NODEI,NODEJ,KTYPE                          ELE01230
  102   FORMAT(1X,20('*'),'ER','ROR - INVALID KTYPE',/                  ELE01240
     &  5X,'IELNO=',I5,5X,'JOINTI=',I5,5X,'JOINTJ=',I5,5X,'KTYPE=',I5,/ ELE01250
     &  5X,'REVISE INPUT  FOR KTYPE EQ 1 ..... ....'/)                  ELE01260
      ENDIF                                                             ELE01270
C                                                                       ELE01280
C ---- PRINT DATA FOR PLOTTING                                          ELE01290
      IF (BTEST(IBUG,2)) THEN                                           ELE01300
        WRITE(7,10) NODEJ, NODEI,(COORD(JOINTJ,I),I=1,3),               ELE01310
     &                          (COORD(JOINTI,J),J=1,3)                 ELE01320
   10  FORMAT (I5,1X,I5,1P,6G12.4)                                      ELE01330
      ENDIF                                                             ELE01340
C                                                                       ELE01350
C-------------------------------------------------------------- ------ ELE01360
C  GET GLOBAL TO LOCAL TRANSFORMATION MATRIX                            ELE01370
C-------------------------------------------------------------- ------ ELE01380
C----- DEFINE VECTOR X, LENGTH                                          ELE01390
      VX(1)=COORD(JOINTJ,1)-COORD(JOINTI,1)                             ELE01400
      VX(2)=COORD(JOINTJ,2)-COORD(JOINTI,2)                             ELE01410
      VX(3)=COORD(JOINTJ,3)-COORD(JOINTI,3)                             ELE01420
      LENGTH=SQRT(VX(1)**2+VX(2)**2+VX(3)**2)-XS-XE                     ELE01430
      SE(5)=LENGTH                                                      ELE01440
      SE(2)=SLK                                                         ELE01450
      IF (LENGTH.LE.0) THEN                                             ELE01460
        WRITE(108,101) IELNO,JOINTI,JOINTJ,LENGTH                       ELE01470
  101   FORMAT(1X,20('**'),'ER', 'ROR - LENGTH IS LE 0',/               ELE01480
     &  5X,'IELNO=',I5,5X,'JOINTI=',I5,5X,'JOINTJ=',I5,                ELE01490
     &  5X,'LENGTH',1P,G15.6/                                           ELE01500
     &  5X,'REVISE INPUT..... ..... .  .  . ..... '/)                   ELE01510
        LENGTH=1.0                                                      ELE01520
      ENDIF                                                             ELE01530
C----- GET LOCAL TO GLOBAL TRANSFORMATION MATRICIES CT AND B            ELE01540
      ICS=JTCOS(JOINTI)                                                 ELE01550
      ICE=JTCOS(JOINTJ)                                                 ELE01560
      CALL ROTMY2(VX,VY,COSINE(1,1,ICS),CTS, XS ,0. ,0  ,BS,            ELE01570
     &            CONST(JOINTI,1),MAXNOD)                               ELE01580
      CALL ROTMY2(VX,VY,COSINE(1,1,ICE),CTE,-XE ,0  ,0  ,BE,            ELE01590
     &            CONST(JOINTJ,1),MAXNOD)                               ELE01600
C                                                                       ELE01610
C----- ZERO MATRICIES                                                   ELE01620
        F =0                                                            ELE01630
        FT=0                                                            ELE01640
        R =0                                                            ELE01650
        RT=0                                                            ELE01660
        V =0                                                            ELE01670
        VT=0                                                            ELE01680
        PMAX(1)=0                                                       ELE01690
        PMAX(2)=0                                                       ELE01700
C                                                                       ELE01710
C----  ASSEMBLE TRANSFORMATION MATRIX A                                 ELE01720
C  NOTE. A IS DIMMENSIONED AS A 12X2 MATRIX, BUT ONLY THE FIRST IELDOF  ELE01730
C        ROW'S CONTAIN INFORMATION. THE BALANCE OF THE MATRIX IS        ELE01740
C        UNDEFINED.  THE GLOBAL DOF COORESPONDING TO THE ROW'S OF A     ELE01750
C        ARE STORED IN VECTOR LM.  AGAIN, ONLY THE FIRST IELDOF ROW'S   ELE01760
C        OF LM ARE DEFINED.                                             ELE01770
      IELDOF=0                                                          ELE01780
      DO 50 I=1,12                                                      ELE01790
C                                                                       ELE01800
C             REMOVE DOF'S THAT ARE CONSTRAINED TO THE SAME DOF.        ELE01810
```

```
      J=I                                                               ELE01820
      IF (I.GT.6) J= I-6                                                ELE01830
      IF ( IDOF(JOINTI,J).EQ.IDOF(JOINTJ,J) ) GO TO 50                  ELE01840
      IF (I.LE 6) THEN                                                  ELE01850
          LMTI=IDOF(JOINTI,I)                                           ELE01860
      ELSE                                                              ELE01870
          LMTI=IDOF(JOINTJ,J)                                           ELE01880
      ENDIF                                                             ELE01890
C                                                                       ELE01900
      A1=0                                                              ELE01910
      A2=0                                                              ELE01920
      IF (I.LE.3) THEN                                                  ELE01930
          IF (KTYPE.LE.3) A1=CTS(I,KTYPE)                               ELE01940
      ELSE IF (I.LE.6) THEN                                             ELE01950
          IF (KTYPE.LE 3) THEN                                          ELE01960
              A1=BS(I-3,KTYPE)                                          ELE01970
          ELSE                                                          ELE01980
              A1=CTS(I-3,KTYPE-3)                                       ELE01990
          ENDIF                                                         ELE02000
      ELSE IF (I.LE.9) THEN                                             ELE02010
          IF (KTYPE.LE.3) A2=CTE(I-6,KTYPE)                             ELE02020
      ELSE                                                              ELE02030
          IF (KTYPE.LE.3) THEN                                          ELE02040
              A2=BE(I-9,KTYPE)                                          ELE02050
          ELSE                                                          ELE02060
              A2=CTE(I-9,KTYPE-3)                                       ELE02070
          ENDIF                                                         ELE02080
      ENDIF                                                             ELE02090
      IF (A1.EQ.0 .AND. A2.EQ.0) GO TO 50                               ELE02100
      IELDOF=IELDOF+1                                                   ELE02110
      A(IELDOF,1)=A1                                                    ELE02120
      A(IELDOF,2)=A2                                                    ELE02130
      LM(IELDOF)=LMTI                                                   ELE02140
C                                                                       ELE02150
   50 CONTINUE                                                          ELE02160
C                                                                       ELE02170
C---- GET GOEL STRUT STIFFNESS                                          ELE02180
      FALSE=.FALSE.                                                     ELE02190
      LSTTYP=0                                                          ELE02200
      CALL MATLIB(2,LSTTYP,FALSE,EESE,EPSE, DUCT,EXCR,                  ELE02210
     &        FT,F,RT,R,0.,0.,LELEM,LMAT,MAT,MATTYP,SE,IELNO,DAMAGE)    ELE02220
C                                                                       ELE02230
C------- H IS THE STRUT STIFFNESS                                       ELE02240
C                                                                       ELE02250
      H = SE(1)                                                         ELE02260
      HH= SE(1)                                                         ELE02270
      S(1,1)= HH                                                        ELE02280
      S(1,2)=-HH                                                        ELE02290
      S(2,1)=-HH                                                        ELE02300
      S(2,2)= HH                                                        ELE02310
C                                                                       ELE02320
C----- CALCULATE SAT                                                    ELE02330
      DO 60 I=1,2                                                       ELE02340
        DO 60 J=1,IELDOF                                                ELE02350
        SAT(I,J)=0                                                      ELE02360
          DO 60 K=1,2                                                   ELE02370
   60       SAT(I,J)=SAT(I,J)+S(I,K)*A(J,K)                             ELE02380
C                                                                       ELE02390
C----- CALCULATE ASAT AND CONVERT TO HALF STORAGE MODE  BY COLUMNS      ELE02400
C                                                                       ELE02410
      L=0                                                               ELE02420
      DO 70 J=1,IELDOF                                                  ELE02430
        DO 70 I=J,1,-1                                                  ELE02440
        L=L+1                                                           ELE02450
        STIFF(L)=0                                                      ELE02460
          DO 70 K=1,2                                                   ELE02470
   70       STIFF(L)=STIFF(L)+A(I,K)*SAT(K,J)                           ELE02480
C                                                                       ELE02490
C----- RELEASE UNUSED STORAGE                                           ELE02500
  180 IREL=78-L                                                         ELE02510
C                                                                       ELE02520
C     PRINT COLUMN DATA                                                 ELE02530
      IF (FIRST) WRITE (108,192)                                        ELE02540
  191 WRITE(108,193) NAME,IELNO,MAT,NODE1,NODEJ,TYPE(KTYPE),LENGTH,     ELE02550
     &                VY(1),VY(2),VY(3),XS, KE                         ELE02560
C                                                                       ELE02570
  192 FORMAT(/' ELEMENT D8, GNZ (LOCAL) DOF STRUT ELEMENT'//            ELE02580
     & T22,'# MATL START   END',2X,'---TYPE---',6X,'LENGTH',3X,         ELE02590
     & 11('-'),' Y-AXIS  ',12('-'),'  START DIST    END DIST')          ELE02600
  193 FORMAT(1X,A15,A16,2X,A5,3X,1P,G12.4,0P,                           ELE02610
     &    F9.5,' I',SP,F9.5,' J',F9.5,' K',1P,5E,3G12.4)               ELE02620
      RETURN                                                            ELE02630
C------------------------ DETERMINE GEOMETRIC STIFF ---------------------ELE02640
  200 RETURN                                                            ELE02650
C    THE GEOMETRIC STIFFNESS MATRIX IS NOT AVIALABLE FOR THIS ELEMENT   ELE02660
C                                                                       ELE02670
C------------------------ DETERMINE STIFFNESS ---------------------------ELE02680
  300 CONTINUE                                                          ELE02690
C                                                                       ELE02700
C-- --- H CONTAINS THE STRUT STIFFNESS, WHICH WAS DETERMINED IN THE     ELE02710
C       LAST CALL TO MAT_LIB                                            ELE02720
C       IF H=SE(1) THEN THE STIFFNESS HAS NOT CHANGED, RETURN           ELE02730
C                                                                       ELE02740
      IF (H EQ.SE(1)) RETURN                                            ELE02750
C                                                                       ELE02760
      H = SE(1)                                                         ELE02770
      HH= SE(1)                                                         ELE02780
      S(1,1)= HH                                                        ELE02790
      S(1,2)=-HH                                                        ELE02800
      S(2,1)=-HH                                                        ELE02810
      S(2,2)= HH                                                        ELE02820
C                                                                       ELE02830
C----- CALCULATE SAT                                                    ELE02840
      DO 360 I=1,2                                                      ELE02850
        DO 360 J=1,IELDOF                                               ELE02860
        SAT(I,J)=0                                                      ELE02870
          DO 360 K=1,2                                                  ELE02880
  360       SAT(I,J)=SAT(I,J)+S(I,K)*A(J,K)                             ELE02890
C                                                                       ELE02900
C----  CALCULATE ASAT, STORE IN UPPER TRIANGULAR MATRIX                 ELE02910
      L=0                                                               ELE02920
      DO 370 J=1,IELDOF                                                 ELE02930
        DO 370 I=J,1,-1                                                 ELE02940
        L=L+1                                                           ELE02950
        STIFF(L)=0                                                      ELE02960
          DO 370 K=1,2                                                  ELE02970
  370       STIFF(L)=STIFF(L)+A(I,K)*SAT(K,J)                           ELE02980
C                                                                       ELE02990
      RETURN                                                            ELE03000
C------------------------ DETERMINE INCRIMENTAL FORCES ------------------ELE03010
  400 CONTINUE                                                          ELE03020
      IF (PRINT.AND.FIRST) WRITE(108,491) STEPID                        ELE03030
C                                                                       ELE03040
C------ LOOP FOR EACH LOAD                                              ELE03050
      DO 430 LOAD=1,NLOAD                                               ELE03060
C                                                                       ELE03070
C------ TRANSFORM FRAME DISPL INTO LOCAL DISPL                          ELE03080
C .  LOCAL DISPL ARE AT THE END END OF THE STPUT, THUS                  ELE03090
C..  POSITIVE IS TENSION AND NEGATIVE IS COMPRESSION                    ELE03100
      R=0                                                               ELE03110
      V=0                                                               ELE03120
      DO 410 I=1,IELDOF                                                 ELE03130
         J=LM(I)                                                        ELE03140
           A1= A(I,1)                                                   ELE03150
           A2= A(I,2)                                                   ELE03160
           DISP= DISPL(J,LOAD)                                          ELE03170
           VEL= VELOC(J)                                                ELE03180
         R= R + (A(I,2) A(I,1)) * DISPL(J,LOAD)                         ELE03190
  410    V= V + (A(I,2)-A(I,1)) * VELOC(J)                              ELE03200
C                                                                       ELE03210
C------ INCRIMENTAL FORCES                                             ELE03220
      P= H*P                                                            ELE03230
C                                                                       ELE03240
C------- CHECK STIFFNESS AND CALCULATE UNBALANCED FORCES                ELE03250
C       THIS IS PREFORMED ONLY FOR INELASTIC ANALYSIS                   ELE03260
      P =FT-F                                                           ELE03270
      IF (.NOT. ELSTIC) THEN                                            ELE03280
C          -- SET UP TEMPORARY - TOTAL LOADS, DISPL'S AND VELOC         ELE03290
           FL=FT                                                        ELE03300
           D =(RT-R)                                                    ELE03310
           DL=RT                                                        ELE03320
           VC=VT+V                                                      ELE03330
           VL=VT                                                        ELE03340
C          -- TRANSFER PERMANENT HYSTERESIS DATA TO TEMPROARY STORAGE... ELE03350
           ISE2=ISE/2                                                   ELE03360
           DO 415 I=1,ISE2                                              ELE03370
  415        SE(I+ISE2)=SE(I)                                           ELE03380
C            - CALL HYST MODEL TO GET NEW STIFFNESS AND                 ELE03390
C          THE TOTAL FORCE P AT DISPL D                                 ELE03400
           CALL MATLIB(3,LSTTYP,FALSE,EESE,EPSE,DUCT,EXCR,P,PL,D,       ELE03410
     &        DL,VC,VL,LELEM,LMAT,MAT,MATTYP,SE(ISE2+1),IELNO,DAMAGE)   ELE03420
           IF (SE(ISE2+1).NE.SE(1)) NEWK= TRUE.                         ELE03430
      ENDIF                                                             ELE03440
C                                                                       ELE03450
C------ SAVE PEAK VALUES FOR MULTIPLE LOAD CASES(FOR STATICS CASE)      ELE03460
      IF (NLOAD GT.1 .AND.  ABS(P) GT.ABS(PMAX(1)) ) THEN               ELE03470
        PMAX(1)=P                                                       ELE03480
        PMAX(2)=R                                                       ELE03490
      ENDIF                                                             ELE03500
C                                                                       ELE03510
C------ PRINT DATA                                                      ELE03520
      HH= H                                                             ELE03530
      IF (PRINT) WRITE(108,492) IELNO,LOAD,TYPE(KTYPE),NODE1,NODEJ,     ELE03540
     &                          P,R,HH                                  ELE03550
C                                                                       ELE03560
  430 CONTINUE                                                          ELE03570
C                                                                       ELE03580
C------ THIS IS FOR SOL05A USE ONLY                                     ELE03590
      IF( ICOMN .EQ. 1 .OR  ICOMN .EQ  2 ) RSAF(1)=P                    ELE03600
C                                                                       ELE03610
C------ CALCULATE THE UNBALANCED MEMBER FORCE ON A JOINT.               ELE03620
           UBAL= P-FT-P                                                 ELE03630
           P   = P                                                      ELE03640
           FT  = FT                                                     ELE03650
           P   = P                                                      ELE03660
           DO 440 I=1,IELDOF                                            ELE03670
              J=LM(I)                                                   ELE03680
              A1= A(I,1)                                                ELE03690
              A2= A(I,2)                                                ELE03700
              FUB(J)= FUB(J) + (A(I,1)-A(I,2)) * UBAL                   ELE03710
  440      CONTINUE                                                     ELE03720
C                                                                       ELE03730
  491 FORMAT (/'       GOEL FORCES...',5X,A/,                           ELE03740
     & '   ELEMENT LOAD',T17,'       TYPE',' NODEI',' NODEJ',         ELE03750
     & '       FORCE    ',' DISPLACMENT   ',    STIFFNESS    ')        ELE03760
  492 FORMAT (I10,I5,2X,A9.2I6,1P,3G15.6)                               ELE03770
C                                                                       ELE03780
C------ ADJUST P FOR UNBALANCED LOAD P                                 ELE03790
      P=P-FT                                                            ELE03800
      RETURN                                                            ELE03810
C------------------------ DETERMINE TOTAL FORCES, ENERGIES ECT ----------ELE03820
  500 CONTINUE                                                          ELE03830
      IF( ICOMN .NE. 1 .AND. ICOMN .NE. 2 ) THEN                        ELE03840
       IF (PRINT.AND.FIRST) WRITE(108,491) STEPID                       ELE03850
C                                                                       ELE03860
C------ TOTAL FORCES, DISPLACEMENTS AND VELOCITIES                      ELE03870
C                                                                       ELE03880
      FT=FT+P                                                           ELE03890
      RT=RT+R                                                           ELE03900
      VT=VT+V                                                           ELE03910
      IF ( ABS(FT).GT.ABS(PMAX(1)) ) PMAX(1)=FT                         ELE03920
      IF ( ABS(RT).GT.ABS(PMAX(2)) ) PMAX(2)=RT                         ELE03930
C                                                                       ELE03940
C------ PRINT DATA                                                      ELE03950
      HH= H                                                             ELE03960
      IF (PRINT) WRITE(108,492) IELNO,1,TYPE(KTYPE),NODE1,NODEJ,FT,RT,HHELE03970
      ELSE IF( ICOMN .EQ. 1 .OR. ICOMN .EQ. 2 ) THEN                    ELE03980
       IF (PRINT.AND.FIRST) WRITE(108,497) STEPID                       ELE03990
        FT=REAF(1)                                                      ELE04000
        IF (PRINT) WRITE(108,492) IELNO,1,TYPE(KTYPE),NODE1,NODEJ,FT,HH ELE04010
      ENDIF                                                             ELE04020
  497 FORMAT (/'       GOEL FORCES...',5X,A/,                           ELE04030
     & '   ELEMENT LOAD',T17,'       TYPE',' NODE1',' NODEJ',         ELE04040
     & '       FORCE    ',' STIFFNESS    ')                           ELE04050
C                                                                       ELE04060
C------ TRANSFER TEMPROARY HYSTERESIS DATA TO PERMANENT STORAGE...      ELE04070
      ISE2=ISE/2                                                        ELE04080
      IF (.NOT. ELSTIC) THEN                                            ELE04090
        DO 510 I=1,ISE2                                                 ELE04100
  510     SE(I)=SE(I+ISE2)                                              ELE04110
      ENDIF                                                             ELE04120
C                                                                       ELE04130
      RETURN                                                            ELE04140
C                                                                       ELE04150
C------------------------ DETERMINE FIXED END FORCES, AND ADD TO FORCE-ELE04160
  600 RETURN                                                            ELE04170
C                                                                       ELE04180
C ------ FIXED END FORCES ARE NOT AVAILABLE FOR THE STRUT               ELE04190
C                                                                       ELE04200
C-------------------- WRITE MEMBER FORCES TO OUTPUT FILE ---------------ELE04210
  700 CONTINUE                                                          ELE04220
C                                                                       ELE04230
      CALL MATLIB(4,LSTTYP,FALSE.EESE,EPSE, DUCT,EXCR,                  ELE04240
     &        FT,F,RT,R,0.,0.,LELEM,LMAT,MAT,MATTYP,SE,IELNO,DAMAGE)    ELE04250
C                                                                       ELE04260
      WRITE (IUNIT,710)  NODE1,NODEJ,KTYPE,FT,RT,H,EESE,EPSE,           ELE04270
     &                   (DUCT(I),EXCR(I),I=1,3)                        ELE04280
  710 FORMAT (3I6,1P,5G12.4/6G12.4)                                     ELE04290
C                                                                       ELE04300
      RETURN                                                            ELE04310
C                                                                       ELE04320
C------------------- READ AND PRINT MEMBER FORCES FROM OUTPUT FILE ----ELE04330
  800 CONTINUE                                                          ELE04340
      BACKSPACE(IUNIT)                                                  ELE04350
      READ (IUNIT,"   ) ITYPE,IELNO,IWRITE,TO,DT                       ELE04360
      ISTEP=-IWRITE                                                     ELE04370
C                                                                       ELE04380
  810 READ (IUNIT,815,END=830) NODE1,NODEJ,KTYPE,FT,RT,H,ESE,PSE,       ELE04390
     &                   (WORK(I),I=1,6)                                ELE04400
  815 FORMAT (3I6,5G12.4/6G12.4)                                        ELE04410
        ISTEP=ISTEP-IWRITE                                              ELE04420
        T=TO+DT*ISTEP                                                   ELE04430
        IF (ISTEP.EQ.0) WRITE(108,805) NODE1,NODEJ,TYPE(KTYPE)          ELE04440
        IF (MOD(ISTEP,INC) EQ.0) WRITE(108,820) ISTEP,T,FT,RT,H,ESE,PSEELE04450
        GO TO 810                                                       ELE04460
C                                                                       ELE04470
  805 FORMAT (/' GOEL MEMBER FORCES  ',/                                ELE04480
     & 5X,'NODE1:',I6,5X,'NODEJ:',I6.5X,'GOEL"S TYPE:',A,//             ELE04490
     & '    STEP      TIME       ','       FORCE      ','             ELE04500
     & ' DISPLACMENT ','  STIFFNESS','      ESE      ','      PSE     ') ELE04510
  820 FORMAT (I10,1P,G15.5,6G13.5)                                      ELE04520
  830 WRITE(108,840) (WORK(I),I=1,6)                                    ELE04530
  840 FORMAT (/T10,' MAXIMUN DUCTILITIES AND EXCURSION RATIOS'/         ELE04540
     &        T10,'----------------------------------------'/          ELE04550
     & T15,'DISPLACEMENT DEFINATION: U1=',1P,G12.4,5X,'E1=',G12.4/      ELE04560
     & T15,'ENERGY DEFINATION #1.    U2=',1P,G12.4,5X,'E2=',G12.4/      ELE04570
     & T15,'ENERGY DEFINATION #2.    U3=',1P,G12.4,5X,'E3=',G12.4)      ELE04580
      RETURN                                                            ELE04590
C                                                                       ELE04600
C-------------------- DETERMINE THE LENGTH OF THE MATERIAL DATA --------ELE04610
  900 CONTINUE                                                          ELE04620
      ISE=0                                                             ELE04630
C                                                                       ELE04640
        MAT   = RINPUT( 1)                                              ELE04650
        CALL MATLIB(0,LSTTYP,FALSE,EESE,EPSE, DUCT,EXCR,                ELE04660
     &        FT,F,RT,R,0.,0 ,LELEM,LMAT,MAT,MATTYP,SE,IELNO,DAMAGE)    ELE04670
        ISE=ISE+LELEM                                                   ELE04680
C                                                                       ELE04690
C------  DOUBLE THE STORAGE, TO ALLOW FOR TEMPROARY VALUES...           ELE04700
      ISE=ISE*2                                                         ELE04710
      RETURN                                                            ELE04720
C                                                                       ELE04730
```

```
C-------------------- DETERMINE THE STRAIN ENERGY                          ELE04740
 1000 CONTINUE                                                            ELE04750
C                                                                         ELE04760
C-------------------- DUCTILITIES AND EXCURSION RATIO'S                    ELE04770
 1100 CONTINUE                                                            ELE04780
C.... OPTIONS 10 AND 11 HAVE THE SAME CODE FOR THIS ELEMENT...            ELE04790
C.... THE ENERGY WAS CALCULATED WITH THE LAST CALL FOR INCRIMENTAL FORCES ELE04800
C     THE TOTAL FORCES HAVE NOT BEEN CALLED YET,THUS THE ENERGIES RESIDE  ELE04810
C     IN THE MATRIX SE BEGINING AT ADDRESS ISE21                          ELE04820
C                                                                         ELE04830
      ISE21=ISE/2 +1                                                      ELE04840
      CALL MATLIB(4,LSTTYP,FALSE,EESE,EPSE, DUCT,EXCR,                    ELE04850
     &   FT,F,RT,R,0.,0 ,LELEM,LMAT,MAT,MATTYP,SE(ISE21),IELNO,DAMAGE)    ELE04860
C                                                                         ELE04870
      IF(ABS(DUCT(1)) .LT. 1 ) DUCT(1)=0                                  ELE04880
      IF( DUCFAG .EQ. 0 ) THEN                                            ELE04890
         IF( ABS(DUCT(1)) .LT. 1 ) DUCFAG=0                               ELE04900
         IF( ABS(DUCT(1)) .GE. 1 ) DUCFAG=ABS(DUCT(1))                    ELE04910
      ENDIF                                                               ELE04920
C                                                                         ELE04930
      IF (ELSTIC) EESE=(FT+F)*(RT+R)*0.50                                 ELE04940
C                                                                         ELE04950
      IF (PRINT.AND.FIRST) WRITE(108,1191)                                ELE04960
      IF (PRINT) WRITE(108,1192) IELNO,DUCT(1),EXCR(1)                    ELE04970
 1191 FORMAT (/'  STRUT DUCTILITIES AND EXCURSION RATIOS . .',5X/         ELE04980
     & '    ELEMENT',T12,'!- DUCTILITY DEFINATION #1 -! ',/ T12,          ELE04990
     & '    DUCTILITY    EXCURSION      ')                               ELE05000
 1192 FORMAT (I10,1P,6G15.6)                                              ELE05010
C                                                                         ELE05020
      RETURN                                                              ELE05030
C                                                                         ELE05040
C-------------------- MAXIMUM FORCES                                       ELE05050
 1200 CONTINUE                                                            ELE05060
      IF (FIRST) WRITE(108,491)                                           ELE05070
     &           'MAXIMUM LOAD AND DISPL AT MAXIMUM LOAD '                ELE05080
     &          //' NOTE- MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'    ELE05090
      WRITE(108,492) IELNO,0,TYPE(KTYPE),NODEI,NODEJ,FMAX(1),FMAX(2)      ELE05100
      RETURN                                                              ELE05110
C                                                                         ELE05120
C                                                                         ELE05130
C-------------------- RESET INTERNAL FORCES                                ELE05140
 1300 CONTINUE                                                            ELE05150
C                                                                         ELE05160
C----- ZERO MATRICIES                                                     ELE05170
         P =0                                                             ELE05180
         PT=0                                                             ELE05190
         R =0                                                             ELE05200
         RT=0                                                             ELE05210
         V =0                                                             ELE05220
         VT=0                                                             ELE05230
         FMAX(1)=0                                                        ELE05240
         FMAX(2)=0                                                        ELE05250
      SLX=SE(2)                                                           ELE05260
      DO 1310 I=1,ISE                                                     ELE05270
 1310   SE(I)=0                                                           ELE05280
      SE(2)=SLX                                                           ELE05290
      SE(5)=LENGTH                                                        ELE05300
C                                                                         ELE05310
      WRITE(108,1330) IELNO                                               ELE05320
 1330 FORMAT (' GOEL ......... ELEMENT #',I6,                            ELE05330
     & '  INTERNAL FORCES AND HYSTERESIS MODELS ARE RESET TO ZERO')       ELE05340
      CALL MATLIB(2,LSTTYP,FALSE,EESE,EPSE, DUCT,EXCR,                    ELE05350
     &          FT,F,RT,R,0.,0.,LELEM,LMAT,MAT,MATTYP,SE,IELNO,DAMAGE)    ELE05360
      RETURN                                                              ELE05370
C                                                                         ELE05380
C-------------------- DAMAGE INDEX ----------------------                 ELE05390
C                                                                         ELE05400
 1400 IF (ELSTIC) RETURN                                                  ELE05410
C                                                                         ELE05420
      FMAX1=FMAX(1)                                                       ELE05430
      FMAX2=FMAX(2)                                                       ELE05440
      CALL MATLIB(5,LSTTYP,FALSE,EESE,EPSE,DUCT,EXCR,FMAX1,0.,            ELE05450
     &      FMAX2,0.,0 ,0.,LELEM,LMAT,MAT,MATTYP,SE,IELNO,DAMAGE)        ELE05460
C                                                                         ELE05470
      IF (PRINT.AND.FIRST) WRITE(108,1410)                                ELE05480
      IF (PRINT) WRITE(108,1420) IELNO,DAMAGE,FMAX1,FMAX2,EESE,EPSE       ELE05490
 1410 FORMAT (/'   ELEMENT',T12,' DAMAGE INDEX      F MAX      ',        ELE05500
     &          '     D MAX          ESE          ','      PSE       ')  ELE05510
 1420 FORMAT (I10,1P,6G15.6)                                              ELE05520
      DAMAGE=DAMAGE*(EESE+EPSE)                                           ELE05530
C                                                                         ELE05540
      RETURN                                                              ELE05550
      END                                                                 ELE05560
C                                                                         ELE10010
C-------------------------------------------------------------------------ELE10020
C                                                                         ELE10030
      SUBROUTINE ELE14(                                                   ELE10040
     &  (IOPT        ,FIRST       ,PRINT       ,PGEOM       ,NAME       ,ELE10050
     &   ID          ,IDOF        ,COORD       ,COSINE      ,JTCOS      ,ELE10060
     &   CONST       ,DISPL       ,PUB         ,IRBL        ,KGDATA     ,ELE10070
     &   IELNO       ,RINPUT      ,AXIAL       ,ESE         ,PSE        ,ELE10080
     &   IELTYP      ,IELDOF      ,ISES        ,ISE         ,JX         ,ELE10090
     &   MATS        ,R           ,                                      ELE10100
     &   RT          ,P           ,PT          ,V           ,VT         ,ELE10110
     &   PKG         ,LKG         ,LM          ,LMKG        ,A          ,ELE10120
     &   SS          ,SA1         ,SA2         ,DAMAGE      ,DUCFAG     ,ELE10130
     &   KGDOF       ,LENGTH      ,FMAX        ,SE          ,STIFF      )ELE10140
      COMMON /RSA/ICOMN,RSAP(12)                                          ELE10150
      REAL ICOMN,LENGTH                                                   ELE10160
      LOGICAL FIRST,PRINT,FALSE,AXIAL                                     ELE10170
      CHARACTER*80 NAME                                                   ELE10180
$INCLUDE '2COMN3'                                                         ELE10190
      DIMENSION IDOF(MAXNOD,6),COORD(MAXNOD,6),COSINE(3,3,NCOS)          ELE10200
      DIMENSION CONST(MAXNOD,6),ID(MAXNOD),JTCOS(MAXNOD),AA(4,6,3)       ELE10210
      DIMENSION SAT(3,24),STIFF(600),KGDATA(3),SAT2(6,24),VZB(3)         ELE10220
      DIMENSION AT(24,6),A(24,3),S(6,6),LM(24),AC(6,3),LMKG(24)          ELE10230
      DIMENSION CT(3,3),BS(3,3),VX(3),VY(3),VMB(3),VXT(3),VZT(3)         ELE10240
      DIMENSION SE(ISE),MATS(3),ISES(3),ESES(3),PSES(3),DAMS(3)          ELE10250
      DIMENSION F(3),R(3),V(3),FT(3),RT(3),VT(3),P(3),JX(4)              ELE10260
      DIMENSION DISPL(NDOF,NLOAD),PUB(NDOF),EPUB(24),FMAX(6)             ELE10270
      DIMENSION RINPUT(100),DUCT(3,3),EXCR(6,3)                          ELE10280
C                                                                         ELE10290
C-------------------------------------------------------------------------ELE10300
C VARIABLES                                                               ELE10310
C-------------------------------------------------------------------------ELE10320
C---------- GLOBAL VARIABLES                                             ELE10330
C IOPT   - 1, INITIALIZE BEAM COLUMN ELEMENT                              ELE10340
C        - 2, CALCULATE GEOMETRIC STIFFNESS                               ELE10350
C        - 3, FORM STIFFNESS                                              ELE10360
C        - 4, CALCULATE INCREMENTAL FORCES                                ELE10370
C        - 5, CALCULATE TOTAL FORCES AND ENERGIES                         ELE10380
C IUNIT  - OUTPUT FILE UNIT # FOR PRINTING OUTPUT                         ELE10390
C FIRST  - FLAG, FIRST=.TRUE., PRINT HEADERS                              ELE10400
C PRINT  - FLAG, PRINT= TRUE., PRINT DATA                                 ELE10410
C NDOF   - # OF GLOBAL DOF                                                ELE10420
C NLOAD  - # OF LOAD COMBINATIONS                                         ELE10430
C MAXNOD - # ROW DIMENSION OF NODE ARRAY                                  ELE10440
C NNODE  - # OF NODES                                                     ELE10450
C ID     - ARRAY OF EXTERNAL NODE NUMBERS                                 ELE10460
C IDOF   - ARRAY OF DEGREES OF FREEDOM                                    ELE10470
C COORD  - ARRAY OF COORDINATES                                           ELE10480
C COSINE - ARRAY OF DIRECTION COSINES OF NODES                            ELE10490
C CONST  - ARRAY OF CONSTRAINT TRANSFORMATIONS                            ELE10500
C DISPL  - GLOBAL DISPLACEMENT MATRIX                                     ELE10510
C Z      - REAL GLOBAL STORTAG VECTOR                                     ELE10520
C NZ     - INTEGER GLOBAL STORTAG VECTOR                                  ELE10530
C IZMAT  - LOCATION OF MATERIAL DATA IN GLOBAL STORAGE VECTOR             ELE10540
C NAME   - ELEMENT NAME                                                   ELE10550
C IELNO  - ELEMENT NUMBER                                                 ELE10560
C RINPUT - INPUT DATA                                                     ELE10570
C STIFF  - OUTPUT ELEMENT STIFFNESS (UPPER TRIANGULAR FORM)               ELE10580
C LM     - OUTPUT LOCAL TO GLOBAL STIFFNESS MAPPING MATRIX                ELE10590
C----  ---- ELEMENT VARIABLES                                            ELE10600
C H,Q,A1,A2,C,B1,B2,S22,S26,S212,S33,S35,S311,D                           ELE10610
C        - INDIVIDUAL TERMS IN THE ELEMENT STIFFNESS (S) MATRIX           ELE10620
C PKG    - INPUT LOAD FOR FORMING GEOMETRIC STIFFNESS MATRIX...           ELE10630
C R      - INCRIMENTAL DISPLACEMENTS                                      ELE10640
C RT     - TOTAL DISPLACEMENTS                                            ELE10650
C F      - INCRIMENTAL FORCES                                             ELE10660
C FT     - TOTAL FORCES                                                   ELE10670
C LM     - LOCAL TO GLOBAL STIFFNESS MAPPING MATRIX                       ELE10680
C CTS    - LOCAL TO GLOBAL ROTATION MATRIX START                          ELE10690
C CTE    - LOCAL TO GLOBAL ROTATION MATRIX END                            ELE10700
C BS     - LOCAL TO GLOBAL ROTATED ECCENTRICITY/CONSTRAINT  MATRIX START  ELE10710
C BE     - LOCAL TO GLOBAL ROTATED ECCENTRICITY/CONSTRAINT  MATRIX END    ELE10720
C STIFF  - ELEMENT STIFFNESS                                              ELE10730
C IELTYP - ELEMENT TYPE - 10                                              ELE10740
C NODEI  - EXTERNAL JOINT NUMBER, END I                                   ELE10750
C NODEJ  - EXTERNAL JOINT NUMBER, END J                                   ELE10760
C JOINTI - INTERNAL JOINT NUMBER, END I                                   ELE10770
C JOINTJ - INTERNAL JOINT NUMBER, END J                                   ELE10780
C---------- TEMPROARY VARIABLES                                           ELE10790
C VX     - VECTOR DEFINING THE ELEMENT X AXIS                             ELE10800
C VY     - VECTOR DEFINING THE ELEMENT Y AXIS                             ELE10810
C COSLOC - LOCAL COSINE MATRIX                                            ELE10820
C A      - GLOBAL TO LOCAL FORCE TRANSFORMATION MATRIX                    ELE10830
C S      - ELEMENT STIFFNESS AT LOCAL COORD, AT ENDS OF FLEXABLE PART     ELE10840
C SAT    - PRODUCT OF S*A'                                                ELE10850
C ASAT   - PRODUCT OF A*SAT                                               ELE10860
C ---------------------------------------------------------------------   ELE10870
C                                                                         ELE10880
C                                                                         ELE10890
C------- CHOOSE OPTION                                                    ELE10900
         IF (IOPT.LT.1 .OR.IOPT.GT.14) THEN                               ELE10910
           WRITE(108,*) 'INVALID OPTION IN ELE-14, IOPT=',IOPT            ELE10920
           STOP                                                           ELE10930
         ENDIF                                                            ELE10940
C                                                                         ELE10950
      GO TO (100,200,300,400,500,600,700,800,900,1000,                    ELE10960
     &                              1100,1200,1300,1400) IOPT             ELE10970
C                                                                         ELE10980
  100 CONTINUE                                                            ELE10990
C                                                                         ELE11000
C---- INTERPERTATE INPUT DATA                                            ELE11010
      IELTYP= 14                                                          ELE11020
      MATS(1)= RINPUT( 1)                                                 ELE11030
      MATS(2)= RINPUT( 2)                                                 ELE11040
      MATS(3)= RINPUT( 3)                                                 ELE11050
      NODE1 = RINPUT( 4)                                                  ELE11060
      NODE2 = RINPUT( 5)                                                  ELE11070
      NODE3 = RINPUT( 6)                                                  ELE11080
      NODE4 = RINPUT( 7)                                                  ELE11090
      ALPIN = RINPUT( 8)                                                  ELE11100
      PKG   = RINPUT( 9)                                                  ELE11110
C                                                                         ELE11120
C----- GET INTERNAL NODE # ASSOCIATED WITH EXTERNAL NODE #'S              ELE11130
      JX(1)=IQUICK(NODE1,ID,NNODE)                                        ELE11140
      JX(2)=IQUICK(NODE2,ID,NNODE)                                        ELE11150
      JX(3)=IQUICK(NODE3,ID,NNODE)                                        ELE11160
      JX(4)=IQUICK(NODE4,ID,NNODE)                                        ELE11170
C                                                                         ELE11180
C-------------------------------------------------------------------      ELE11190
C  GET GLOBAL TO LOCAL TRANSFORMATION MATRIX                              ELE11200
C-------------------------------------------------------------------      ELE11210
      DO 15 I=1,3                                                         ELE11220
C----- DEFINE VECTOR X, AT BOTTOM OF THE WALL                             ELE11230
      VXB(I)=COORD(JX(4),I)-COORD(JX(3),I)                                ELE11240
C                                                                         ELE11250
C----- DEFINE VECTOR X, AT TOP OF THE WALL                                ELE11260
      VXT(I)=COORD(JX(1),I)-COORD(JX(2),I)                                ELE11270
C                                                                         ELE11280
C----- CALCULATE Y VECTOR ..                                              ELE11290
      VY(I)=(COORD(JX(1),I)+COORD(JX(2),I)-COORD(JX(3),I)-                ELE11300
     &       COORD(JX(4),I))/2                                            ELE11310
   15 CONTINUE                                                            ELE11320
C                                                                         ELE11330
C ---- CALCULATE AVERAGE LENGTH,ALPHA AND BETA.                           ELE11340
      LENGTH=SQRT(VY(1)**2+VY(2)**2+VY(3)**2)                             ELE11350
      SE(ISES(1)+7)=LENGTH                                                ELE11360
      IF (LENGTH.LE.0) THEN                                               ELE11370
         WRITE(108,101) IELNO,LENGTH                                      ELE11380
  101    FORMAT(1X,20('*'),'ER',I,'ROR - LENGTH IS LE 0',/                ELE11390
     &          5X,'IELNO=',I5,5X,5X,'LENGTH',1P,G15.6/                   ELE11400
     &          5X,'REVISE INPUT.......................... ....')        ELE11410
         LENGTH=1.0                                                       ELE11420
      ENDIF                                                               ELE11430
C                                                                         ELE11440
      ALPHA=ALPIN*LENGTH                                                  ELE11450
      BETA = LENGTH-ALPHA                                                 ELE11460
C                                                                         ELE11470
C---- NORMALIZE Y VECTOR ...                                             ELE11480
      VY(1)=VY(1)/LENGTH                                                  ELE11490
      VY(2)=VY(2)/LENGTH                                                  ELE11500
      VY(3)=VY(3)/LENGTH                                                  ELE11510
C                                                                         ELE11520
C---- CALCULATE THE WIDTH ..                                             ELE11530
      CALL VCROSS(VZT,VXT,VY)                                             ELE11540
      CALL VCROSS(VZB,VXB,VY)                                             ELE11550
      WT = SQRT(VZT(1)**2 + VZT(2)**2 + VZT(3)**2)                        ELE11560
      WB = SQRT(VZB(1)**2 + VZB(2)**2 + VZB(3)**2)                        ELE11570
      W  = (WT+WB)/2.0                                                    ELE11580
C                                                                         ELE11590
C----- NORMALIZE Z VECTOR  .                                              ELE11600
      VZB(1)=VZB(1)/W                                                     ELE11610
      VZB(2)=VZB(2)/W                                                     ELE11620
      VZB(3)=VZB(3)/W                                                     ELE11630
      SE(ISES(1)+8)=W                                                     ELE11640
C                                                                         ELE11650
C----- CHECK FOR TWIST                                                    ELE11660
      TWIST=ACOS(VCOS(VZT,VZB))*57.2958                                   ELE11670
      IF (ABS(TWIST).GT .5.0) WRITE(108,102) TWIST                        ELE11680
  102 FORMAT(5X,10('*'),' WARNING THE WALL IS TWISTED=',1P,G12.4,         ELE11690
     &       ' DEGREES, REVISE INPUT AND RERUN')                          ELE11700
C                                                                         ELE11710
C----- CALCULATE VX, AND NORMALIZE                                        ELE11720
      CALL VCROSS(VX,VY,VZB)                                              ELE11730
      WX=SQRT(VX(1)**2+VX(2)**2+VX(3)**2)                                 ELE11740
      VX(1)=VX(1)/WX                                                      ELE11750
      VX(2)=VX(2)/WX                                                      ELE11760
      VX(3)=VX(3)/WX                                                      ELE11770
C                                                                         ELE11780
C----- GET LOCAL TO GLOBAL TRANSFORMATION MATRICIES                       ELE11790
C--- JOINT #1---#6                                                        ELE11800
      DO 35 L=1,6                                                         ELE11810
         CALL ROTXYZ(VX,VY,COSINE(1,1,JTCOS(JX(L))),CT,                   ELE11820
     &               0 ,0.,0.,BS,  CONST(JX(L),1),MAXNOD)                 ELE11830
         DO 21 I=1,3                                                      ELE11840
         DO 21 J=1,3                                                      ELE11850
           AA(L,I  ,J)=CT(I,J)                                            ELE11860
   21    AA(L,I+3,J)=BS(I,J)                                              ELE11870
   35 CONTINUE                                                            ELE11880
C                                                                         ELE11890
C----- ZERO MATRICIES                                                     ELE11900
      DO 25 I=1,3                                                         ELE11910
         FMAX(1  )= 0                                                     ELE11920
         FMAX(1+3)= 0                                                     ELE11930
         P (I)=0                                                          ELE11940
         PT(I)=0                                                          ELE11950
         R (I)=0                                                          ELE11960
         RT(I)=0                                                          ELE11970
         V (I)=0                                                          ELE11980
   25    VT(I)=0                                                          ELE11990
      DO 27 I=1,24                                                        ELE12000
         LM  (I)=0                                                        ELE12010
         LMKG(I)=0                                                        ELE12020
         DO 26 J=1,3                                                      ELE12030
   26       A(I,J)=0.00                                                  ELE12040
         DO 27 K=1,6                                                      ELE12050
   27       AT(I,K)=0.00                                                 ELE12060
C                                                                         ELE12070
C----- ASSEMBLE TRANSFORMATION MATRIX A = AS X A4 X A3                    ELE12080
C  NOTE  AT IS DIMMENSIONED AS A 24X6 MATRIX, BUT ONLY THE FIRST IELDOF   ELE12090
```

```
C       ROW'S CONTAIN INFORMATION. THE BALANCE OF THE MATRIX IS
C       UNDEFINED   THE GLOBAL DOF COORESPONDING TO THE ROW'S OF A
C       ARE STORED IN VECTOR LM.  AGAIN, ONLY THE FIRST IELDOF ROW'S
C       OF LM ARE DEFINED.
        IELDOF=0
        DO 50 I=1,4
        DO 50 J=1,6
C
C               REMOVE DOF'S THAT ARE CONSTRAINED TO THE SAME DOF..
        IF ( (IDOF(JK(1),J).EQ.IDOF(JK(4),J))  OR
     &       (IDOF(JK(1),J).EQ.IDOF(JK(3),J))  OR
     &       (IDOF(JK(2),J).EQ.IDOF(JK(4),J))  .OR.
     &       (IDOF(JK(2),J).EQ.IDOF(JK(3),J)) ) GO TO 50
C
        L0=I-1
        IF (I.EQ.4) L0=I-2
        IELDOF=IELDOF+1
        LMTI=IDOF(JK(1),J)
        AT(IELDOF,L0)=AA(I,J,2)
        IF (I.EQ.1.OR.I.EQ.4) AT(IELDOF,L0-1)=AA(I,J,1)
C
        LM(IELDOF)=LMTI
        IF((AT(IELDOF,1).EQ.0).AND.(AT(IELDOF,2).EQ.0)  .AND.
     &     (AT(IELDOF,3).EQ.0).AND.(AT(IELDOF,4).EQ.0)  .AND.
     &     (AT(IELDOF,5).EQ.0).AND.(AT(IELDOF,6).EQ.0)) THEN
          IELDOF=IELDOF-1
        ELSE IF (IELDOF.GT.1) THEN
          DO 49  II=1,IELDOF-1
            IF (LM(II).NE.LM(IELDOF)) GOTO 49
            DO 48 JJ=1,6
             AT(II,JJ)=AT(II,JJ)+AT(IELDOF,JJ)
   48       AT(IELDOF,JJ)=0
            IELDOF=IELDOF-1
            GO TO 50
   49     CONTINUE
        ENDIF
C
   50   CONTINUE
C
C- ---- A2A1 FOR STIFFNESS ONLY...
        DO 52 I=1,6
        DO 52 J=1,3
52      AC(I,J)=0.
        AC(1,1)=  1.
        AC(2,1)=  ALPHA/W
        AC(3,1)= -ALPHA/W
        AC(4,1)= -BETA /W
        AC(5,1)= -1.
        AC(6,1)=  BETA /W
        AC(2,2)=  1.
        AC(6,2)= -1.
        AC(3,3)=  1.
C
        DO 55 I=1,IELDOF
        DO 55 J=1,3
          A(I,J)=0.
          DO 55 K=1,6
   55     A(I,J)=A(I,J)+AT(I,K)*AC(K,J)
C
C----- GET SPRING STIFFNESS
        FALSE= FALSE
        LSTTYP=0
        DO 58 I=1,3
        CALL MATLIB(2,LSTTYP,FALSE,ESES(I),PSES(I),DUCT(1,I),EXCR(1,I),
     &              FT,F,RT,R,0.,0.,LELEM,LMAT,MATS(I),MATTYP,
     &              SE(ISES(I)),IELNO,DAMAGE)
58      CONTINUE
C
C TO CHECK XXL,CYCLE,CO1,CO2,CO5
C
        ESE=0
        PSE=0.
C
C-- ---- SET UP THE LOCAL STIFFNESS MATRIX
        SS = SE(ISES(1))
        SA1= SE(ISES(2))
        SA2= SE(ISES(3))
C
C----- CALCULATE SAT
        SAL1=SA1/(LENGTH*2)
        SAL2=SA2/(LENGTH*2)
        SSL=SS
        DO 60 J=1,IELDOF
          SAT(1,J)=SSL*A(J,1)
          SAT(2,J)=SAL1*A(J,2)
   60   SAT(3,J)=SAL2*A(J,3)
C
C----- CALCULATE ASAT AND CONVERT TO HALF STORAGE MODE  BY COLUMNS
C
C          1  3  6 10 15 21 28 36 45 55 66 78    300.  NUMBER INDICATES
C             2  5  9 14 20 27 35 44 54 65 77      .   LOCATION OF DATA
C                4  8 13 19 26 34 43 53 64 76      .   IN ARRAY STIFF
C                   7 12 18 25 33 42 52 63 75      .
C                  11 17 24 32 41 51 62 74         .
C                     16 23 31 40 50 61 73         .
C                        22 30 39 49 60 72         .
C                           29 38 48 59 71         .
C                              37 47 58 70         .
C                                 46 57 69         .
C                                    56 68         .
C                                       67         .
C                                                  .
C                                                  .
C
        L=0
        DO 70 J=1,IELDOF
          DO 70 I=J,1,-1
          L=L+1
          STIFF(L)=0
          DO 70 K=1,3
   70     STIFF(L)=STIFF(L)+A(I,K)*SAT(K,J)
C
C- -- --  --  ------ -- ---------- ---------- ---------- ----------
C ASSEMBLE GEOMETRIC STIFFNESS MATRIX FOR A UNIT LOAD.
C ------------------------------------------ ---------- --- ------ ------
        LKG=L+1
        IF ( PKG EQ.0 .AND. KGDATA(1) NE.2 ) GO TO 180
        IF (KGDATA(1).EQ.0 .OR. KGDATA(2) EQ 0 ) GO TO 180
        AXIAL=.TRUE
C
C--- ------ DETERMINE THE AXIAL LOAD TO BE USED FOR KG...
          IF (KGDATA(1) EQ.1) THEN
              PGEOM =  PKG
          ELSE IF (KGDATA(1).EQ.2) THEN
              PGEOM = -F(2)
          ENDIF
C
C----  ASSEMBLE TRANSFORMATION MATRIX A
C NOTE. AT IS DIMMENSIONED AS A 12X6 MATRIX, BUT ONLY THE FIRST IELDOF
C       ROW'S CONTAIN INFORMATION. THE BALANCE OF THE MATRIX IS
C       UNDEFINED   THE GLOBAL DOF COORESPONDING TO THE ROW'S OF A
C       ARE STORED IN VECTOR LM   AGAIN, ONLY THE FIRST IELDOF ROW'S
C       OF LM ARE DEFINED.
        DO 140 I=1,24
        DO 140 K=1,6
  140     AT(I,K)=0.00
        KGDOF=0
        DO 145 I=1,6
        DO 145 J=1,6
C
          KGDOF =KGDOF -1
          L0=I+1
          IF (I.EQ.4) L0=I+2
          IELDOF=IELDOF+1
          LMTI=IDOF(JK(1),J)
          AT(IELDOF,L0)=AA(I,J,2)
          IF (I.EQ.1.OR.I.EQ.4) AT(IELDOF,L0-1)=AA(I,J,1)
C         LMKG(KGDOF)=LMTI
          IF((AT(KGDOF,1).EQ.0).AND.(AT(KGDOF,2).EQ.0) AND.
     &       (AT(KGDOF,3).EQ.0) AND.(AT(KGDOF,4).EQ.0).AND
     &       (AT(KGDOF,5).EQ.0).AND.(AT(KGDOF,6).EQ.0)) THEN
            KGDOF=KGDOF-1
          ELSE IF (KGDOF.GT.1) THEN
             DO 149  II=1,KGDOF-1
               IF (LMKG(II).EQ.LMKG(KGDOF)) THEN
                 DO 148 JJ=1,6
                   AT(II,JJ)=AT(II,JJ)+AT(KGDOF,JJ)
  148              AT(KGDOF,JJ)=0
                KGDOF=KGDOF-1
                GO TO 145
               ENDIF
  149        CONTINUE
          ENDIF
C
  145     CONTINUE
C
C----- LOCAL GEOMETRIC STIFFNESS MATRIX
        DO 150 I2=1,12
        DO 150 J1=1,12
  150     S(I2,J1)=0.0
          S(1,1)=  1.0/LENGTH
          S(2,2)=  0.5/LENGTH
          S(3,3)=  0.5/LENGTH
          S(4,4)=  0.5/LENGTH
          S(5,5)=  1.0/LENGTH
          S(6,6)=  0.5/LENGTH
          S(1,5)= -1.0/LENGTH
          S(2,6)= -0.5/LENGTH
          S(3,4)= -0.5/LENGTH
          S(5,1)= -1.0/LENGTH
          S(6,2)= -0.5/LENGTH
          S(4,3)= -0.5/LENGTH
C
C----- CALCULATE SAT
        DO 160 I=1,6
          DO 160 J=1,KGDOF
          SAT2(I,J)=0
          DO 160 K=1,6
  160       SAT2(I,J)=SAT2(I,J)+S(I,K)*AT(J,K)
C
C----- CALCULATE ASAT AND CONVERT TO HALF STORAGE MODE  BY COLUMNS
        DO 170 J=1,KGDOF
          DO 170 I=J,1,-1
          L=L+1
          STIFF(L)=0
          DO 170 K=1,6
  170       STIFF(L)=STIFF(L)+AT(I,K)*SAT2(K,J)
C
C ---- RELEASE UNUSED STORAGE
  180 IREL=600-L
C
C     PRINT COLUMN DATA
        IF (FIRST) WRITE(108,192) SE(ISES(1)+47)
  191 WRITE(108,193) NAME,IELNO,(MATS(I),I=1,3),(ID(JK(I)),I=1,6),
     &               LENGTH,W,ALPIN,PKG
C
  192 FORMAT(/' ELEMENT 14, R/C SHEAR WALL ELEMENT( ',
     &    'SHEAR SPAN LENGTH RATIO= ',G12.4, ')'// T17,
     &    '      ELEM',' SHEAR',' AXIAL1',' AXIAL2',' JOINT',' JOINT',
     &    ' JOINT',' JOINT',' LENGTH ',' WIDTH ',
     &    '      ALPHA ',' PKG ','/',T17,
     &    '      #',' MATL',' MATL',' MATL','    #1','    #2',
     &    '    #3','    #4'/)
C
  193 FORMAT(1X,A16,8I6,2X,1P,4G12.4)
        IOPT=IOPT
        RETURN
C---------------------- DETERMINE GEOMETRIC STIFF ----------------------
  200 CONTINUE
C- KG IS DETERMINED FOR A UNIT LOAD WITH IOPT=1,
C- THE ACTUAL LOAD (PGEOM) IS DETERMINED HERE
C- KG IS MULTIPLIED BY PGEOM WHEN THE TOTAL STIFFNESS IS ASSEMBLED
C- NEGATIVE PGEOM IS COMPRESSION....
C
        AXIAL=.TRUE.
C---------- DETERMINE THE AXIAL LOAD TO BE USED FOR KG ..
        IF (KGDATA(1).EQ.1) THEN
              PGEOM =  PKG
        ELSE IF (KGDATA(1).EQ.2) THEN
              PGEOM = -F(2)
        ENDIF
C
        RETURN
C
C---------------------- DETERMINE STIFFNESS ----------------------
  300 CONTINUE
        SS=SS
        SA1=SA1
        SA2=SA2
C
C------ SS,SA1,SA2 CONTAIN THE STIFFNESSES, WHICH WERE DETERMINED IN
C       THE LAST CALL TO MA_TLIB
C       IF SS-SE(ISES) AND SA1-SE(ISEA1) AND SA2-SE(ISEA2)
C       THEN THE STIFFNESS HAS NOT CHANGED, RETURN
C
        IF (SS.EQ.SE(ISES(1)).AND.SA1 EQ SE(ISES(2))
     &    .AND.SA2.EQ.SE(ISES(3))) RETURN
C
C---- -- SET UP THE LOCAL STIFFNESS MATRIX
        SS = SE(ISES(1))
        SA1= SE(ISES(2))
        SA2= SE(ISES(3))
        SAL1= SE(ISES(2))/(LENGTH*2)
        SAL2= SE(ISES(3))/(LENGTH*2)
        SSL = SE(ISES(1))
C
C--- - CALCULATE SAT
        DO 360 J=1,IELDOF
          SAT(1,J)=SSL*A(J,1)
          SAT(2,J)=SAL1*A(J,2)
  360     SAT(3,J)=SAL2*A(J,3)
C
C-- -- CALCULATE ASAT, STORE IN UPPER TRIANGULAR MATRIX
        L=0
        DO 370 J=1,IELDOF
          DO 370 I=J,1,-1
          L=L+1
          STIFF(L)=0
          DO 370 K=1,3
  370       STIFF(L)=STIFF(L)+A(I,K)*SAT(K,J)
C
        RETURN
C---------------------- DETERMINE INCRIMENTAL FORCES ----------------------
  400 CONTINUE
        IF (PRINT AND.FIRST) WRITE(108,491) STEPID
C
C----    SET UP THE COMPONENT STIFFNESSES
        SS = SE(ISES(1))
        SA1= SE(ISES(2))
        SA2= SE(ISES(3))
C
C------ LOOP FOR EACH LOAD
        DO 430 LOAD=1,NLOAD
C
C ----- TRANSFORM FRAME DISPL INTO LOCAL DISPL
        DO 410 I=1,3
          R(I)=0
          V(I)=0
          DO 410 J=1,IELDOF
            K=LM(J)
  410       R(I)=R(I)+A(J,I)*DISPL(K,LOAD)
```

```
C                                                                       ELE15020
C------ INCRIMENTAL FORCES                                              ELE15030
      F(2)=SA1*R(2)/(LENGTH*2)                                          ELE15040
      F(3)=SA2*R(3)/(LENGTH*2)                                          ELE15050
      F(1)=SS*R(1)                                                      ELE15060
C                                                                       ELE15070
C--- -- CHECK STIFFNESS AND CALCULATE UNBALANCED FORCES                 ELE15080
C       THIS IS PREFORMED ONLY FOR INELASTIC ANALYSIS                   ELE15090
      P(1) =PT(1)+P(1)                                                  ELE15100
      P(2) =PT(2)+P(2)                                                  ELE15110
      P(3) =PT(3)+P(3)                                                  ELE15120
      IF (.NOT. ELSTIC) THEN                                            ELE15130
C -- TRANSFER PERMANENT HYSTERESIS DATA TO TEMPROARY STORAGE...         ELE15140
       ISE2=ISE/2                                                       ELE15150
       DO 415 I=1,ISE2                                                  ELE15160
  415    SE(I+ISE2)=SE(I)                                              ELE15170
C --   SET INITIAL KKL BASED ON INITIAL DISP. INCREMENT(DSINC)          ELE15180
C --   IF INITIAL DISPL. INCREMENT IS POSITIVE, POSTPG= TRUE.           ELE15190
       IF(SE(ISE2+44).EQ.0) THEN                                        ELE15200
          IF(SE(ISE2+4).GT.0) THEN                                      ELE15210
             SE(ISE2+3)=1                                               ELE15220
             SE(ISE2+44)=1                                              ELE15230
          END IF                                                        ELE15240
          IF(SE(ISE2+4).LT.0) THEN                                      ELE15250
             SE(ISE2+3)=3                                               ELE15260
             SE(ISE2+44)=1                                              ELE15270
             SE(ISE2+37)=0                                              ELE15280
          END IF                                                        ELE15290
       END IF                                                           ELE15300
C -- SHEAR ---- ------------                                           ELE15310
C -- SET UP TEMPORARY - TOTAL LOADS, DISPL'S AND VELOC                  ELE15320
       DO 425 I=1,3                                                     ELE15330
        IF (I.EQ.1) THEN                                                ELE15340
         PL=PT(1)                                                       ELE15350
         DL=RT(1)                                                       ELE15360
         D =(RT(1)+R(1))                                                ELE15370
        ELSE                                                            ELE15380
         PL=PT(I)*2                                                     ELE15390
         P(I)=P(I)*2                                                    ELE15400
         D =(RT(I)+R(I))/LENGTH                                         ELE15410
         DL=RT(I) /LENGTH                                               ELE15420
        ENDIF                                                           ELE15430
        VC=VT(I)+V(I)                                                   ELE15440
        VL=VT(I)                                                        ELE15450
C    - CALL HYST MODEL TO GET NEW STIFFNESS AND THE TOTAL FORCE P AT DISPELE15460
C                                                                       ELE15470
       CALL MATLIB(3,LSTTYP,FALSE,ESES(I),PSES(I),DUCT(1,I),EXCR(1,I),  ELE15480
     &              P(I),PL,D,DL,VC,VL,LELEM,LMAT,MATS(I),MATTYP,      ELE15490
     &              SE(ISE2+ISES(I)),IELNO,DAMAGE)                     ELE15500
       IF (SE(ISE2+ISES(I)).NE.SE(ISES(I))) NEWK=.TRUE                  ELE15510
       IF (I.NE.1) P(I)=P(I)/2.                                         ELE15520
  425  CONTINUE                                                         ELE15530
      ENDIF                                                             ELE15540
C                                                                       ELE15550
      IF (ELSTIC) THEN                                                  ELE15560
       ESE=0                                                            ELE15570
       DO 426 I=1,3                                                     ELE15580
         ESES(I)=0.50*(PT(I)+P(I))*(RT(I)+R(I))                         ELE15590
         ESE=ESE+ESES(I)                                                ELE15600
  426  CONTINUE                                                         ELE15610
       PSE=0.0                                                          ELE15620
      ELSE                                                              ELE15630
         PSE=PSES(1)+PSES(2)+PSES(3)                                    ELE15640
         ESE=ESES(1)+ESES(2)+ESES(3)                                    ELE15650
      ENDIF                                                             ELE15660
C                                                                       ELE15670
C------ SAVE MAXIMUM LOADS                                             ELE15680
      IF (NLOAD.GT.1) THEN                                              ELE15690
        DO 389 I=1,3                                                    ELE15700
          IF (ABS(P(I)).GT.ABS(PMAX(I)) ) THEN                          ELE15710
            PMAX(I  )= P(I)                                             ELE15720
            PMAX(I+3)= R(I)                                             ELE15730
          ENDIF                                                         ELE15740
  389   CONTINUE                                                        ELE15750
      ENDIF                                                             ELE15760
C                                                                       ELE15770
C------ PRINT DATA                                                     ELE15780
      IF (PRINT) WRITE(108,392) IELNO,LOAD,(ID(JK(I)),I=1,4),           ELE15790
     &                          (P(I),R(I),I=1,3)                       ELE15800
  392 FORMAT (I10,I5,2X,4I5,1P,6G15.6)                                  ELE15810
C                                                                       ELE15820
C------ CALCULATE THE UNBALANCED MEMBER FORCE ON A JOINT.              ELE15830
      IF (LOAD.EQ.1) THEN                                               ELE15840
            UBS = -( P(1)+PT(1)-P(1) )                                  ELE15850
            UBA1 = -( P(2)+PT(2)-P(2) )                                 ELE15860
            UBA2 = -( P(3)+PT(3)-P(3) )                                 ELE15870
            DO 440 I=1,IELDOF                                           ELE15880
              K=LM(I)                                                   ELE15890
              EFUB(I)= A(I,1)*UBS + A(I,2)*UBA1 + A(I,3)*UBA2           ELE15900
  440         PUB(K)= PUB(K) + EFUB(I)                                  ELE15910
      ENDIF                                                             ELE15920
      P(1)=P(1)-PT(1)                                                   ELE15930
      P(2)=P(2)-PT(2)                                                   ELE15940
      P(3)=P(3)-PT(3)                                                   ELE15950
  430 CONTINUE                                                          ELE15960
C                                                                       ELE15970
C------ THIS IS FOR SOL05A ONLY                                        ELE15980
      IF( ICOMN .EQ. 1 .OR. ICOMN .EQ. 2 ) THEN                         ELE15990
        DO 431 I=1,3                                                    ELE16000
  431     RSAP(I)=P(I)                                                  ELE16010
      ENDIF                                                             ELE16020
C                                                                       ELE16030
  491 FORMAT (/'    R/C SHEAR WALL FORCES...',5X,A/                     ELE16040
     & '    ELEMENT LOAD',T17,' JNT-1',' JNT 2',' JNT-3',' JNT-4',      ELE16050
     & '      SHEAR      ',' SHEAR DISPL.  ',                          ELE16060
     & '      AXIAL1     ',' AXIAL DISPL. 1',                          ELE16070
     & '      AXIAL2     ',' AXIAL DISPL. 2'/)                         ELE16080
  492 FORMAT (I10,I5,2X,4I5,1P,6G15.6)                                  ELE16090
C                                                                       ELE16100
      IOPT=IOPT                                                         ELE16110
      RETURN                                                            ELE16120
C---------------------- DETERMINE TOTAL FORCES, ENERGIES ECT --------   ELE16130
  500 CONTINUE                                                          ELE16140
      IF( ICOMN .NE. 1  AND. ICOMN .NE. 2 ) THEN                        ELE16150
       IF (PRINT.AND.FIRST) WRITE(108,491) STEPID                       ELE16160
C                                                                       ELE16170
C- ---- TOTAL FORCES, DISPLACEMENTS AND VELOCITIES                      ELE16180
      DO 510 I=1,3                                                      ELE16190
      FT(I) = FT(I) + P(I)                                              ELE16200
      RT(I) = RT(I) + R(I)                                              ELE16210
      VT(I) = VT(I) + V(I)                                              ELE16220
      IF ( ABS(PT(I)).GT.ABS(PMAX(I)) ) THEN                            ELE16230
        PMAX(I  )=PT(I)                                                 ELE16240
        PMAX(I+3)=RT(I)                                                 ELE16250
      ENDIF                                                             ELE16260
  510 CONTINUE                                                          ELE16270
C                                                                       ELE16280
C------   PRINT DATA                                                   ELE16290
      IF (PRINT) WRITE(108,492) IELNO,LOAD,(ID(JK(I)),I=1,4),           ELE16300
     &                          (PT(I),RT(I),I=1,3)                     ELE16310
      ELSE                                                              ELE16320
       IF( PRINT .AND. FIRST) WRITE(108,497) STEPID                     ELE16330
        DO 513 I=1,3                                                    ELE16340
  513     PT(I)=RSAP(I)                                                 ELE16350
       IF (PRINT) WRITE(108,492) IELNO,LOAD,(ID(JK(I)),I=1,4),          ELE16360
     &                          (PT(I),I=1,3)                          ELE16370
      ENDIF                                                             ELE16380
  497 FORMAT (/'    R/C SHEAR WALL FORCES..  ',5X,A/                    ELE16390
     & '    ELEMENT LOAD',T17,' JNT-1',' JNT-2',' JNT 3',' JNT-4',     ELE16400
     & '    SHEAR      ',' AXIAL1     ',' AXIAL2     ')                ELE16410
C                                                                       ELE16420
C - - TRANSFER TEMPROARY HYSTERESIS DATA TO PERMANENT STORAGE..         ELE16430
      ISE2=ISE/2                                                        ELE16440
      IF (.NOT.ELSTIC) THEN                                             ELE16450
        DO 520 I=1,ISE2                                                 ELE16460
  520     SE(I)=SE(I+ISE2)                                              ELE16470
      ENDIF                                                             ELE16480
C                                                                       ELE16490
      RETURN                                                            ELE16500
C                                                                       ELE16510
C------------------------ DETERMINE FIXED END FORCES, AND ADD TO FORCE-ELE16520
  600 RETURN                                                            ELE16530
C                                                                       ELE16540
C------- FIXED END FORCES ARE NOT AVAILABLE FOR R/C SHEAR WALL          ELE16550
C                                                                       ELE16560
C---------------------- WRITE MEMBER FORCES TO OUTPUT FILE ------------ELE16570
  700 CONTINUE                                                          ELE16580
C                                                                       ELE16590
C...    .... SHEAR ENERGY, DUCTILITY AND EXCURSION ..                  ELE16600
      IF (ELSTIC) THEN                                                  ELE16610
        DO 703 I=1,3                                                    ELE16620
703       ESES(I)=0.50*(PT(I)+P(I))*(RT(I)+R(I))                        ELE16630
C CHANGE FOR DAMAGE                                                     ELE16640
         ESE= ESES(1)/LENGTH+ESES(2)+ESES(3)                            ELE16650
         PSE=0.0                                                        ELE16660
      ELSE                                                              ELE16670
       DO 705 I=1,3                                                     ELE16680
        CALL MATLIB(4,LSTTYP,FALSE,ESES(I),PSES(I),DUCT(1,I),EXCR(1,I), ELE16690
     &              FT,F,RT,R,0.,0.,LELEM,LMAT,MATS(I),MATTYP,          ELE16700
     &              SE(ISES(I)),IELNO,DAMAGE)                           ELE16710
705    CONTINUE                                                         ELE16720
C.......... AXIAL ENERGY, DUCTILITY AND EXCURSION (1)..                 ELE16730
C                                                                       ELE16740
         ESE=ESES(1)+ESES(2)+ESES(3)                                    ELE16750
         PSE=PSES(1)+PSES(2)+PSES(3)                                    ELE16760
      ENDIF                                                             ELE16770
C                                                                       ELE16780
                                                                        ELE16790
      IF (PRINT) WRITE (IUNIT,710) (ID(JK(I)),I=1,4)                    ELE16800
      WRITE (IUNIT,711) (FT(I),RT(I),I=1,3),ESE,PSE                     ELE16810
  710 FORMAT (4I5,1P,6E10.3/(8E10.3) )                                  ELE16820
  711 FORMAT (1P,8E10.3)                                                ELE16830
C                                                                       ELE16840
      RETURN                                                            ELE16850
C                                                                       ELE16860
C------------------- READ AND PRINT MEMBER FORCES FROM OUTPUT FILE -----ELE16870
  800 CONTINUE                                                          ELE16880
      BACKSPACE(IUNIT)                                                  ELE16890
      READ(IUNIT,*  ) ITYPE,IELNO,IWRITE,TO,DT                         ELE16900
      READ(IUNIT,710) NODE1,NODE2,NODE3,NODE4                           ELE16910
      ISTEP=-IWRITE                                                     ELE16920
C                                                                       ELE16930
  810 READ(IUNIT,815,END=830) (F(I),R(I),I=1,3),ESE,PSE                 ELE16940
  815 FORMAT (8E10.3)                                                   ELE16950
        ISTEP=ISTEP+IWRITE                                              ELE16960
        T=TO+DT*ISTEP                                                   ELE16970
        IF (ISTEP.EQ.0) WRITE(108,805) NODE1,NODE2,NODE3,NODE4          ELE16980
        IF (MOD(ISTEP,INC).EQ.0)                                        ELE16990
     &  WRITE (108,820) ISTEP,T,(F(I),R(I),I=1,3),ESE,PSE               ELE17000
        GO TO 810                                                       ELE17010
C                                                                       ELE17020
  830 CONTINUE                                                          ELE17030
C                                                                       ELE17040
  805 FORMAT (/'    R/C SHEAR WALL FORCES.. ',/                         ELE17050
     & 5X,'NODE1:',I6,5X,'NODE2:',I6,5X,'NODE3:',I6,5X,'NODE4:',I6,//   ELE17060
     & '  STEP      TIME     ','     SHEAR     ',' SHEAR DISPL  ',     ELE17070
     & '      AXIAL1    ',' AXIAL DISPL. 1',' ',     AXIAL2    ',     ELE17080
     & ' AXIAL DISPL. 2',' ESE       ',' PSE       '/)                 ELE17090
  820 FORMAT (I6,1P,G12.4,6G14.5,2G12.4)                                ELE17100
      RETURN                                                            ELE17110
C                                                                       ELE17120
C------------------- DETERMINE THE LENGTH OF THE MATERIAL DATA          ELE17130
  900 CONTINUE                                                          ELE17140
      ISE=0                                                             ELE17150
C------- LOOP FOR EACH MATEIAL -------                                  ELE17160
      DO 910 I=1,3                                                      ELE17170
       ISES(I)=ISE+1                                                    ELE17180
       MATI = RINPUT(I)                                                 ELE17190
       CALL MATLIB(0,LSTTYP,FALSE,ESE, PSE,DUCT(1,2),EXCR(1,2),         ELE17200
     &        FT,F,RT,R,0.,0.,LELEM,LMAT,MATI,MATTYP,SE,IELNO,DAMAGE)  ELE17210
  910   ISE=ISE + LELEM                                                 ELE17220
C                                                                       ELE17230
C------- DOUBLE THE STORAGE, TO ALLOW FOR TEMPROARY VALUES .            ELE17240
       ISE=ISE*2                                                        ELE17250
C                                                                       ELE17260
      RETURN                                                            ELE17270
C                                                                       ELE17280
C------------------- DETERMINE THE STRAIN ENERGY                        ELE17290
 1000 CONTINUE                                                          ELE17300
C------------------- DUCTILITEIS AND EXCURSION RATIOS                   ELE17310
 1100 CONTINUE                                                          ELE17320
C      . NOTE ELE03 WAS LAST CALLED TO CALC INCIRMENTAL DISPL,          ELE17330
C              THE ENERGIES ASSOCIATED WITH INCR DISPLR ARE STORED IN   ELE17340
C              THE ADDRESSES: ISES+ISE/2                                ELE17350
C                             ISEA1+ISE/2                               ELE17360
C                             ISEA2+ISE/2                               ELE17370
       ISE2=ISE/2                                                       ELE17380
       DO 1105 I=1,3                                                    ELE17390
C . . . . . SHEAR ENERGY, DUCTILITY AND EXCURSION . . ...               ELE17400
      . CALL MATLIB(4,LSTTYP,FALSE,ESES(I),PSES(I),DUCT(1,I),EXCR(1,I), ELE17410
     &              FT,F,RT,R,0.,0.,LELEM,LMAT,MATS(I),MATTYP,          ELE17420
     &              SE(ISES(I)+ISE2),IELNO,DAMAGE)                      ELE17430
1105   CONTINUE                                                         ELE17440
C.......... AXIAL ENERGY, DUCTILITY AND EXCURSION (1).                  ELE17450
C                                                                       ELE17460
       IF( DUCFAG .EQ. 0 .AND  ( ABS(DUCT(1,1)) .GE. 1 .OR.             ELE17470
     &     ABS(DUCT(1,2)) .GE. 1.OR. ABS(DUCT(1,3)) .GE  1 )) THEN      ELE17480
           DUCFAG=ABS(DUCT(1,1))                                        ELE17490
           IF( DUCFAG .LT. ABS(DUCT(1,2)) ) DUCFAG=ABS(DUCT(1,2))       ELE17500
           IF( DUCFAG .LT. ABS(DUCT(1,3)) ) DUCFAG=ABS(DUCT(1,3))       ELE17510
       ENDIF                                                            ELE17520
C                                                                       ELE17530
      IF (ELSTIC) THEN                                                  ELE17540
        ESE=0                                                           ELE17550
        DO 1106 I=1,3                                                   ELE17560
         ESES(I)=0.50*(PT(I)+P(I))*(RT(I)+R(I))                         ELE17570
1106      ESE= ESE+ESES(I)                                              ELE17580
         PSE=0.0                                                        ELE17590
      ELSE                                                              ELE17600
         ESE=ESES(1)+ESES(2)+ESES(3)                                    ELE17610
         PSE=PSES(1)+PSES(2)+PSES(3)                                    ELE17620
      ENDIF                                                             ELE17630
C                                                                       ELE17640
      IF (PRINT.AND.FIRST) WRITE(108,1191)                              ELE17650
      IF (PRINT) WRITE(108,1192) IELNO,((DUCT(I,J),EXCR(I,J),I=1,3)     ELE17660
     &                                  ,J=1,3)                         ELE17670
 1192 FORMAT (1X,I5,18F7.2)                                             ELE17680
 1191 FORMAT (/'    R/C SHEAR WALL DUCTILITY AND EXCURSION RATIOS'/6X,  ELE17690
     &  ' |:------- - SHEAR COMPONENT -----------:|',                    ELE17700
     &  ' | |------- - AXIAL COMPONENT 1----------:|',                  ELE17710
     &  ' | |------- - AXIAL COMPONENT 2- --- --- -:|'/,' ELEM#',       ELE17720
     &  ' | : - DEFN 1---    DEFN 2--- ---  DEFN 3---:|',               ELE17730
     &  ' | |-- DEFN 1--   --- DEFN 2--- ---  DEFN 3---:|',            ELE17740
     &  ' | |-- DEFN 1--   --- DEFN 2--- ---  DEFN 3---:|'/6X,          ELE17750
     &  '  DUCT   EXCR.  DUCT.  EXCR.  DUCT.  EXCR.',                  ELE17760
     &  '  DUCT.  EXCR.  DUCT.  EXCR   DUCT.  EXCR ',                  ELE17770
     &  '  DUCT.  EXCR.  DUCT.  EXCR.  DUCT   EXCR.')                  ELE17780
      RETURN                                                            ELE17790
C                                                                       ELE17800
C------------------------ WRITE MAXIMUM AND MINIMUM VALUES              ELE17810
 1200 CONTINUE                                                          ELE17820
      IF (FIRST) WRITE(108,491)                                         ELE17830
     &           'MAXIMUM LOADS AND DISPL AT MAXIMUM LOADS '            ELE17840
     &         //' NOTE  MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'   ELE17850
      WRITE(108,492) IELNO,LOAD,(ID(JK(I)),I=1,4),                      ELE17860
     &               (PMAX(I),PMAX(3+I),I=1,3)                          ELE17870
C                                                                       ELE17880
      RETURN                                                            ELE17890
C                                                                       ELE17900
C------------------------ RESET INTERNAL FORCES                         ELE17910
 1300 CONTINUE                                                          ELE17920
C                                                                       ELE17930
```

```
C------ ZERO MATRICIES                                                    ELE17940
      W = SE(ISES(1) + 78)                                               ELE17950
      CO1=SE(ISES(1) + 37)                                               ELE17960
C                                                                        ELE17970
      DO 1310 I=1,ISE                                                    ELE17980
 1310   SE(I)=0                                                         ELE17990
C                                                                        ELE18000
      SE(ISES(1)+77)=LENGTH                                              ELE18010
      SE(ISES(1)+78)=W                                                   ELE18020
      SE(ISES(1)+37)=CO1                                                 ELE18030
C                                                                        ELE18040
      DO 1325 I=1,2                                                      ELE18050
         FMAX(I)=0                                                       ELE18060
         FMAX(I+2)=0                                                     ELE18070
         F (I)=0                                                         ELE18080
         FT(I)=0                                                         ELE18090
         R (I)=0                                                         ELE18100
         RT(I)=0                                                         ELE18110
         V (I)=0                                                         ELE18120
 1325    VT(I)=0                                                         ELE18130
      WRITE(108,1330) IELNO                                              ELE18140
 1330 FORMAT (' SHEAR WALL  ... ELEMENT #',I6,                           ELE18150
     & ' INTERNAL FORCES AND HYSTERESIS MODELS ARE RESET TO ZERO')       ELE18160
      DO 1305 I=1,3                                                      ELE18170
        CALL MATLIB(2,LSTTYP,FALSE,ESES(I),PSES(I),DUCT(1,I),EXCR(1,I),  ELE18180
     &              FT,F,RT,R,0.,0.,LELEM,LMAT,MATS(I),MATTYP,          ELE18190
     &              SE(ISES(I)),IELNO,DAMAGE)                           ELE18200
1305  CONTINUE                                                           ELE18210
C                                                                        ELE18220
      RETURN                                                             ELE18230
C------------------- DAMAGE INDEX                                        ELE18240
 1400 IF (ELSTIC) RETURN                                                 ELE18250
C---- ..   SHEAR ENERGY, AND DAMAGE INDEX                                ELE18260
      DAMAGE=0                                                           ELE18270
      DO 1405 I=1,3                                                      ELE18280
        CALL MATLIB(5,LSTTYP,FALSE,ESES(I),PSES(I),DUCT(1,I),EXCR(1,I),  ELE18290
     &              FMAX(I),0.,FMAX(I+3),0.,0.,0.,LELEM,LMAT,MATS(I),   ELE18300
     &              MATTYP,SE(ISES(I)),IELNO,DAMS(I))                   ELE18310
        DAMAGE=DAMAGE+DAMS(I)*(ESES(I)+PSES(I))                          ELE18320
1405  CONTINUE                                                           ELE18330
C... . . .  AXIAL ENERGY, AND DAMAGE INDEX (1)                           ELE18340
C                                                                        ELE18350
      ESE=ESES(1)+ESES(2)+ESES(3)                                        ELE18360
      PSE=PSES(1)+PSES(2)+PSES(3)                                        ELE18370
      IF (FIRST) WRITE(108,1420)                                         ELE18380
      WRITE(108,1410) IELNO,(DAMS(I),ESES(I),PSES(I),I=1,3)              ELE18390
 1410 FORMAT (1X,I5,1P,3(F14.5,2G14.5))                                  ELE18400
 1420 FORMAT (/6X,                                                       ELE18410
     &  ' |:---------- SHEAR COMPONENT --------- ---|',                  ELE18420
     &  ' |:---------- AXIAL COMPONENT ------------|',/,' ELEM#',        ELE18430
     &  ' |:- DAMAGE -- ---- ESE ---- ---- PSE ---|',                   ELE18440
     &  ' |:- DAMAGE --  ---- ESE ---- ---- PSE  --|')                   ELE18450
      RETURN                                                             ELE18460
      END                                                                ELE18470
      SUBROUTINE MAT01(IOPT,FIRST,MAT,SE,P,PL,D,DL,LELEM,LMAT,           MAT00010
     &                 SMAT,EESE,EPSE,DUCT,EXCR)                       MAT00020
C                                                                        MAT00030
      IMPLICIT REAL(A-H,O-Z)                                             MAT00040
      LOGICAL FIRST                                                      MAT00050
      DIMENSION SE(15),SMAT(14),DUCT(3),EXCR(6)                          MAT00060
      CHARACTER*80 TYPE                                                  MAT00070
C                                                                        MAT00080
      IF (IOPT.EQ 0) THEN                                                MAT00090
           LELEM=14                                                      MAT00100
           RETURN                                                        MAT00110
      ENDIF                                                              MAT00120
C                                                                        MAT00130
C------------ ..... ------------------------------------------------    MAT00140
C VARIABLES:                                                             MAT00150
C------ ..... ------------------------------------------------------    MAT00160
C---------- GLOBAL VARIABLES                                            MAT00170
C IOPT    - 1, INITIALIZE MATERIAL                                       MAT00180
C         - 2, GET STIFFNESS                                             MAT00190
C SMAT    - INTERNAL STORAGE                                             MAT00200
C SE      - OUTPUT STIFFNESS                                             MAT00210
C ------------------------------------ ------------------------------    MAT00220
C                                                                        MAT00230
      GO TO (100,200,300,400) IOPT                                       MAT00240
C                                                                        MAT00250
  100 CONTINUE                                                           MAT00260
C                                                                        MAT00270
C----- INPUT DATA                                                        MAT00280
      READ (105,*) TYPE ,( SMAT(I),I=1, 8)                              MAT00290
      LELEM = 12                                                         MAT00300
      LMAT= 14                                                           MAT00310
C                                                                        MAT00320
      IF (FIRST) WRITE(108,191)                                          MAT00330
      WRITE(108,192) MAT,(SMAT(K),K=1, 8)                                MAT00340
C                                                                        MAT00350
  191 FORMAT(/'     3-D ELASTIC BEAM ELEMENT'/                           MAT00360
     &       '     ------------------------'//                          MAT00370
     & 4X,'MATL.',T14,'E' ,T20,'GAMMA',T33,'AX',T43,'AY',T53,'AZ',       MAT00380
     & T63,'IX',T73,'IY'  ,T83,'IZ'/)                                    MAT00390
  192 FORMAT (4X,I4,1P,8G10.3/)                                          MAT00400
C                                                                        MAT00410
      RETURN                                                             MAT00420
C------------------------ GET STIFFNESS TERMS ------------------------   MAT00430
  200 CONTINUE                                                           MAT00440
            E     - SMAT( 1)                                             MAT00450
            G     - SMAT( 2)                                             MAT00460
            AX    - SMAT( 3)                                             MAT00470
            AY    - SMAT( 4)                                             MAT00480
            AZ    - SMAT( 5)                                             MAT00490
           RIX    - SMAT( 6)                                             MAT00500
           RIY    - SMAT( 7)                                             MAT00510
           RIZ    - SMAT( 8)                                             MAT00520
           RKIIZ  - SMAT( 9)                                             MAT00530
           RKJJZ  - SMAT(10)                                             MAT00540
           RKIJZ  - SMAT(11)                                             MAT00550
           RKJJY  - SMAT(12)                                             MAT00560
           RKIIY  - SMAT(13)                                             MAT00570
           RKIJY  - SMAT(14)                                             MAT00580
C                                                                        MAT00590
      SE(1)=E*AX                                                         MAT00600
      SE(2)=G*AY                                                         MAT00610
      SE(3)=G*AZ                                                         MAT00620
      SE(4)=G*RIX                                                        MAT00630
      SE(5)=E*RIY                                                        MAT00640
      SE(6)=E*RIZ                                                        MAT00650
      SE( 7)=RKIIZ                                                       MAT00660
      SE( 8)=RKJJZ                                                       MAT00670
      SE( 9)=RKIJZ                                                       MAT00680
      SE(10)=RKIIY                                                       MAT00690
      SE(11)=RKJJY                                                       MAT00700
      SE(12)=RKIJY                                                       MAT00710
C                                                                        MAT00720
  300 SE(13)=0                                                           MAT00730
      SE(14)= (P-PL)/2 * (D-DL)                                          MAT00740
C                                                                        MAT00750
C------ TRANSFER ENERGIES FOR BOTH IOPT=3 AND 4                          MAT00760
  400 EESE=SE(13)                                                        MAT00770
      EPSE=SE(14)                                                        MAT00780
C                                                                        MAT00790
C--- .. TRANSFER D & P FOR BOTH IOPT=3 AND 4                             MAT00800
      DO 410  I=1,3                                                      MAT00810
      DUCT(I)=0                                                          MAT00820
      EXCR(I)=0.                                                         MAT00830
  410 EXCR(I+3)=0.                                                       MAT00840
C                                                                        MAT00850
      RETURN                                                             MAT00860
      END                                                                MAT00870
C                                                                        MAT00010
C------------------------------------------------------------------      MAT00020
C                                                                        MAT00030
      SUBROUTINE MAT02(IOPT,IELNO,FIRST,MAT,SE,P,PL,D,DL,V,VL,           MAT00040
     &                 LELEM,LMAT,SMAT,EESE,EPSE,DUCT,EXCR,DAMAGE)     MAT00050
C                                                                        MAT00060
      IMPLICIT REAL(A-H,O-Z)                                             MAT00070
      LOGICAL FIRST                                                      MAT00080
      DIMENSION SE(100),SMAT(23),DUCT(3),EXCR(6)                         MAT00090
      CHARACTER*80 TYPE                                                  MAT00100
C                                                                        MAT00110
C                                                                        MAT00120
C------------- LENGTH OF ELEMENT DATA REQUIRED...                       MAT00130
      LELEM=33                                                           MAT00140
      IF (IOPT EQ.0)   RETURN                                            MAT00150
C                                                                        MAT00160
C------------ ..... ------------------------------------------------    MAT00170
C VARIABLES:                                                             MAT00180
C--------- ..... ------------------------------------------------------ MAT00190
C---------- GLOBAL VARIABLES                                            MAT00200
C IOPT    - 1, INITIALIZE MATERIAL                                       MAT00210
C         - 2, GET STIFFNESS                                             MAT00220
C RINPUT  - INPUT DATA                                                   MAT00230
C SMAT    - INTERNAL STORAGE                                             MAT00240
C SE      - OUTPUT STIFFNESS                                             MAT00250
C---------- ------------------------------ ------ ---------------------MAT00260
C                    ---- AXLMOD DATA ----                               MAT00270
C       SMAT(1)- SC,          COMPRESSION STIFFNESS                      MAT00280
C       SMAT(2)- ST1,         INITIAL TENSILE STIFFNESS                  MAT00290
C       SMAT(3)- ST2,         POST YEILD TENSILE STIFFNESS               MAT00300
C       SMAT(4)- PY,          YEILD POINT                                MAT00310
C       SMAT(5)- ALPHA,       AXLMOD PARAMETER                           MAT00320
C       SMAT(6)- BETA,        AXLMOD PARAMETER                           MAT00330
C       SMAT(7)- DYT,         AXLMOD PARAMETER                           MAT00340
C       SMAT(8)- DYC,         AXLMOD PARAMETER                           MAT00350
C       SMAT(9)- CSE,         CONSTANT STRAIN ENERGY                     MAT00360
C       SMAT(10)=DMAX,        ULTIMATE DISPLACEMENT                      MAT00370
C       SMAT(11)=BDAM,        BETA FOR DAMAGE INDEX                      MAT00380
C                                                                        MAT00390
C---------------- INPUT DATA FOR AXLMOD HYSTERESIS MODEL ------------CMAT00400
C     THE FOLLOWING DATA IS INPUT ONCE FOR EACH MODEL                    CMAT00410
C                                                                        CMAT00420
C  SC      - SOMPRESSION STIFFNESS                                       CMAT00430
C  ST1     - INITIAL TENSILE STIFFNESS                                   CMAT00440
C  ST2     - POST YEILD TENSILE STIFFNESS                                CMAT00450
C  PY      - YEILD LOAD                                                  CMAT00460
C  ALPHA   - MODEL COIEFICIENTS                                          CMAT00470
C  BETA    - MODEL COIEFICIENTS                                          CMAT00480
C                                                                        CMAT00490
C---------- ------------------------------ ------ ---------------------CMAT00500
C                                                                        MAT00510
C                                                                        MAT00520
      GO TO (100,200,300,400,500) IOPT                                   MAT00530
C                                                                        MAT00540
  100 CONTINUE                                                           MAT00550
      READ(105,*) TYPE,( SMAT(I),I=1,6),DUCMAX,BDAM                     MAT00560
      LMAT = 11                                                          MAT00570
      SMAT(7)=  SMAT(4)/SMAT(2)                                          MAT00580
      SMAT(8)=  SMAT(4)/SMAT(1)                                          MAT00590
      SMAT(9)=  SMAT(4)**2/(2*SMAT(2))                                   MAT00600
      SMAT(10)=DUCMAX*SMAT(7)                                            MAT00610
      SMAT(11)=BDAM                                                      MAT00620
C                                                                        MAT00630
      IF (FIRST) WRITE(108,192)                                          MAT00640
      WRITE(108,191) MAT,(SMAT(K),K=1,6),DUCMAX,BDAM                     MAT00650
C                                                                        MAT00660
  191  FORMAT (1X,I5,8G15.6/)                                            MAT00670
  192  FORMAT (///'     AXLMOD HYSTERESIS MODEL- FOR UNIT LENGTH MEMBER'/MAT00680
     &       '     ------------------------------------------------'//MAT00690
     &' MAT.',6X,'  SC',10X,'  ST1',10X,'  ST2',10X,'   PY',            MAT00700
     &  10X,'ALPHA',10X,' BETA'. 6X,'MAX DUCT.',7X,'BETA-DI')           MAT00710
C                                                                        MAT00720
      RETURN                                                             MAT00730
C------------------------ GET STIFFNESS TERMS ------------------------   MAT00740
  200 CONTINUE                                                           MAT00750
C                    ---- AXLMOD MODEL DATA STORAGE FOR MEMBER ----     MAT00760
C       SE( 1)= S                                                        MAT00770
C       SE( 2)= HRN                                                      MAT00780
C       SE( 3)= FMAX                                                     MAT00790
C       SE( 4)= DMAX                                                     MAT00800
C       SE( 5)= FX                                                       MAT00810
C       SE( 6)= DX                                                       MAT00820
C       SE( 7)= FP                                                       MAT00830
C       SE( 8)= DP                                                       MAT00840
C       SE( 9)= KR                                                       MAT00850
C       SE(10)= KS                                                       MAT00860
C       SE(11)= KT                                                       MAT00870
C       SE(12)= NRL                                                      MAT00880
C       SE(13)= A                                                        MAT00890
C       SE(14)= S_EQ_UNLOAD - EQUIVALENT UNLOADING STIFFNESS             MAT00900
C       SE(15)= UPOS(1)                                                  MAT00910
C       SE(16)= UPOS(2)                                                  MAT00920
C       SE(17)= UPOS(3)                                                  MAT00930
C       SE(18)= UNEG(1)                                                  MAT00940
C       SE(19)= UNEG(2)                                                  MAT00950
C       SE(20)= UNEG(3)                                                  MAT00960
C       SE(21)= EXCR(1)                                                  MAT00970
C       SE(22)= EXCR(2)                                                  MAT00980
C       SE(23)= EXCR(3)                                                  MAT00990
C       SE(24)= EXCR(4)                                                  MAT01000
C       SE(25)= EXCR(5)                                                  MAT01010
C       SE(26)= EXCR(6)                                                  MAT01020
C       SE(27)= ESE(1)                                                   MAT01030
C       SE(28)= ESE(2)                                                   MAT01040
C       SE(29)= ESE(3)                                                   MAT01050
C       SE(30)= PSE                                                      MAT01060
C       SE(31)= PSE_OLD                                                  MAT01070
C       SE(32)= POS_DMAX                                                 MAT01080
C       SE(33)= POS_PMAX                                                 MAT01090
C                                                                        MAT01100
      DO 210 I=1,LELEM                                                   MAT01110
  210   SE(I)=0                                                          MAT01120
      CALL HYST02(P,D,PL,DL,V,VL,SE(1),SE(2),SMAT(1),SMAT(2),SMAT(3),    MAT01130
     &  SMAT(4),SMAT(5),SMAT(6),SMAT(7),SMAT(8),SMAT(9),SE(3),SE(4),     MAT01140
     &  SE(5),SE(6),SE(7),SE(8),SE(9),SE(10),SE(11),SE(12),SE(13),       MAT01150
     &  SE(14),SE(27),SE(30),SE(31),SE(15),SE(21),IDR)                   MAT01160
      RETURN                                                             MAT01170
C                                                                        MAT01180
C------------------------ GET STIFFNESS TERMS ------------------------   MAT01190
  300 CONTINUE                                                           MAT01200
      ICYC=0                                                             MAT01210
      IDR=999                                                            MAT01220
  310 STIFF=SE(1)                                                        MAT01230
      P=PL+(D-DL)*STIFF                                                  MAT01240
      A=SE(13)                                                           MAT01250
      IF (A.EQ.0 .AND  SE(2) EQ 0) A=P                                   MAT01260
      ICYC=ICYC+1                                                        MAT01270
C                                                                        MAT01280
C- ... IS P WITHIN LIMITS ?                                              MAT01290
      IF (ICYC.GT.5) THEN                                                MAT01300
         WRITE(108,*) 'MORE THAN 5 CYCLES IN MAT02,  IELNO:',IELNO       MAT01310
      ELSE IF ((PL.LE P.AND.P.LT.A).OR.(PL.GE.P.AND.P GT.A) OR.          MAT01320
     &        (DL EQ.D)) THEN                                            MAT01330
         CALL STRENG(SE(27),SE(30),SE(31),P,PL,D,DL,SE(14) )             MAT01340
         CALL HYST02(P,D,PL,DL,V,VL,SE(1),SE(2),SMAT(1),SMAT(2),SMAT(3), MAT01350
     &    SMAT(4),SMAT(5),SMAT(6),SMAT(7),SMAT(8),SMAT(9),SE(3),SE(4),    MAT01360
     &    SE(5),SE(6),SE(7),SE(8),SE(9),SE(10),SE(11),SE(12),SE(13),      MAT01370
     &    SE(14),SE(27),SE(30),SE(31),SE(15),SE(21),IDR )                 MAT01380
         V=VL                                                            MAT01390
C------ IS P GREATER THAN LIMITS ?                                       MAT01400
      ELSE IF (((PL.LE A .AND  A.LE.P)  OR. (PL GE A .AND. A GE.P)).AND  MAT01410
     &        (STIFF.NE.0.00)) THEN                                      MAT01420
         D1=DL+(A-PL)/STIFF                                              MAT01430
         P1=A                                                            MAT01440
         CALL STRENG(SE(27),SE(30),SE(31),P1,PL,D1,DL,SE(14))            MAT01450
         DL=D1                                                           MAT01460
         PL=P1                                                           MAT01470
         IF (ABS(P-PL).GT. 0.005*ABS(P-PL)) THEN                         MAT01480
           PI=PL+(P-PL)*.01                                              MAT01490
           DI=DL+(D-DL)*.01                                              MAT01500
         ELSE                                                            MAT01510
```

```
          PI=P                                                          MAT01520
          DI=D                                                          MAT01530
        ENDIF                                                           MAT01540
        CALL HYST02(PI,DI,PL,DL,1.,1.,SE(1),SE(2),SMAT(1),SMAT(2),     MAT01550
     &      SMAT(3),SMAT(4),SMAT(5),SMAT(6),SMAT(7),SMAT(8),SMAT(9),   MAT01560
     &      SE(3),SE(4),SE(5),SE(6),SE(7),SE(8),SE(9),SE(10),SE(11),    MAT01570
     &      SE(12),SE(13),SE(14),SE(27),SE(30),SE(31),SE(15),           MAT01580
     &      SE(21),IDR )                                                MAT01590
        V=VL                                                            MAT01600
        GO TO 310                                                       MAT01610
C--- - IS P LESS THAN LIMITS ?                                          MAT01620
      ELSE IF ((P LT PL  AND. PL.LE.A) .OR. (P.GT.PL  AND. PL.GE.A)}THENMAT01630
            IF (ABS(P-PL).GT. 0.005*ABS(P-PL)} THEN                     MAT01640
                PI=PL+(P-PL)*.01                                        MAT01650
                DI=DL+(D-DL)*.01                                        MAT01660
            ELSE                                                        MAT01670
                PI=P                                                    MAT01680
                DI=D                                                    MAT01690
            ENDIF                                                       MAT01700
            CALL HYST02(PI,DI,PL,DL,1.,1.,SE(1),SE(2),SMAT(1),SMAT(2),  MAT01710
     &      SMAT(3),SMAT(4),SMAT(5),SMAT(6),SMAT(7),SMAT(8),SMAT(9),    MAT01720
     &      SE(3),SE(4),SE(5),SE(6),SE(7),SE(8),SE(9),SE(10),SE(11),    MAT01730
     &      SE(12),SE(13),SE(14),SE(27),SE(30),SE(31),SE(15),           MAT01740
     &      SE(21),IDR )                                                MAT01750
            V=VL                                                        MAT01760
            GO TO 310                                                   MAT01770
      ENDIF                                                             MAT01780
C                                                                       MAT01790
C------ SAVE MAXIMUM PEAK POSITIVE DISPL                                MAT01800
      IF (D GT.SE(32)) THEN                                             MAT01810
          SE(32)=D                                                      MAT01820
          SE(33)=P                                                      MAT01830
      ENDIF                                                             MAT01840
C                                                                       MAT01850
C------ TRANSFER ENERGIES FOR BOTH IOPT=3 AND 4.                        MAT01860
  400 ESE=SE(27)                                                        MAT01870
      EPSE=SE(30)                                                       MAT01880
C                                                                       MAT01890
C------  TRANSFER DUCTILITIES AND EXCRUSION RATIOS FOR BOTH IOPT=3 AND 4.MAT01900
      DO 410 I=1,3                                                      MAT01910
      DUCT(I)=MAX(SE(14+I),SE(17+I))                                    MAT01920
      EXCR(I)=SE(20+I)                                                  MAT01930
  410 EXCR(I+3)=SE(20+I+3)                                              MAT01940
C                                                                       MAT01950
      RETURN                                                            MAT01960
C                                                                       MAT01970
C--------------------- DAMAGE INDEX ------------------------------------MAT01980
  500 CONTINUE                                                          MAT01990
      ESE=SE(27)                                                        MAT02000
      EPSE=SE(30)                                                       MAT02010
      BDAM=SMAT(11)                                                     MAT02020
      D =SE(32)                                                         MAT02030
      P =SE(33)                                                         MAT02040
      DAMAGE=ABS(D)/SMAT(10) + BDAM /( SMAT(4)*SMAT(10)) * EPSE         MAT02050
C                                                                       MAT02060
      RETURN                                                            MAT02070
      END                                                               MAT02080
C                                                                       MAT00010
C-----------------------------------------------------------------------MAT00020
C                                                                       MAT00030
      SUBROUTINE MAT03(IOPT,IELNO,FIRST,MAT,SE,P,PL,D,DL,V,LELEM,       MAT00040
     &                 LMAT,SMAT,ESSE,EPSE,DUCT,EXCR,DAMAGE,NSOLN)      MAT00050
C                                                                       MAT00060
      IMPLICIT REAL(A-H,O-Z)                                            MAT00070
      LOGICAL FIRST                                                     MAT00080
      DIMENSION SE(100),SMAT(30),DUCT(3),EXCR(6)                        MAT00090
      CHARACTER*80 TYPE                                                 MAT00100
C                                                                       MAT00110
      IF (IOPT.NE.1) LELEM=34+3*SMAT(2)*2                               MAT00120
      IF (IOPT.EQ.0)  RETURN                                            MAT00130
C                                                                       MAT00140
C-----------------------------------------------------------------------MAT00150
C VARIABLES:                                                            MAT00160
C-----------------------------------------------------------------------MAT00170
C---------- GLOBAL VARIABLES                                           MAT00180
C IOPT   = 1, INITIALIZE MATERIAL                                       MAT00190
C        = 2, GET STIFFNESS                                             MAT00200
C SMAT   = INTERNAL STORAGE                                             MAT00210
C SE     = OUTPUT STIFFNESS                                             MAT00220
C---------- ---------- ---------- ---------- ---------- ----------     MAT00230
C          ---- BEND1 DATA - --                                         MAT00240
C        SMAT(1)= NSEGS,          NUMBER OF BACKBONE SEGMENTS           MAT00250
C        SMAT(2)= NI,             NUMBER OF SMALL AMPLITUDE LOOPS       MAT00260
C        SMAT(3)= CSE,            CONSTANT STRAIN ENERGY.               MAT00270
C        SMAT(4)= DY,             YEILD POINT FOR DUCTILITIES           MAT00280
C        SMAT(5          )= BDAM  BETA FOR DAMAGE INDEX                 MAT00290
C        SMAT(5+I        )= PP,   BACKBONE MOMENT FOR POINT I           MAT00300
C        SMAT(5+I+NSEGS)= DP,     BACKBONE ROTATION FOR POINT I         MAT00310
C                                                                       MAT00320
      GO TO (100,200,300,400,500) IOPT                                  MAT00330
C                                                                       MAT00340
C                                                                       MAT00350
  100 CONTINUE                                                          MAT00360
C------ ---------- INPUT DATA FOR BEND1  HYSTERESIS MODEL ----------CMAT00370
C                                                                     CMAT00380
C  NSEG    = NUMBER OF BACKBONE SEGEMENTS                             CMAT00390
C  NI      = NUMBER OF SMALL LOOP POINTS                              CMAT00400
C  EI      = ELASTIC STIFFNESS                                        CMAT00410
C  G       = MOMENT GRADIENT                                          CMAT00420
C  PP      = BACKBONE LOAD                                            CMAT00430
C  DP      = BACKBONE DISPLACEMENT                                    CMAT00440
C                                                                     CMAT00450
C---------- ---------- ---------- ---------- ---------- ----------   CMAT00460
      READ(105,*) TYPE,NSEG,NI,DY,BDAM                                  MAT00470
      READ(105,*) (SMAT(5+I      ),I=1,NSEG)                            MAT00480
      READ(105,*) (SMAT(5+I+NSEG),I=1,NSEG)                             MAT00490
C                                                                       MAT00500
      LELEM = 34+3*NI +2                                                MAT00510
      LMAT = 5+2*NSEG                                                   MAT00520
      SMAT(1)=NSEG                                                      MAT00530
      SMAT(2)=NI                                                        MAT00540
      SMAT(4)=DY                                                        MAT00550
      SMAT(5)=BDAM                                                      MAT00560
C------------- CALC CONSTANT STRAIN ENERGY....                          MAT00570
      DL=0                                                              MAT00580
      PL=0                                                              MAT00590
      DO 10 I=1,NSEG                                                    MAT00600
            P=SMAT(5+I)                                                 MAT00610
            D=SMAT(5+I+NSEG)                                            MAT00620
            IF (D LE DY) THEN                                           MAT00630
              CSE=CSE + 0.5*(P+PL)*(D-DL)                               MAT00640
              PL=P                                                      MAT00650
              DL=D                                                      MAT00660
            ELSE IF (D EQ.DL) THEN                                      MAT00670
              GO TO 20                                                  MAT00680
            ELSE                                                        MAT00690
              PY=PL+ (DY-DL)*(P-PL)/(D DL)                              MAT00700
              CSE=CSE + 0.5*(PY+PL)*(DY-DL)                             MAT00710
              GO TO 20                                                  MAT00720
            ENDIF                                                       MAT00730
   10 CONTINUE                                                          MAT00740
   20 SMAT(3)=CSE                                                       MAT00750
C                                                                       MAT00760
      IF (FIRST) WRITE(108,55)                                          MAT00770
      WRITE(108,57) MAT,NI,DY,BDAM,(SMAT(5-K),SMAT(5-K+NSEG),K=1,NSEG)  MAT00780
      WRITE(108,*) ' '                                                  MAT00790
C                                                                       MAT00800
   55   FORMAT(/'     BEND1 HYSTERESIS MODEL DATA - UNIT LENGTH MEMBER'  MAT00810
     &       /'     ------------------------------------------',       MAT00820
     &       /'     BACKBONE CURVE POINTS - MATL  NI ',5X,'DY',9X,      MAT00830
     &       /'       BETA          ',3X,' MOMENT',8X,'ROTATION')       MAT00840
   57   FORMAT      ( (28X,2I4,2X,G15.6,3G15.6)/(68X,2G15.6))           MAT00850
C                                                                       MAT00860
      RETURN                                                            MAT00870
C---------------------- GET STIFFNESS TERMS ----------------------------MAT00880
  200 CONTINUE                                                          MAT00890
C          ---- BEND1 MODEL DATA STORAGE FOR MEMBER - --               MAT00900
C            SE( 1)= K                                                 MAT00910
C            SE( 2)= RULE                                              MAT00920
C            SE( 3)= DIR                                               MAT00930
C            SE( 4)= A                                                 MAT00940
C            SE( 5)= PMG                                               MAT00950
C            SE( 6)= PMH                                               MAT00960
C            SE( 7)= DMG                                               MAT00970
C            SE( 8)= DMH                                               MAT00980
C            SE( 9)= BACKB                                             MAT00990
C            SE(10)= ALPHA1                                            MAT01000
C            SE(11)= ALPHA2                                            MAT01010
C            SE(12)= LR                                                MAT01020
C            SE(13)= SR1                                               MAT01030
C            SE(14)= SR2                                               MAT01040
C            SE(15)= SR3                                               MAT01050
C            SE(16)= SRM                                               MAT01060
C            SE(17)= E_EQ_UNLOAD                                       MAT01070
C            SE(18)= UPOS(1)                                           MAT01080
C            SE(19)= UPOS(2)                                           MAT01090
C            SE(20)= UPOS(3)                                           MAT01100
C            SE(21)= UNEG(1)                                           MAT01110
C            SE(22)= UNEG(2)                                           MAT01120
C            SE(23)= UNEG(3)                                           MAT01130
C            SE(24)= EXCR(1)                                           MAT01140
C            SE(25)= EXCR(2)                                           MAT01150
C            SE(26)= EXCR(3)                                           MAT01160
C            SE(27)= EXCR(4)                                           MAT01170
C            SE(28)= EXCR(5)                                           MAT01180
C            SE(29)= EXCR(6)                                           MAT01190
C            SE(30)= ESE(1)                                            MAT01200
C            SE(31)= ESE(2)                                            MAT01210
C            SE(32)= ESE(3)                                            MAT01220
C            SE(33)= ESE                                               MAT01230
C            SE(34)= PSE_OLD                                           MAT01240
C            SE(34+1      )=PR( 1)                                     MAT01250
C            SE(34+    NI)=PR(NI)                                      MAT01260
C            SE(34+1+  NI)=DR( 1)                                      MAT01270
C            SE(34+  2*NI)=DR(NI)                                      MAT01280
C            SE(34+1+2*NI)=FR( 1)                                      MAT01290
C            SE(34+  3*NI)=FR(NI)                                      MAT01300
C            SE(35+  3*NI)=P_MAX                                       MAT01310
C            SE(36+  3*NI)=D_MAX                                       MAT01320
C                                                                      MAT01330
      NSEG=SMAT(1)                                                      MAT01340
      NI  =SMAT(2)                                                      MAT01350
      J1=  NSEG-6                                                       MAT01360
      J3=  NI+35                                                        MAT01370
      J4=2*NI+35                                                        MAT01380
      J5=3*NI+35-1 +2                                                   MAT01390
      SI=SMAT(6)/SMAT(6+NSEG)                                           MAT01400
      DO 210 I=1,J5                                                     MAT01410
  210 SE(I)=0                                                           MAT01420
      CALL HYST03(P,D,PL,DL,V,NSEG,NI,SMAT(6),SMAT(J1),SMAT(6),         MAT01430
     &   SMAT(J1),SI,SMAT(3),SE(1),SE(2),SE(3),SE(4),SE(5),SE(6),SE(7), MAT01440
     &   SE(8),SE(9),SE(10),SE(11),SE(12),SE(35),SE(J3),SE(J4),         MAT01450
     &   SE(17),SMAT(4),SE(30) ,SE(33),SE(34),SE(18),SE(21),SE(24))     MAT01460
      RETURN                                                            MAT01470
C                                                                       MAT01480
C------------------------ GET STIFFNESS TERMS ----------------------------MAT01490
  300 CONTINUE                                                          MAT01500
C     SE( 4)= A                                                         MAT01510
      NSEG=SMAT(1)                                                      MAT01520
C .... MODIFIED TO SATISFIED THE PORTION OF DROPING DOWN OF BACKBONE     MAT01530
C .... CURVE AFTER THE LAST POINT (I.E., THE NSEG'S POINT)              MAT01540
C .... ON JULY 8, 1994 BASED ON YANG'S WORK                             MAT01550
      IF (NSOLN.EQ.4) THEN                                              MAT01560
       IF (SE(1).EQ.0.0.AND.PL.EQ.0.0) THEN                             MAT01570
        P=0.                                                            MAT01580
        SE(1)=0.                                                        MAT01590
        RETURN                                                          MAT01600
       ENDIF                                                            MAT01610
       JJJ1=5+2*NSEG                                                    MAT01620
       IF (ABS(D).GT.SMAT(JJJ1).AND.ABS(DL).LT.SMAT(JJJ1)) THEN         MAT01630
        SMATJJ=SIGN(SMAT(JJJ1),DL)                                      MAT01640
        P1=PL+(SMATJJ-DL)*SE(1)                                         MAT01650
        D1=SMATJJ                                                       MAT01660
        CALL STRENG(SE(30),SE(33),SE(34),P1,PL,D1,DL,SE(17))            MAT01670
C ...  AFTER REACHING FAILURE DUCTILITY, ENFORCE P=0.                   MAT01680
        P=0                                                             MAT01690
        D=SIGN(SMAT(JJJ1),D)                                            MAT01700
        DDUCT=ABS(D)/SMAT(JJJ1)                                         MAT01710
        RETURN                                                          MAT01720
       ENDIF                                                            MAT01730
       IF (ABS(DL).GT.SMAT(JJJ1).AND.ABS(D).GT.SMAT(JJJ1)) THEN         MAT01740
C ... TO ENFORCE THE FORCE=0.0 AFTER FAILURE DUCTILITY                  MAT01750
        P=0.D0                                                          MAT01760
        RETURN                                                          MAT01770
       ENDIF                                                            MAT01780
      ENDIF                                                             MAT01790
C .... MODIFICATION END                                                 MAT01800
      NI  =SMAT(2)                                                      MAT01810
      J1=  NSEG-6                                                       MAT01820
      J3=  NI+35                                                        MAT01830
      J4=2*NI+35                                                        MAT01840
      SI=SMAT(6)/SMAT(6+NSEG)                                           MAT01850
C     SET VELOCITY FLAG TO 1 TO REMOVE INFL OF HIGH FREQ VELOC          MAT01860
      DVEL = 1                                                          MAT01870
      ICYC=0                                                            MAT01880
  310 STIFF=SE(1)                                                       MAT01890
      P=PL+(D-DL)*STIFF                                                 MAT01900
      A=SE(4)                                                           MAT01910
      IF (A.EQ.0 .AND. SE(2).EQ.0) THEN                                 MAT01920
            A=SIGN(SMAT(6),P)                                           MAT01930
            IF (P.LT.PL .AND. PL.GT.0 .AND. P GT.0) A=0                 MAT01940
            IF (P.GT.PL .AND. PL.LT.0 .AND  P.LT.0) A=0                 MAT01950
      ENDIF                                                             MAT01960
      ICYC=ICYC+1                                                       MAT01970
C                                                                       MAT01980
C-----  IS P WITHIN LIMITS ?                                            MAT01990
      IF (ICYC GT.10) THEN                                              MAT02000
            WRITE(108,*) 'MORE THAN 10 CYCLES IN MAT03,  IELNO:',IELNO  MAT02010
      ELSE IF ((PL.LE.P.AND.P.LT.A).OR.(PL.GE.P.AND.P.GT.A).OR          MAT02020
     &         (DL.EQ.D)) THEN                                          MAT02030
            CALL STRENG (SE(30),SE(33),SE(34),P,PL,D,DL,SE(17) )        MAT02040
            CALL HYST03(P,D,PL,DL,DVEL,NSEG,NI,SMAT(6),SMAT(J1),SMAT(6),MAT02050
     &   SMAT(J1),SI,SMAT(3),SE(1),SE(2),SE(3),SE(4),SE(5),SE(6),SE(7), MAT02060
     &   SE(8),SE(9),SE(10),SE(11),SE(12),SE(35),SE(J3),SE(J4),         MAT02070
     &   SE(17),SMAT(4),SE(30) ,SE(33),SE(34),SE(18),SE(21),SE(24))     MAT02080
            DVEL=1                                                      MAT02090
C ... - IS P GREATER THAN LIMITS ?                                      MAT02100
      ELSE IF (((PL.LE.A .AND. A.LE.P) .OR. (PL.GE.A .AND. A GE.P)).AND.MAT02110
     &         (STIFF NE 0.00)) THEN                                    MAT02120
            D1=DL+(A-PL)/STIFF                                          MAT02130
            P1=A                                                        MAT02140
            CALL STRENG (SE(30),SE(33),SE(34),P1,PL,D1,DL,SE(17) )      MAT02150
            DL=D1                                                       MAT02160
            PL=P1                                                       MAT02170
            IF (ABS(P-PL) GT  0.0005*ABS(P+PL)) THEN                    MAT02180
                  P1=PL+(P-PL)*.001                                     MAT02190
                  D1=DL+(D-DL)*.001                                     MAT02200
            ELSE                                                        MAT02210
                  P1=P                                                  MAT02220
                  D1=D                                                  MAT02230
            ENDIF                                                       MAT02240
            CALL HYST03(P1,D1,PL,DL,1.,NSEG,NI,SMAT(6),SMAT(J1),SMAT(6),MAT02250
     &   SMAT(J1),SI,SMAT(3),SE(1),SE(2),SE(3),SE(4),SE(5),SE(6),SE(7), MAT02260
     &   SE(8),SE(9),SE(10),SE(11),SE(12),SE(35),SE(J3),SE(J4),         MAT02270
     &   SE(17),SMAT(4),SE(30) ,SE(33),SE(34),SE(18),SE(21),SE(24))     MAT02280
            GO TO 310                                                   MAT02290
C--- - IS P LESS THAN LIMITS ?                                          MAT02300
      ELSE IF ((P.LT.PL .AND  PL.LE.A) .OR. (P.GT.PL .AND  PL.GE A))THENMAT02310
            IF (ABS(P-PL).GT  0.0005*ABS(P+PL)) THEN                    MAT02320
                  P1=PL+(P-PL)* 001                                     MAT02330
                  D1=DL+(D-DL)*.001                                     MAT02340
            ELSE                                                        MAT02350
```

```
                PI=P                                                    MAT02360
                DI=D                                                    MAT02370
              ENDIF                                                     MAT02380
              CALL HYST03(PI,D1,PL,DL,1.,NSEG,NI,SMAT(6),SMAT(J1),SMAT(6),MAT02390
     &     SMAT(J1),SI,SMAT(3),SE(1),SE(2),SE(3),SE(4),SE(5),SE(6),SE(7),MAT02400
     &     SE(8),SE(9),SE(10),SE(11),SE(12),SE(35),SE(J3),SE(J4),        MAT02410
     &     SE(17),SMAT(4),SE(30) ,SE(33),SE(34),SE(18),SE(21),SE(24))    MAT02420
              GO TO 310                                                 MAT02430
          ENDIF                                                         MAT02440
C                                                                       MAT02450
C-------- SAVE MAXIMUM DISPL AND IT'S LOAD                             MAT02460
          IF (ABS(D).GT.ABS(SE(36+3*NI))) THEN                          MAT02470
                SE(35+ 3*NI)=P                                          MAT02480
                SE(36+ 3*NI)=D                                          MAT02490
          ENDIF                                                         MAT02500
C                                                                       MAT02510
C------ TRANSFER ENERGIES FOR BOTH IOPT=3 AND 4.                        MAT02520
  400 EESE=SE(30)                                                       MAT02530
          EPSE=SE(33)                                                   MAT02540
C                                                                       MAT02550
C------ TRANSFER DUCTILITIES AND EXCRUSION RATIOS FOR BOTH IOPT=3 AND 4.MAT02560
          DO 410 I=1,3                                                  MAT02570
          DUCT(I)=MAX(SE(17+I),SE(20+I))                                MAT02580
          EXCR(I+3)=SE(23+I+3)                                          MAT02590
  410 EXCR(I)=SE(23+I)                                                  MAT02600
C                                                                       MAT02610
          RETURN                                                        MAT02620
C                                                                       MAT02630
C------------------------- DAMAGE INDEX ---------------------------     MAT02640
  500 EESE =SE(30)                                                      MAT02650
          EPSE =SE(33)                                                  MAT02660
          NSEGS=SMAT(1)                                                 MAT02670
          NI   =SMAT(2)                                                 MAT02680
          DMAX =SMAT(5+2*NSEGS)                                         MAT02690
          DY   =SMAT(4)                                                 MAT02700
          BDAM =SMAT(5)                                                 MAT02710
          P    =SE(35+ 3*NI)                                            MAT02720
          D    =SE(36+ 3*NI)                                            MAT02730
                                                                        MAT02740
          D1   =SMAT(6+NSEGS)                                           MAT02750
          IF (DY.LE.D1) THEN                                            MAT02760
              PY=SMAT(6)*DY/D1                                          MAT02770
          ELSE                                                          MAT02780
              DO 510 I=2,NSEGS                                          MAT02790
                P1=SMAT(4+I)                                            MAT02800
                P2=SMAT(5+I)                                            MAT02810
                D1=SMAT(4+I+NSEGS)                                      MAT02820
                D2=SMAT(5+I+NSEGS)                                      MAT02830
                IF (DY.GE.D1 .AND. DY.LE D2)  THEN                      MAT02840
                  PY=P1+(DY-D1)*(P2-P1)/(D2-D1)                         MAT02850
                  GO TO 520                                             MAT02860
                ENDIF                                                   MAT02870
  510           CONTINUE                                                MAT02880
          ENDIF                                                         MAT02890
  520 DAMAGE=ABS(D)/DMAX + BDAM / (PY*DMAX) * EPSE                      MAT02900
C                                                                       MAT02910
          RETURN                                                        MAT02920
          END                                                           MAT02930
C                                                                       MAT02940
C-----------------------------------------------------------------------MAT02950
C                                                                       MAT03030
          SUBROUTINE MAT04(IOPT,IELNO,FIRST,MAT,SE,P,PL,D,DL,V,VL,      MAT00040
     &           LELEM,LMAT,SMAT,EESE,EPSE,DUCT,EXCR,DAMAGE,NSOLN)      MAT00050
C                                                                       MAT00060
          LOGICAL FIRST                                                 MAT00070
          DIMENSION SE(100),SMAT(20),DUCT(3),EXCR(6)                    MAT00080
          CHARACTER*80 TYPE                                             MAT00090
C                                                                       MAT00100
          IF (IOPT.NE.1) LELEM=32+3*SMAT(2)  +2                         MAT00110
          IF (IOPT.EQ.0) RETURN                                         MAT00120
C                                                                       MAT00130
C.. ..................  ..................................            MAT00140
C VARIABLES:                                                            MAT00150
C-------------------------------------------------------------------    MAT00160
C---------- GLOBAL VARIABLES                                           MAT00170
C IOPT   - 1, INITIALIZE MATERIAL                                       MAT00180
C        - 2, GET STIFFNESS                                             MAT00190
C RINPUT - INPUT DATA                                                   MAT00200
C SMAT   - INTERNAL STORAGE                                             MAT00210
C SE     - OUTPUT STIFFNESS                                             MAT00220
C-------------------------------------------------------------------    MAT00230
C           ---- SHEAR1 DATA ----                                       MAT00240
C            SMAT(1)= NSEGS,          NUMBER OF BACKBONE SEGMENTS       MAT00250
C            SMAT(2)= NI,             NUMBER OF SMALL AMPLITUDE LOOPS   MAT00260
C            SMAT(3)= CSE,            CONSTAINT STRAIN ENERGY...        MAT00270
C            SMAT(4)= DY,             YEILD DISPLACEMENT                MAT00280
C            SMAT(4+I)= PP,           BACKBONE SHEAR FOR POINT I        MAT00290
C            SMAT(4+I+NSEG)= DP,      BACKBONE DISPLACEMENT FOR POINT I MAT00300
C            SMAT(5+2*NSEG)= BDAM     BETA FOR DAMAGE INDEX             MAT00310
C                                                                       MAT00320
          GO TO (100,200,300,400,500) IOPT                              MAT00330
C                                                                       MAT00340
  100 CONTINUE                                                          MAT00350
C                                                                       MAT00360
C------------------ INPUT DATA FOR SHEAR1 HYSTERESIS MODEL ------------CMAT00370
C                                                                       CMAT00380
C  NSEG    - NUMBER OF BACKBONE SEGEMENTS                               CMAT00390
C  NI      - NUMBER OF SMALL LOOP POINTS                                CMAT00400
C  PP      - BACKBONE LOAD                                              CMAT00410
C  DP      - BACKBONE DISPLACEMENT                                      CMAT00420
C-------------------------------------------------------------------    CMAT00430
          READ(105,*) TYPE ,NSEG,NI,DY,BDAM                             MAT00440
          READ(105,*) ( SMAT(4+I     ),I=1,NSEG)                        MAT00450
          READ(105,*) ( SMAT(4+I+NSEG),I=1,NSEG)                        MAT00460
          LELEM = 32+3*NI +2                                            MAT00470
          LMAT = 5+2*NSEG                                               MAT00480
          SMAT(1)=NSEG                                                  MAT00490
          SMAT(2)=NI                                                    MAT00500
          SMAT(4)=DY                                                    MAT00510
          SMAT(5+2*NSEG)=BDAM                                           MAT00520
C-------------- CALC CONSTANT STRAIN ENERGY . .                         MAT00530
          DL=0                                                          MAT00540
          PL=0                                                          MAT00550
          DO 10 I=1,NSEG                                                MAT00560
                P=SMAT(4+I)                                             MAT00570
                D=SMAT(4+I+NSEG)                                        MAT00580
                IF (D.LE.DY) THEN                                       MAT00590
                  CSE=CSE + 0.5*(P+PL)*(D-DL)                           MAT00600
                  PL=P                                                  MAT00610
                  DL=D                                                  MAT00620
                ELSE IF (D.EQ DL) THEN                                  MAT00630
                  GO TO 20                                              MAT00640
                ELSE                                                    MAT00650
                  PY=PL+ (DY-DL)*(P PL)/(D-DL)                          MAT00660
                  CSE=CSE + 0.5*(PY+PL)*(DY-DL)                         MAT00670
                  GO TO 20                                              MAT00680
                ENDIF                                                   MAT00690
   10 CONTINUE                                                          MAT00700
   20 SMAT(3)=CSE                                                       MAT00710
C                                                                       MAT00720
          IF (FIRST) WRITE(108,65)                                      MAT00730
          WRITE(108,67) MAT,NI,DY,BDAM,(SMAT(4+X),SMAT(4+X+NSEG),X=1,NSEG)  MAT00740
          WRITE(108,*)  '  '                                            MAT00750
   65   FORMAT(////'     SHEAR1 HYSTERESIS MODEL DATA - UNIT LENGTH MEMBER'MAT00760
     &    /   '     ----------------------------------------------------',MAT00770
     &    /'     BACKBONE CURVE POINTS - MAT.  NI          DY      ',    MAT00780
     &   5X,'BETA',6X,    3X,' STRESS',8X,' STRAIN')                     MAT00790
   67   FORMAT      ( (28X,2I4,3X,4G15.6)/(68X,2G15.6))                  MAT00800
C                                                                       MAT00810
          RETURN                                                        MAT00820
C------------------------- GET STIFFNESS TERMS ------------------------MAT00830
  200 CONTINUE                                                          MAT00840
C           ---- SHEAR1 MODEL DATA STORAGE FOR MEMBER ----              MAT00850
C       SE( 1)= X                                                       MAT00860
C       SE( 2)= RULE                                                    MAT00870
C       SE( 3)= DIR                                                     MAT00880
C       SE( 4)= A                                                       MAT00890
C       SE( 5)= PMG                                                     MAT00900
C       SE( 6)= PMH                                                     MAT00910
C       SE( 7)= DMG                                                     MAT00920
C       SE( 8)= DMH                                                     MAT00930
C       SE( 9)= BACKB                                                   MAT00940
C       SE(10)= SR1                                                     MAT00950
C       SE(11)= SR2                                                     MAT00960
C       SE(12)= SR3                                                     MAT00970
C       SE(13)= SRM                                                     MAT00980
C       SE(14)= IR                                                      MAT00990
C       SE(15)= S_EQ_UNLOAD                                             MAT01000
C       SE(16)= UPOS(1)                                                 MAT01010
C       SE(17)= UPOS(2)                                                 MAT01020
C       SE(18)= UPOS(3)                                                 MAT01030
C       SE(19)= UNEG(1)                                                 MAT01040
C       SE(20)= UNEG(2)                                                 MAT01050
C       SE(21)= UNEG(3)                                                 MAT01060
C       SE(22)= EXCR(1)                                                 MAT01070
C       SE(23)= EXCR(2)                                                 MAT01080
C       SE(24)= EXCR(3)                                                 MAT01090
C       SE(25)= EXCR(4)                                                 MAT01100
C       SE(26)= EXCR(5)                                                 MAT01110
C       SE(27)= EXCR(6)                                                 MAT01120
C       SE(28)= ESE                                                     MAT01130
C       SE(29)= ESE                                                     MAT01140
C       SE(30)= ESE                                                     MAT01150
C       SE(31)= PSE                                                     MAT01160
C       SE(32)= PSE_OLD                                                 MAT01170
C       SE(32+1      )=PR( 1)                                           MAT01180
C       SE(32+   NI)=PR(NI)                                             MAT01190
C       SE(32+1+ NI)=DR( 1)                                             MAT01200
C       SE(32+  2*NI)=DR(NI)                                            MAT01210
C       SE(32+1+2*NI)=FR( 1)                                            MAT01220
C       SE(32+  3*NI)=FR(NI)                                            MAT01230
C       SE(33+  3*NI)=P_MAX                                             MAT01240
C       SE(34+  3*NI)=D_MAX                                             MAT01250
C                                                                       MAT01260
          NSEG=SMAT(1)                                                  MAT01270
          NI  =SMAT(2)                                                  MAT01280
          SI  = SMAT(5)/SMAT(5+NSEG)                                    MAT01290
          J1=NSEG+5                                                     MAT01300
          J3=NI+33                                                      MAT01310
          J4=2*NI+33                                                    MAT01320
          J5=3*NI+33-1.+2                                               MAT01330
          DO 210 I=1,J5                                                 MAT01340
  210   SE(I)=0                                                         MAT01350
              CALL HYST04( P, D,PL,DL, V  ,NSEG,NI,                     MAT01360
     &           SMAT( 5),SMAT(J1),SMAT( 5),SMAT(J1),SI,SMAT(3),SMAT(4),MAT01370
     &           SE( 1)  ,SE( 2)  ,SE( 3)  ,SE( 4)  ,SE( 5) ,SE( 6) ,  MAT01380
     &           SE( 7)  ,SE( 8)  ,SE( 9)  ,SE(10)  ,SE(11) ,SE(12) ,  MAT01390
     &           SE(13)  ,SE(14)  ,SE(15)  ,SE(16)  ,SE(19) ,SE(22) ,  MAT01400
     &           SE(28)  ,SE(31)  ,SE(32)  ,SE(33)  ,SE(J3) ,SE(J4) )  MAT01410
          RETURN                                                        MAT01420
C                                                                       MAT01430
C------------------------- GET STIFFNESS TERMS ------------------------MAT01440
  300 CONTINUE                                                          MAT01450
          NSEG=SMAT(1)                                                  MAT01460
C ... MODIFIED TO SATISFIED THE PORTION OF DROPING DOWN OF BACKBONE      MAT01470
C ... CURVE AFTER THE LAST POINT (I.E , THE NSEG'S POINT)               MAT01480
C ... ON JULY 8, 1994 BASED ON YANG'S WORK                              MAT01490
          IF (NSOLN.EQ.4) THEN                                          MAT01500
           IF (SE(1).EQ.0.0.AND.PL.EQ.0.0) THEN                         MAT01510
            P=0.0                                                       MAT01520
            SE(1)=0.                                                    MAT01530
            RETURN                                                      MAT01540
           ENDIF                                                        MAT01550
           JJJ1=4+2*NSEG                                                MAT01560
           IF (ABS(D).GT.SMAT(JJJ1).AND.ABS(DL).LT.SMAT(JJJ1)) THEN     MAT01570
            SMATJJ=SIGN(SMAT(JJJ1),DL)                                  MAT01580
            P1=PL+(SMATJJ-DL)*SE(1)                                     MAT01590
            D1=SMATJJ                                                   MAT01600
            CALL STRENG(SE(30),SE(33),SE(34),P1,PL,D1,DL,SE(17))        MAT01610
C ... AFTER REACHING FAILURE DUCTILITY, ENFORCE P=0.                    MAT01620
            P=0.                                                        MAT01630
            D=SIGN(SMAT(JJJ1),D)                                        MAT01640
            DDUCT=ABS(D)/SMAT(JJJ1)                                     MAT01650
            RETURN                                                      MAT01660
           ENDIF                                                        MAT01670
           IF (ABS(DL).GT.SMAT(JJJ1).AND.ABS(D).GT.SMAT(JJJ1)) THEN     MAT01680
C ... ENFORCED THE FORCE=0. AFTER FAILURE DUCTILITY                     MAT01690
            P=0.                                                        MAT01700
            RETURN                                                      MAT01710
           ENDIF                                                        MAT01720
          ENDIF                                                         MAT01730
C ... MODIFICATION END                                                  MAT01740
          NI  =SMAT(2)                                                  MAT01750
          SI  = SMAT(5)/SMAT(5+NSEG)                                    MAT01760
          J1=NSEG+5                                                     MAT01770
          J3=NI+33                                                      MAT01780
          J4=2*NI+33                                                    MAT01790
          DVEL=V*VL                                                     MAT01800
C     SET VELOCITY FLAG TO 1 TO REMOVE INFL OF HIGH FREQ VELOC          MAT01810
          DVEL = 1                                                      MAT01820
          ICYC=0                                                        MAT01830
  310 STIFF=SE(1)                                                       MAT01840
          P=PL+(D-DL)*STIFF                                             MAT01850
          A=SE(4)                                                       MAT01860
          IF (A.EQ.0 .AND. SE(2).EQ.0) THEN                             MAT01870
              A=SIGN(SMAT(5),P)                                         MAT01880
              IF (P.LT.PL .AND. PL.GT.0 .AND. P.GT.0) A=0               MAT01890
              IF (P GT.PL .AND. PL.LT.0 .AND. P LT.0) A=0               MAT01900
          ENDIF                                                         MAT01910
          ICYC=ICYC+1                                                   MAT01920
C                                                                       MAT01930
C----- IS P WITHIN LIMITS ?                                             MAT01940
          IF (ICYC.GT.10) THEN                                          MAT01950
              WRITE(108,*) 'MORE THAN 10 CYCLES IN MAT04,  IELNO:',IELNO MAT01960
          ELSE IF ((PL LE P .AND. P.LT.A) .OR. (PL GE P .AND. P.GT.A) .OR.MAT01970
     &          (DL.EQ.D)) THEN                                         MAT01980
              CALL STRENG (SE(28),SE(31),SE(32),P,PL,D,DL,SE(15) )      MAT01990
              CALL HYST04( P, D,PL,DL,DVEL,NSEG,NI,                     MAT02000
     &           SMAT( 5),SMAT(J1),SMAT( 5),SMAT(J1),SI,SMAT(3),SMAT(4),MAT02010
     &           SE( 1)  ,SE( 2)  ,SE( 3)  ,SE( 4)  ,SE( 5) ,SE( 6) ,  MAT02020
     &           SE( 7)  ,SE( 8)  ,SE( 9)  ,SE(10)  ,SE(11) ,SE(12) ,  MAT02030
     &           SE(13)  ,SE(14)  ,SE(15)  ,SE(16)  ,SE(19) ,SE(22) ,  MAT02040
     &           SE(28)  ,SE(31)  ,SE(32)  ,SE(33)  ,SE(J3) ,SE(J4) )  MAT02050
C----- IS P GREATER THAN LIMITS ?                                       MAT02060
          ELSE IF (((PL LE.A .AND. A LE.P) .OR  (PL GE.A .AND. A GE.P)) .AND.MAT02070
     &          (STIFF.NE.0.00)) THEN                                   MAT02080
              D1=DL+(A-PL)/STIFF                                        MAT02090
              P1=A                                                      MAT02100
              CALL STRENG (SE(28),SE(31),SE(32),P1,PL,D1,DL,SE(15) )    MAT02110
              DL=D1                                                     MAT02120
              PL=P1                                                     MAT02130
              IF (ABS(P-PL) GT. 0.0005*ABS(P+PL)) THEN                  MAT02140
                  PI=PL+(P-PL)* 001                                     MAT02150
                  DI=DL+(D-DL)*.001                                     MAT02160
              ELSE                                                      MAT02170
                  PI=P                                                  MAT02180
                  DI=D                                                  MAT02190
              ENDIF                                                     MAT02200
              CALL HYST04(PI,D1,PL,DL,1.00,NSEG,NI,                     MAT02210
     &           SMAT( 5),SMAT(J1),SMAT( 5),SMAT(J1),SI,SMAT(3),SMAT(4),MAT02220
     &           SE( 1)  ,SE( 2)  ,SE( 3)  ,SE( 4)  ,SE( 5) ,SE( 6) ,  MAT02230
     &           SE( 7)  ,SE( 8)  ,SE( 9)  ,SE(10)  ,SE(11) ,SE(12) ,  MAT02240
     &           SE(13)  ,SE(14)  ,SE(15)  ,SE(16)  ,SE(19) ,SE(22) ,  MAT02250
     &           SE(28)  ,SE(31)  ,SE(32)  ,SE(33)  ,SE(J3) ,SE(J4) )  MAT02260
              GO TO 310                                                 MAT02270
C----  IS P LESS THAN LIMITS ?                                          MAT02280
          ELSE IF ((P LT.PL .AND. PL.LE.A) .OR. (P GT.PL .AND  PL GE.A))THENMAT02290
              IF (ABS(P-PL).GT  0.0005*ABS(P+PL)) THEN                  MAT02300
                  PI=PL+(P-PL)* 001                                     MAT02310
                  DI=DL+(D-DL)*.001                                     MAT02320
              ELSE                                                      MAT02330
                  PI=P                                                  MAT02340
```

```
            DI-D                                                        MAT02350
          ENDIF                                                         MAT02360
          CALL HYST04(PI,DI,PL,DL,1.00,NSEG,NI,                         MAT02370
     &         SMAT( 5),SMAT(J1),SMAT( 5),SMAT(J1),SI,SMAT(3),SMAT(4),  MAT02380
     &         SE( 1)  ,SE( 2)  ,SE( 3)  ,SE( 4)  ,SE( 5) ,SE( 6) ,     MAT02390
     &         SE( 7)  ,SE( 8)  ,SE( 9)  ,SE(10)  ,SE(11) ,SE(12) ,     MAT02400
     &         SE(13)  ,SE(14)  ,SE(15)  ,SE(16)  ,SE(19) ,SE(22) ,     MAT02410
     &         SE(28)  ,SE(31)  ,SE(32)  ,SE(33)  ,SE(03) ,SE(04) )     MAT02420
          GO TO 310                                                     MAT02430
      ENDIF                                                             MAT02440
C                                                                       MAT02450
C---- - SAVE MAXIMUN DISPL AND LOAD                                     MAT02460
      IF (ABS(D).GT.ABS(SE(34+3*NI))) THEN                              MAT02470
          SE(33+  3*NI)=P                                               MAT02480
          SE(34+  3*NI)=D                                               MAT02490
      ENDIF                                                             MAT02500
C                                                                       MAT02510
C------ TRANSFER ENERGIES FOR BOTH IOPT=3 AND 4                         MAT02520
  400 EESE=SE(28)                                                       MAT02530
      EPSE=SE(31)                                                       MAT02540
C                                                                       MAT02550
C------ TRANSFER DUCTILITIES AND EXCRUSION RATIOS FOR BOTH IOPT=3 AND 4 MAT02560
      DO 410 I=1,3                                                      MAT02570
      DUCT(I)=MAX(SE(15+I),SE(18+I))                                    MAT02580
      EXCR(I)=SE(21+I)                                                  MAT02590
  410 EXCR(I+3)=SE(21+I+3)                                              MAT02600
      RETURN                                                            MAT02610
C                                                                       MAT02620
C------------------------- DAMAGE INDEX ------------------------------- MAT02630
  500 EESE =SE(28)                                                      MAT02640
      EPSE =SE(31)                                                      MAT02650
      NSEGS=SMAT(1)                                                     MAT02660
      NI   =SMAT(2)                                                     MAT02670
      DMAX =SMAT(4+2*NSEGS)                                             MAT02680
      DY   =SMAT(4)                                                     MAT02690
      BDAM =SMAT(5+2*NSEGS)                                             MAT02700
      P=SE(33+  3*NI)                                                   MAT02710
      D=SE(34+  3*NI)                                                   MAT02720
      IF (DY.LE.SMAT(5+NSEGS)) THEN                                     MAT02730
          PY=SMAT(5)*DY/SMAT(5+NSEGS)                                   MAT02740
      ELSE                                                              MAT02750
          DO 510 I=2,NSEGS                                              MAT02760
            P1=SMAT(3+I)                                                MAT02770
            P2=SMAT(4+I)                                                MAT02780
            D1=SMAT(3+I+NSEGS)                                          MAT02790
            D2=SMAT(4+I+NSEGS)                                          MAT02800
            IF (DY.GE.D1 .AND. DY.LE.D2)  THEN                          MAT02810
              PY=P1+(DY-D1)*(P2-P1)/(D2-D1)                             MAT02820
              GO TO 520                                                 MAT02830
            ENDIF                                                       MAT02840
  510       CONTINUE                                                    MAT02850
      ENDIF                                                             MAT02860
  520 DAMAGE=ABS(D)/DMAX + BDAM / (PY*DMAX) * EPSE                      MAT02870
C                                                                       MAT02880
      RETURN                                                            MAT02890
      END                                                               MAT02900
C                                                                       MAT00010
C*********************************************************************  MAT00020
C                                                                       MAT00030
      SUBROUTINE MAT06(IOPT,IELNO,FIRST,MAT,SE,P,PL,D,DL,V,VL,          MAT00040
     &                 LELEM,LMAT,SMAT,EESE,EPSE,DUCT,EXCR,DAMAGE)      MAT00050
      LOGICAL FIRST                                                     MAT00060
      DIMENSION SE(100),SMAT(13),DUCT(3),EXCR(6)                        MAT00070
      CHARACTER*80 TYPE                                                 MAT00080
C                                                                       MAT00090
      LELEM=49                                                          MAT00100
      IF (IOPT.EQ.0) RETURN                                             MAT00110
C                                                                       MAT00120
C---------- ------------------------------------------------------------MAT00130
C VARIABLES                                                             MAT00140
C---------- ------------------------------------------------------------MAT00150
C------- --- GLOBAL VARIABLES                                          MAT00160
C IOPT   = 1, INITIALIZE MATERIAL                                       MAT00170
C        = 2, GET STIFFNESS                                             MAT00180
C RINPUT = INPUT DATA                                                   MAT00190
C SMAT   = INTERNAL STORAGE                                             MAT00200
C SE     = OUTPUT STIFFNESS                                             MAT00210
C---------- ------------------------------------------------------------MAT00220
C           ---- TAKEDA DATA ----                                       MAT00230
C       SMAT( 1)= EI,           ELASTIC STIFFNESS                       MAT00240
C       SMAT( 2)= PC,           CRACKING LOAD                           MAT00250
C       SMAT( 3)= DC,           CRACKING DISPLACEMENT                   MAT00260
C       SMAT( 4)= PY,           YEILD LOAD                              MAT00270
C       SMAT( 5)= DY,           YEILD DISPLACEMENT                      MAT00280
C       SMAT( 6)= PU,           ULTIMATE LOAD                           MAT00290
C       SMAT( 7)= DU,           ULTIMATE DISPLACEMENT                   MAT00300
C       SMAT( 8)= OC,           INITIAL STIFFNESS                       MAT00310
C       SMAT( 9)= CY,           CRACKED STIFFNESS                       MAT00320
C       SMAT(10)= YU,           ULTIMATE STIFFNESS                      MAT00330
C       SMAT(11)= C'Y,          RELOADING STIFFNESS                     MAT00340
C       SMAT(12)= CSE           CONSTANT STRAIN ENERGY AT YEILD         MAT00350
C       SMAT(13)= BDAM          BETA FOR DAMAGE INDEX                   MAT00360
C                                                                       MAT00370
C---------- INPUT DATA FOR TAKEDA HYSTERESIS MODEL ----------           CMAT00380
C     THE FOLLOWING DATA IS INPUT ONCE FOR EACH MODEL                   CMAT00390
C                                                                       CMAT00400
C  EI     = MOMENT OF INERTIA                                           CMAT00410
C  G      = MOMENT GRADIENT                                             CMAT00420
C  PC,DC  = CRACKING LOAD AND DISPLACEMENT                              CMAT00430
C  PY,DY  = YEILD LOAD AND DISPLACEMENT                                 CMAT00440
C  PU,DU  = ULTIMATE LOAD AND DISPLACEMENT                              CMAT00450
C                                                                       CMAT00460
C---------- ------- ---------- ---------- ---------- ---------- -------CMAT00470
      GO TO (100,200,300,400,500) IOPT                                  MAT00480
C                                                                       MAT00490
  100 CONTINUE                                                          MAT00500
      READ(105,*) TYPE ,(SMAT(I),I=1,7),BDAM                            MAT00510
C                                                                       MAT00520
      LMAT = 13                                                         MAT00530
      SMAT( 8)= SMAT(2)          / SMAT(3)                              MAT00540
      SMAT( 9)=(SMAT(4)-SMAT(2))/(SMAT(5)-SMAT(3))                      MAT00550
      SMAT(10)=(SMAT(6)-SMAT(4))/(SMAT(7)-SMAT(5))                      MAT00560
      SMAT(11)=(SMAT(2)*SMAT(4))/(SMAT(3)*SMAT(5))                      MAT00570
      SMAT( 1)=SMAT(1)                                                  MAT00580
      SMAT(12)=SMAT(2)*SMAT(3)/2.*.5*(SMAT(2)*SMAT(4))*(SMAT(5)-SMAT(3)) MAT00590
      SMAT(13)=BDAM                                                     MAT00600
C                                                                       MAT00610
      IF (FIRST) WRITE(108,5)                                           MAT00620
      WRITE(108,6) MAT,BDAM,SMAT(1),SMAT(2),SMAT(4),SMAT(6),           MAT00630
     &             SMAT(3),SMAT(5),SMAT(7)                              MAT00640
    5 FORMAT(///'     TAKEDA HYSTERESIS MODEL- FOR UNIT LENGTH MEMBER'/ MAT00650
     &       '  ----------------------------------------------------'//)MAT00660
     &       '  MAT      BETA          EI          ',7X,                MAT00670
     &       'CRACK',10X,'YEILD',7X,'ULTIMATE')                         MAT00680
    6 FORMAT (/1X,I5,2G15.6,1X, 2X,3G15.6,/39X,3G15.6/)                 MAT00690
      RETURN                                                            MAT00700
C------------------------- GET STIFFNESS TERMS ------------------------ MAT00710
  200 CONTINUE                                                          MAT00720
C           --- TAKEDA MODEL DATA STORAGE FOR MEMBER ----               MAT00730
C       SE( 1)= S                                                       MAT00740
C       SE( 2)= HRN                                                     MAT00750
C       SE( 3)= A                                                       MAT00760
C       SE( 4)= B                                                       MAT00770
C       SE( 5)= IDRO                                                    MAT00780
C       SE( 6)= RF                                                      MAT00790
C       SE( 7)= RNRL                                                    MAT00800
C       SE( 8)= UM1                                                     MAT00810
C       SE( 9)= UM2                                                     MAT00820
C       SE(10)= UM3                                                     MAT00830
C       SE(11)= UM4                                                     MAT00840
C       SE(12)= S1                                                      MAT00850
C       SE(13)= X0                                                      MAT00860
C       SE(14)= X0UM                                                    MAT00870
C       SE(15)= X0Y                                                     MAT00880
C       SE(16)= U0D                                                     MAT00890
C       SE(17)= U0                                                      MAT00900
C       SE(18)= X1UM                                                    MAT00910
C       SE(19)= U1                                                      MAT00920
C       SE(20)= U1D                                                     MAT00930
C       SE(21)= X2U0                                                    MAT00940
C       SE(22)= U2                                                      MAT00950
C       SE(23)= X3U1                                                    MAT00960
C       SE(24)= U3                                                      MAT00970
C       SE(25)= QY                                                      MAT00980
C       SE(26)= MAXPLG                                                  MAT00990
C       SE(27)= ACHNG                                                   MAT01000
C       SE(28)= S_EQ_UNLOAD                                             MAT01010
C       SE(29)= EESE                                                    MAT01020
C       SE(30)= EESE                                                    MAT01030
C       SE(31)= EESE                                                    MAT01040
C       SE(32)= EPSE                                                    MAT01050
C       SE(33)= PSE_OLD                                                 MAT01060
C       SE(34)= UPOS(1)                                                 MAT01070
C       SE(35)= UPOS(2)                                                 MAT01080
C       SE(36)= UPOS(3)                                                 MAT01090
C       SE(37)= UNEG(1)                                                 MAT01100
C       SE(38)= UNEG(2)                                                 MAT01110
C       SE(39)= UNEG(3)                                                 MAT01120
C       SE(40)= EXCR(1)                                                 MAT01130
C       SE(41)= EXCR(2)                                                 MAT01140
C       SE(42)= EXCR(3)                                                 MAT01150
C       SE(43)= EXCR(4)                                                 MAT01160
C       SE(44)= EXCR(5)                                                 MAT01170
C       SE(45)= EXCR(6)                                                 MAT01180
C       SE(46)= P_MAX                                                   MAT01190
C       SE(47)= D_MAX                                                   MAT01200
C       SE(48)= SMAT(8)                                                 MAT01210
C       SE(49)= SMAT(1)                                                 MAT01220
C                                                                       MAT01230
      EI=SMAT(1)                                                        MAT01240
      GRAD=SMAT(12)                                                     MAT01250
      DO 210 I=1,47                                                     MAT01260
  210   SE(I)=0                                                         MAT01270
      CALL HYST06(P,D,PL,DL,V,VL,SE(1),SE(2),SE(3),SE(4),SE(5),         MAT01280
     &  SE(6),SMAT(2),SMAT(3),SMAT(4),SMAT(5),SMAT(6),SMAT(8),          MAT01290
     &  SMAT(9),SMAT(10),SMAT(11),SMAT(12),SE(7),SE(8),SE(9),SE(10),    MAT01300
     &  SE(11),SE(12),SE(13),SE(14),SE(15),SE(16),SE(17),SE(18),SE(19), MAT01310
     &  SE(20),SE(21),SE(22),SE(23),SE(24),SE(25),SE(27),               MAT01320
     &  SE(28),SE(29),SE(32),SE(33),SE(34),SE(37),SE(40))               MAT01330
C                                                                       MAT01340
      SE(48)=SMAT(8)                                                    MAT01350
      SE(49)=EI                                                         MAT01360
C                                                                       MAT01370
      RETURN                                                            MAT01380
C                                                                       MAT01390
C------------------------ GET STIFFNESS TERMS ------------------------- MAT01400
  300 CONTINUE                                                          MAT01410
C       SZ( 1)= STIFF                                                   MAT01420
C       SE(27)= A                                                       MAT01430
C       SE(28)= S_EQ_UNLOAD                                             MAT01440
      ICYC=0                                                            MAT01450
  310 STIFF=SE(1)                                                       MAT01460
      P=PL+(D-DL)*STIFF                                                 MAT01470
      A=SE(27)                                                          MAT01480
      IF (A.EQ.0 .AND  SE(7).EQ.0) A=SMAT(2)                            MAT01490
      ICYC=ICYC+1                                                       MAT01500
C                                                                       MAT01510
C----- IS P WITHIN LIMITS ?                                             MAT01520
      IF (ICYC.GT.5) THEN                                               MAT01530
          WRITE(108,*) 'MORE THAN 5 CYCLES IN MAT06,  IELNO ',IELNO     MAT01540
      ELSE IF ((PL.LE.P .AND. P.LE.A) .OR. (PL.GE.P .AND. P.GE.A) .OR.  MAT01550
     &         (DL.EQ.D)) THEN                                          MAT01560
          CALL STRENG(SE(29),SE(32),SE(33),P,PL,D,DL,SE(28))            MAT01570
          CALL HYST06(P,D,PL,DL,V,VL,SE(1),SE(2),SE(3),SE(4),           MAT01580
     &  SE(5),SE(6),SMAT(2),SMAT(3),SMAT(4),SMAT(5),SMAT(6),SMAT(8),    MAT01590
     &  SMAT(9),SMAT(10),SMAT(11),SMAT(12),SE(7),SE(8),SE(9),SE(10),    MAT01600
     &  SE(1),SE(12),SE(13),SE(14),SE(15),SE(16),SE(17),SE(18),SE(19),  MAT01610
     &  SE(20),SE(21),SE(22),SE(23),SE(24),SE(25),SE(27),               MAT01620
     &  SE(28),SE(29),SE(32),SE(33),SE(34),SE(37),SE(40))               MAT01630
C                                                                       MAT01640
C----- IS P GREATER THAN LIMITS ?                                       MAT01650
      ELSE IF (((PL.LE.A .AND. A.LT.P) .OR. (PL.GE.A .AND  A.GE.P)).AND.MAT01660
     &         (STIFF.NE.0.00)) THEN                                    MAT01670
          DL=DL+(A-PL)/STIFF                                            MAT01680
          PL=A                                                          MAT01690
          CALL STRENG(SE(29),SE(32),SE(33),PL,PL,DL,DL,SE(28))          MAT01700
          CALL HYST06(PL,DL,PL,DL,1.,1.,SE(1),SE(2),SE(3),SE(4),        MAT01710
     &  SE(5),SE(6),SMAT(2),SMAT(3),SMAT(4),SMAT(5),SMAT(6),SMAT(8),    MAT01720
     &  SMAT(9),SMAT(10),SMAT(11),SMAT(12),SE(7),SE(8),SE(9),SE(10),    MAT01730
     &  SE(11),SE(12),SE(13),SE(14),SE(15),SE(16),SE(17),SE(18),SE(19), MAT01740
     &  SE(20),SE(21),SE(22),SE(23),SE(24),SE(25),SE(27),               MAT01750
     &  SE(28),SE(29),SE(32),SE(33),SE(34),SE(37),SE(40))               MAT01760
          DL=DL                                                         MAT01770
          PL=PL                                                         MAT01780
          GO TO 310                                                     MAT01790
C----- IS P LESS THAN LIMITS ?                                          MAT01800
      ELSE IF ((P.LT.PL.AND.PL.LE.A).OR.(P.GT.PL .AND. PL.GE.A)) THEN   MAT01810
          IF (ABS(P-PL).GT  0.0005*ABS(P+PL)) THEN                      MAT01820
              PI=PL+(P-PL)* .001                                        MAT01830
              DI=DL+(D-DL)* .001                                        MAT01840
          ELSE                                                          MAT01850
              PI=P                                                      MAT01860
              DI=D                                                      MAT01870
          ENDIF                                                         MAT01880
          CALL HYST06(PI,DI,PL,DL,1.,1.,SE(1),SE(2),SE(3),SE(4),       MAT01890
     &  SE(5),SE(6),SMAT(2),SMAT(3),SMAT(4),SMAT(5),SMAT(6),SMAT(8),    MAT01900
     &  SMAT(9),SMAT(10),SMAT(11),SMAT(12),SE(7),SE(8),SE(9),SE(10),    MAT01910
     &  SE(11),SE(12),SE(13),SE(14),SE(15),SE(16),SE(17),SE(18),SE(19), MAT01920
     &  SE(20),SE(21),SE(22),SE(23),SE(24),SE(25),SE(27),               MAT01930
     &  SE(28),SE(29),SE(32),SE(33),SE(34),SE(37),SE(40))               MAT01940
          GO TO 310                                                     MAT01950
      ENDIF                                                             MAT01960
C                                                                       MAT01970
C------ MAXIMUM DISPLACEMENT                                            MAT01980
      IF (ABS(D).GT.ABS(SE(47))) THEN                                   MAT01990
          SE(46)= P                                                     MAT02000
          SE(47)= D                                                     MAT02010
      ENDIF                                                             MAT02020
C                                                                       MAT02030
C------ TRANSFER ENERGIES FOR BOTH IOPT=3 AND 4                         MAT02040
  400 EESE=SE(29)                                                       MAT02050
      EPSE=SE(32)                                                       MAT02060
C                                                                       MAT02070
C------ TRANSFER DUCTILITIES AND EXCRUSION RATIOS FOR BOTH IOPT=3 AND 4.MAT02080
      DO 410 I=1,3                                                      MAT02090
      DUCT(I)=MAX(SE(33+I),SE(36+I))                                    MAT02100
      EXCR(I)=SE(39+I)                                                  MAT02110
  410 EXCR(I+3)=SE(42+I)                                                MAT02120
      RETURN                                                            MAT02130
C                                                                       MAT02140
C------------------ DAMAGE INDEX ------------------------------------- MAT02150
  500 EESE=SE(29)                                                       MAT02160
      EPSE=SE(32)                                                       MAT02170
      PY=  SMAT( 4)                                                     MAT02180
      PU=  SMAT( 6)                                                     MAT02190
      DU=  SMAT( 7)                                                     MAT02200
      P =  SE(46)                                                       MAT02210
      D =  SE(47)                                                       MAT02220
      BDAM=SMAT(13)                                                     MAT02230
      DAMAGE=ABS(D)/DU + BDAM/(PY*DU) * EPSE                            MAT02240
C                                                                       MAT02250
      RETURN                                                            MAT02260
      END                                                               MAT02270
C                                                                       MAT00010
C*********************************************************************  MAT00020
      SUBROUTINE MAT07(IOPT,FIRST,MAT,SE,P,PL,D,DL,V,VL,LELEM,          MAT00030
     &                 LMAT,SMAT,EESE,EPSE,DUCT,EXCR,DAMAGE)            MAT00040
C                                                                       MAT00050
      IMPLICIT REAL(A-H,O-Z)                                            MAT00060
      LOGICAL FIRST                                                     MAT00070
      DIMENSION SE(100),SMAT(7),DUCT(3),EXCR(6)                         MAT00080
      CHARACTER*80 TYPE                                                 MAT00090
```

```
C                                                                       MAT00100
C------  RESERVE STORAGE FOR ELEMENT                                    MAT00110
      LELEM=24                                                          MAT00120
      IF (IOPT EQ.0) RETURN                                             MAT00130
C                                                                       MAT00140
C-----------------------------------------------------------------------MAT00150
C VARIABLES:                                                            MAT00160
C-----------------------------------------------------------------------MAT00170
C---------- GLOBAL VARIABLES                                           MAT00180
C IOPT   = 1, INITIALIZE MATERIAL                                       MAT00190
C        = 2, GET STIFFNESS                                             MAT00200
C RINPUT = INPUT DATA                                                   MAT00210
C SMAT   = INTERNAL STORAGE                                             MAT00220
C SE     = OUTPUT STIFFNESS                                             MAT00230
C-----------------------------------------------------------------------MAT00240
      GO TO (100,200,300,400,500) IOPT                                  MAT00250
C                                                                       MAT00260
  100 CONTINUE                                                          MAT00270
C                                                                       MAT00280
C----- INTERPERTATE INPUT DATA                                          MAT00290
C      KE     = SMAT( 1)                                               MAT00300
C      PY     = SMAT( 2)                                               MAT00310
C      KIE    = SMAT( 3)                                               MAT00320
C      DY     = SMAT( 4)                                               MAT00330
C      CSE    = SMAT( 5)                                               MAT00340
C      DMAX   = SMAT( 6)                                               MAT00350
C      BDAM   = SMAT( 7)                                               MAT00360
C                                                                       MAT00370
      READ(105,*) TYPE ,(SMAT(I),I=1,3),DUCMAX,BDAM                     MAT00380
C                                                                       MAT00390
      LMAT = 7                                                          MAT00400
      SMAT(4)=SMAT(2)/SMAT(1)                                           MAT00410
      SMAT(5)=SMAT(4)*SMAT(2)/2                                         MAT00420
      SMAT(6)=SMAT(4)*DUCMAX                                            MAT00430
      SMAT(7)=BDAM                                                      MAT00440
C                                                                       MAT00450
      IF (FIRST) WRITE(108,191)                                         MAT00460
      WRITE(108,192) MAT,(SMAT(K),K=1,3),DUCMAX,BDAM                    MAT00470
C                                                                       MAT00480
  191  FORMAT (///'     BILINEAR HYSTERESIS MODEL PARAMETERS'/          MAT00490
     & '     ----------------------------------'//,'  MAT.',6X,        MAT00500
     & '   KE',10X,'   PY',10X,'  KIE', 7X,'MAX DUCT',10X,' BETA')     MAT00510
  192  FORMAT (1X,I5,7G15.6/)                                           MAT00520
C                                                                       MAT00530
      RETURN                                                            MAT00540
C                                                                       MAT00550
C---------------------- INITILIZE STIFFNESS ---------------------------MAT00560
  200 CONTINUE                                                          MAT00570
C          ---- ELASTO-PLASTIC MODEL DATA STORAGE FOR MEMBER ----      MAT00580
C      SE( 1)= K                                                        MAT00590
C      SE( 2)= RULE                                                     MAT00600
C      SE( 3)= EPSE_OLD                                                 MAT00610
C      SE( 4)= A, UPPER LIMIT BEFORE RULE CHANGE                        MAT00620
C      SE( 5)= B, LOWER LIMIT BEFORE RULE CHANGE                        MAT00630
C      SE( 6)=EESE(1)                                                   MAT00640
C      SE( 7)=EESE(2)                                                   MAT00650
C      SE( 8)=EESE(3)                                                   MAT00660
C      SE( 9)=EPSE                                                      MAT00670
C      SE(10)=KEY                                                       MAT00680
C      SE(11)=UPOS(1)                                                   MAT00690
C      SE(12)=UPOS(2)                                                   MAT00700
C      SE(13)=UPOS(3)                                                   MAT00710
C      SE(14)=UNEG(1)                                                   MAT00720
C      SE(15)=UNEG(2)                                                   MAT00730
C      SE(16)=UNEG(3)                                                   MAT00740
C      SE(17)=EXCR(1)                                                   MAT00750
C      SE(18)=EXCR(2)                                                   MAT00760
C      SE(19)=EXCR(3)                                                   MAT00770
C      SE(20)=EXCR(4)                                                   MAT00780
C      SE(21)=EXCR(5)                                                   MAT00790
C      SE(22)=EXCR(6)                                                   MAT00800
C      SE(23)=F_MAX                                                     MAT00810
C      SE(24)=D_MAX                                                     MAT00820
C                                                                       MAT00830
C     INITILIZE RULE #1   ELASTIC BEHAVIOR                              MAT00840
C                                                                       MAT00850
      DO 210 I=1,22                                                     MAT00860
  210   SE(I)=0                                                         MAT00870
C                                                                       MAT00880
      CALL HYST07(0.,0.,0.,0.,SE(1),SE(2),SE(3),SE(4),SE(5),V,VL,       MAT00890
     &            SMAT(1),SMAT(2),SMAT(3),SMAT(4),SMAT(5),            MAT00900
     &            SE(6),SE(9),SE(10),SE(11),SE(14),SE(17))            MAT00910
C                                                                       MAT00920
      RETURN                                                            MAT00930
C                                                                       MAT00940
C---------- GET STIFFNESS TERMS AND ACTUAL FORCE ----------------------MAT00950
  300 CONTINUE                                                          MAT00960
C                                                                       MAT00970
C------  CHECK TO SEE IF DD = 0 ?                                       MAT00980
      P=P                                                               MAT00990
      D=D                                                               MAT01000
      IF (D.EQ.DL)  RETURN                                              MAT01010
C                                                                       MAT01020
C-------- CALL HYST07 TO CALC NEW STIFFNESS AND GET LOAD P              MAT01030
      CALL HYST07(P,D,PL,DL,SE(1),SE(2),SE(3),SE(4),SE(5),V,VL,         MAT01040
     &            SMAT(1),SMAT(2),SMAT(3),SMAT(4),SMAT(5),            MAT01050
     &            SE(6),SE(9),SE(10),SE(11),SE(14),SE(17))            MAT01060
C                                                                       MAT01070
      SE1=SE(1)                                                         MAT01080
C                                                                       MAT01090
C------ TRANSFER ENERGIES FOR BOTH IOPT=3 AND 4.                        MAT01100
      IF (ABS(D).GT.ABS(SE(24)))THEN                                    MAT01110
          SE(23)=P                                                      MAT01120
          SE(24)=D                                                      MAT01130
      ENDIF                                                             MAT01140
C                                                                       MAT01150
C ----- TRANSFER ENERGIES FOR BOTH IOPT=3 AND 4.                        MAT01160
  400 EESE=SE(6)                                                        MAT01170
      EPSE=SE(9)                                                        MAT01180
C                                                                       MAT01190
C------ TRANSFER DUCTILITIES AND EXCRUSION RATIOS FOR BOTH IOPT=3 AND 4 MAT01200
      DO 410 I=1,3                                                      MAT01210
      DUCT(I)=MAX(SE(10+I),SE(13+I))                                    MAT01220
      EXCR(I+3)=SE(16+I+3)                                              MAT01230
  410 EXCR(I)=SE(16+I)                                                  MAT01240
      RETURN                                                            MAT01250
C                                                                       MAT01260
C---------------------------- DAMAGE INDEX ----------------------------MAT01270
  500 EESE=SE(6)                                                        MAT01280
      EPSE=SE(9)                                                        MAT01290
      PY   = SMAT( 2)                                                   MAT01300
      DY   = SMAT( 4)                                                   MAT01310
      DMAX = SMAT( 6)                                                   MAT01320
      BDAM = SMAT( 7)                                                   MAT01330
      P=   SE(23)                                                       MAT01340
      D=   SE(24)                                                       MAT01350
      DAMAGE=ABS(D)/DMAX + BDAM / (PY*DMAX) * EPSE                      MAT01360
C                                                                       MAT01370
      RETURN                                                            MAT01380
      END                                                               MAT01390
C                                                                       MAT00010
C-----------------------------------------------------------------------MAT00020
C                                                                       MAT00030
      SUBROUTINE MAT08(IOPT,FIRST,MAT,SE,P,PL,D,DL,LELEM,               MAT00040
     &                 LMAT,SMAT,EESE,EPSE,DUCT,EXCR,DAMAGE)           MAT00050
C                                                                       MAT00060
      LOGICAL FIRST                                                     MAT00070
      DIMENSION SE(100),SMAT(9),DUCT(3),EXCR(6)                         MAT00080
      CHARACTER*80 TYPE                                                 MAT00090
C                                                                       MAT00100
C------  RESERVE STORAGE FOR ELEMENT                                    MAT00110
      LELEM=39                                                          MAT00120
      IF (IOPT.EQ.0) RETURN                                             MAT00130
C                                                                       MAT00140
C ------------------------- ---------- ---- ---- --- ----- -- ------ ---MAT00150
C VARIABLES                                                             MAT00160
C-----------------------------------------------------------------------MAT00170
C---------- GLOBAL VARIABLES                                           MAT00180
C IOPT   = 1, INITIALIZE MATERIAL                                       MAT00190
C        = 2, GET STIFFNESS                                             MAT00200
C RINPUT = INPUT DATA                                                   MAT00210
C SMAT   = INTERNAL STORAGE                                             MAT00220
C SE     = OUTPUT STIFFNESS                                             MAT00230
C-----------------------------------------------------------------------MAT00240
      GO TO (100,200,300,400,500) IOPT                                  MAT00250
C                                                                       MAT00260
  100 CONTINUE                                                          MAT00270
C                                                                       MAT00280
C----- INTERPERTATE INPUT DATA                                          MAT00290
C      E      = SMAT( 1)                                                MAT00300
C      A      = SMAT( 2)                                                MAT00310
C      R      = SMAT( 3)                                                MAT00320
C      YS     = SMAT( 4)                                                MAT00330
C      PY     = SMAT( 5)                                                MAT00340
C      CC     = SMAT( 6)                                                MAT00350
C      DMAX   = SMAT( 7)    : ULTIMATE DISPLACEMENT = DUCMAX*DELY        MAT00360
C      BDAM   = SMAT( 8)    : BATA PARAMETER                             MAT00370
C      SHAPE  = SMAT( 9)    : CROSS SECTION SHAPE                        MAT00380
C                             1: FOR BOX OR ANGLE SECTION                MAT00390
C                             2: FOR I SHAPE SECTION                     MAT00400
C                             3: FOR BOX SECTION CONSIDERING             MAT00410
C                                LOCAL BUCKLING. THIS IS ONLY FOR MEMBER MAT00420
C                                WITH KL/R LESS THAN 40 CASE             MAT00430
C-----------------------------------------------------------------------MAT00440
C                                                                       MAT00450
      READ (105,*) TYPE ,(SMAT(I),I=1,4), SHAPE,DUCMAX, BDAM            MAT00460
C                                                                       MAT00470
      LMAT = 9                                                          MAT00480
      PI=3.1415926                                                      MAT00490
      SMAT(5)=SMAT(2)*SMAT(4)                                           MAT00500
      SMAT(6)=SQRT(2.*PI*PI*SMAT(1)/SMAT(4) )                           MAT00510
      SMAT(7)=DUCMAX                                                    MAT00520
      SMAT(8)=BDAM                                                      MAT00530
      SMAT(9)=SHAPE                                                     MAT00540
C                                                                       MAT00550
      IF (FIRST) WRITE(108,191)                                         MAT00560
      WRITE(108,192) MAT,(SMAT(X),K=1,4),SHAPE,DUCMAX,BDAM              MAT00570
C                                                                       MAT00580
  191  FORMAT (///'     GOEL HYSTERESIS MODEL PARAMETERS'/               MAT00590
     &'     ------------------------------'//                          MAT00600
     & ' MAT.',3X,'    E',6X,'    A',6X,'    R',6X,'   YS',            MAT00610
     &  6X,' SHAPE',6X, 'MAX DUC', 6X,'  BETA')                         MAT00620
  192  FORMAT (1X,I5,7G12.6/)                                           MAT00630
C                                                                       MAT00640
      RETURN                                                            MAT00650
C                                                                       MAT00660
C---------------------- INITIALIZE STIFFNESS TERMS ---------------------MAT00670
  200 CONTINUE                                                          MAT00680
C          ---- GOEL HYSTERESIS MODEL DATA STORAGE FOR MEMBER ----     MAT00690
C      SE( 1)=         STIFFMEMBER AXIAL STIFFNESS                      MAT00700
C      SE( 2)= K       EFFECT SLENDER COEFFICIENT (TRANSFERD FROM ELE08) MAT00710
C      SE( 3)= NOCY    NO. OF CYCLE                                     MAT00720
C      SE( 4)= NRULE   RULE NUMBER                                      MAT00730
C      SE( 5)= EL      MEMBER LENGTH (TRANSFERED FROM ELE08)            MAT00740
C      SE( 6)= IRV     0:REVRSC= FALSE.   1:REVRSC=.TRUE.               MAT00750
C      SE( 7)= IDUC    0: HC=-12          1. HC .LT. -12                MAT00760
C      SE( 8)= PBOT                                                     MAT00770
C      SE( 9)= DBOT                                                     MAT00780
C      SE(10)= PTOP                                                     MAT00790
C      SE(11)= DTOP                                                     MAT00800
C      SE(12)= KEQ     UNLOADING EQUIVALENT STIFFNESS                   MAT00810
C      SE(13)= DUCT(1)                                                  MAT00820
C      SE(14)= DUCT(2)                                                  MAT00830
C      SE(15)= DUCT(3)                                                  MAT00840
C      SE(16)= EXCR(1)                                                  MAT00850
C      SE(17)= EXCR(2)                                                  MAT00860
C      SE(18)= EXCR(3)                                                  MAT00870
C      SE(19)= EXCR(4)                                                  MAT00880
C      SE(20)= EXCR(5)                                                  MAT00890
C      SE(21)= EXCR(6)                                                  MAT00900
C      SE(22)= DELY    YIELDING DISPLACEMENT                            MAT00910
C      SE(23)= EESE(1)                                                  MAT00920
C      SE(24)= EESE(2)                                                  MAT00930
C      SE(25)= EESE(2)                                                  MAT00940
C      SE(26)= EPSE                                                     MAT00950
C      SE(27)= PSEOLD                                                   MAT00960
C      SE(28)= CSE                                                      MAT00970
C      SE(29)= DMAX   ULTIMATE DISPLACEMENT                             MAT00980
C      SE(30)= D_MAX  MAXIMUM ELEMENT DISPLACEMENT FOR DAMAGE INDEX     MAT00990
C      SE(31)= P_MAX  ELEMENT LOAD CORRESPONDING TO D_MAX               MAT01000
C      SE(32)= NA     CURRENT CONTROL POINT NA                          MAT01010
C      SE(33)= VA     CURRENT CONTROL POINT VA                          MAT01020
C      SE(34)= HC     CURRENT CONTROL POINT HC                          MAT01030
C      SE(35)= VC     CURRENT CONTROL POINT VC                          MAT01040
C      SE(36)= RSN    CURRENT RESIDUAL STRAIN VALUE                     MAT01050
C      SE(37)= DMAX   CURRENT MAXIMUM AXIAL DISPLACEMENT FOR DUCTILITIES MAT01060
C      SE(38)= REGIN  CURRENT REGINE ZONE NUMBER                        MAT01070
C      SE(39)= PREGIN PREVIOUT REGIN ZONE NUMBER                        MAT01080
C-----------------------------------------------------------------------MAT01090
      SE(1)=1                                                           MAT01100
      SE(3)=1                                                           MAT01110
      SE(4)=1                                                           MAT01120
      DO 210 I=6,28                                                     MAT01130
  210   SE(I)=0                                                         MAT01140
      SE(32)=0                                                          MAT01150
      SE(33)=1                                                          MAT01160
      SE(39)=1                                                          MAT01170
C                                                                       MAT01180
      CALL HYST08(0.,0.,0.,0.,SE(5),SE(2),SMAT(1),SMAT(2),SMAT(3),      MAT01190
     &    SMAT(4),SMAT(5),SMAT(9),SMAT(6),SE(3),SE(4),SE(1),SE(6),      MAT01200
     &    SE(7),SE(32),SE(33),SE(34),SE(35),SE(36),SE(37),SE(38),       MAT01210
     &    SE(39),SE(8),SE(9),SE(10),SE(11),SE(12),SE(13),               MAT01220
     &    SE(16),SE(22),SE(23),SE(26),SE(27),SE(28) )                   MAT01230
      SE(28)=SE(22)*SMAT(5)/2                                           MAT01240
      SE(29)=SE(22)*SMAT(7)                                             MAT01250
      SE(30)=0                                                          MAT01260
      SE(31)=0                                                          MAT01270
C                                                                       MAT01280
      RETURN                                                            MAT01290
C---------------------- GET STIFFNESS TERMS ----------------------------MAT01300
  300 CONTINUE                                                          MAT01310
C                                                                       MAT01320
C-------- CALL HYST07 TO CALC NEW STIFFNESS STIF                        MAT01330
      CALL HYST08(P,PL,D,DL,SE(5),SE(2),SMAT(1),SMAT(2),SMAT(3),        MAT01340
     &    SMAT(4),SMAT(5),SMAT(9),SMAT(6),SE(3),SE(4),SE(1),SE(6),      MAT01350
     &    SE(7),SE(32),SE(33),SE(34),SE(35),SE(36),SE(37),SE(38),       MAT01360
     &    SE(39),SE(8),SE(9),SE(10),SE(11),SE(12),SE(13),               MAT01370
     &    SE(16),SE(22),SE(23),SE(26),SE(27),SE(28) )                   MAT01380
C                                                                       MAT01390
C -- STORE MAXIMUM DISPLACEMENT AND CORRESPONDING LOAD                  MAT01400
      IF(ABS(D) .GT. ABS(SE(30))) THEN                                  MAT01410
            SE(30)=D                                                    MAT01420
            SE(31)=P                                                    MAT01430
      ENDIF                                                             MAT01440
C                                                                       MAT01450
C--- GET ENERGIES, DUCTILITY AND EXCURSION FOR BOTH IOPT=3 AND 4 ----   MAT01460
  400 CONTINUE                                                          MAT01470
C                                                                       MAT01480
C --- TRANSFER ENERGIES FOR BOTH IOPT=3 AND 4.                         MAT01490
      EESE=SE(23)                                                       MAT01500
      EPSE=SE(26)                                                       MAT01510
C                                                                       MAT01520
C --- TRANSFER DUCTILITIES AND EXCURSION RATIOS FOR BOTH IOPT=3 AND 4   MAT01530
      DO 410 I=1,3                                                      MAT01540
        DUCT(I)=SE(12+I)                                                MAT01550
        EXCR(I+3)=SE(15+I+3)                                            MAT01560
  410   EXCR(I)=SE(15+I)                                                MAT01570
C                                                                       MAT01580
        RETURN                                                          MAT01590
C---------------------- DAMAGE INDEX -----------------------------------MAT01600
  500 CONTINUE                                                          MAT01610
      EESE=SE(23)                                                       MAT01620
```

```
      EPSE=SE(26)                                                       MAT01630
      DMAX=SE(29)                                                       MAT01640
      D=SE(30)                                                          MAT01650
      P=SE(31)                                                          MAT01660
      PY=SMAT(5)                                                        MAT01670
      BDAM=SMAT(8)                                                      MAT01680
      DAMAGE=ABS(D)/DMAX + BDAM / (PY*DMAX) * EPSE                      MAT01690
C                                                                       MAT01700
      RETURN                                                            MAT01710
      END                                                               MAT01720
C                                                                       MAT00010
C-----------------------------------------------------------------------MAT00020
C                                                                       MAT00030
      SUBROUTINE MAT09(IOPT,FIRST,MAT,SE,P,PL,D,DL,LELEM,LMAT,          MAT00040
     &                 SMAT,EESE,EPSE,DUCT,EXCR,DAMAGE)                 MAT00050
C                                                                       MAT00060
      LOGICAL FIRST                                                     MAT00070
      DIMENSION SE(100),SMAT(4),DUCT(3),EXCR(6)                         MAT00080
      CHARACTER*80 TYPE                                                 MAT00090
C                                                                       MAT00100
C------- RESERVE STORAGE FOR ELEMENT                                    MAT00110
      LELEM=66                                                          MAT00120
      IF (IOPT.EQ.0) RETURN                                             MAT00130
C                                                                       MAT00140
C ......................................................................MAT00150
C VARIABLES:                                                            MAT00160
C-----------------------------------------------------------------------MAT00170
C---------- GLOBAL VARIABLES                                           MAT00180
C IOPT   = 1, INITIALIZE MATERIAL                                       MAT00190
C        = 2, INITIALIZE STIFFNESS                                      MAT00200
C        = 3, GET STIFFNESS TERM                                        MAT00210
C        = 4, GET ENERGIES, DUCTILITY AND EXCURSION                     MAT00220
C        = 5, CALCULATE DAMAGE INDEX                                    MAT00230
C RINPUT - INPUT DATA                                                   MAT00240
C SMAT   - INTERNAL STORAGE                                             MAT00250
C SE     - OUTPUT STIFFNESS                                             MAT00260
C-----------------------------------------------------------------------MAT00270
      GO TO (100,200,300,400,500),IOPT                                  MAT00280
C                                                                       MAT00290
  100 CONTINUE                                                          MAT00300
C                                                                       MAT00310
C--- INTERPERTATE INPUT DATA                                            MAT00320
C     HA    - SMAT( 1)                                                  MAT00330
C     VA    - SMAT( 2)                                                  MAT00340
C     DUCMAX- SMAT( 3)    : FAILURE DUCTILITY                           MAT00350
C     BDAM  - SMAT( 4)    . BATA PARAMETER                              MAT00360
C                                                                       MAT00370
      READ(105,*) TYPE ,(SMAT(I),I=1,2), DUCMAX, BDAM                   MAT00380
      LMAT = 4                                                          MAT00390
      SMAT(3)=DUCMAX                                                    MAT00400
      SMAT(4)=BDAM                                                      MAT00410
      IF (FIRST) WRITE(108,191)                                         MAT00420
      WRITE(108,192) MAT,(SMAT(K),K=1,2),DUCMAX,BDAM                    MAT00430
  191 FORMAT (///' PINO-SUAREZ TOWER LONG DIR. HYST. MODEL PARAMETERS'/ MAT00440
     &' ------------------------------------'//                        MAT00450
     &'  MAT.',3X,'   HA',6X,'   VA',6X,'MAX DUC', 6X,'  BETA')         MAT00460
  192 FORMAT (1X,I3,2X,6G12.6/)                                         MAT00470
C                                                                       MAT00480
      RETURN                                                            MAT00490
C                                                                       MAT00500
C------------------------ INITIALIZE STIFFNESS TERMS ------------------MAT00510
  200 CONTINUE                                                          MAT00520
C     ---- PINO-SUAREZ TOWER LONG GIRDER MODEL DATA STORAGE FOR MEMBER  MAT00530
C     SE( 1)- STIF    MEMBER MOMENT-ROTATION STIFFNESS                  MAT00540
C     SE( 2)- DP      LOAD INCREMENT                                    MAT00550
C     SE( 3)- DX      DISPLACEMENT INCREMENT                            MAT00560
C     SE( 4)- RV      INDEX FOR TOP OR BOTTOM ENVELOPE                  MAT00570
C                     RV=0 THEN HIGH=.TRUE.                             MAT00580
C                     RV=1 THEN HIGH=.FALSE.                            MAT00590
C     SE( 5)- RULE    RULE NUMBER                                       MAT00600
C     SE( 6)- PTOP                                                      MAT00610
C     SE( 7)- DTOP                                                      MAT00620
C     SE( 8)- PBOT                                                      MAT00630
C     SE( 9)- DBOT                                                      MAT00640
C     SE(10)- SPTOP                                                     MAT00650
C     SE(11)- SDTOP                                                     MAT00660
C     SE(12)- SPBOT                                                     MAT00670
C     SE(13)- SDBOT                                                     MAT00680
C     SE(14)- SSPTOP                                                    MAT00690
C     SE(15)- SSDTOP                                                    MAT00700
C     SE(16)- SSPBOT                                                    MAT00710
C     SE(17)- SSDBOT                                                    MAT00720
C     SE(18)- HB                                                        MAT00730
C     SE(19)- VB                                                        MAT00740
C     SE(20)- HD                                                        MAT00750
C     SE(21)- VD                                                        MAT00760
C     SE(22)- HE                                                        MAT00770
C     SE(23)- VE                                                        MAT00780
C     SE(24)- HAP                                                       MAT00790
C     SE(25)- VAP                                                       MAT00800
C     SE(26)- HBP                                                       MAT00810
C     SE(27)- VBP                                                       MAT00820
C     SE(28)- SSHD                                                      MAT00830
C     SE(29)- SSVD                                                      MAT00840
C     SE(30)- SSHE                                                      MAT00850
C     SE(31)- SSVE                                                      MAT00860
C     SE(32)- SSSPTP                                                    MAT00870
C     SE(33)- SSSDTP                                                    MAT00880
C     SE(34)- SSSPBT                                                    MAT00890
C     SE(35)- SSSDBT                                                    MAT00900
C     SE(36)- SHD                                                       MAT00910
C     SE(37)- SVD                                                       MAT00920
C     SE(38)- SHE                                                       MAT00930
C     SE(39)- SVE                                                       MAT00940
C     SE(40)- XEQ     UNLOADING EQUIVALENT STIFFNESS                    MAT00950
C     SE(41)- DUCT(1)                                                   MAT00960
C     SE(42)- DUCT(2)                                                   MAT00970
C     SE(43)- DUCT(3)                                                   MAT00980
C     SE(44)- EXCR(1)                                                   MAT00990
C     SE(45)- EXCR(2)                                                   MAT01000
C     SE(46)- EXCR(3)                                                   MAT01010
C     SE(47)- EXCR(4)                                                   MAT01020
C     SE(48)- EXCR(5)                                                   MAT01030
C     SE(49)- EXCR(6)                                                   MAT01040
C     SE(50)- DELY    YIELDING DISPLACEMENT                             MAT01050
C     SE(51)- EESE(1)                                                   MAT01060
C     SE(52)- EESE(2)                                                   MAT01070
C     SE(53)- EESE(3)                                                   MAT01080
C     SE(54)- EPSE                                                      MAT01090
C     SE(55)- PSEOLD                                                    MAT01100
C     SE(56)- CSE                                                       MAT01110
C     SE(57)- DMAX    ULTIMATE DISPLACEMENT                             MAT01120
C     SE(58)- D_MAX  MAXIMUM ELEMENT DISPLACEMENT FOR DAMAGE INDEX      MAT01130
C     SE(59)- P_MAX  ELEMENT LOAD CORRESPONDING TO D_MAX                MAT01140
C     SE(60)- DEMAX  CURRENT MAXIMUM AXIAL DISPLACEMENT FOR DUCTILITIES MAT01150
C     SE(61)- INITIAL STIFFNESS                                         MAT01160
C     SE(62)- DISP FACTOR FACH                                          MAT01170
C     SE(63)- MOMENT FACTOR FACV                                        MAT01180
C     SE(64)- ROTATION STIFFNESS FACTOR=FACV/FACH                       MAT01190
C     SE(65)- HA                                                        MAT01200
C     SE(66)- VA                                                        MAT01210
C ------------------------------------------------------------------    MAT01220
      DO 210 I=1,61                                                     MAT01230
  210   SE(I)=0                                                         MAT01240
      SE(50)=SMAT(1)                                                    MAT01250
      SE(5)=1                                                           MAT01260
      SE(65)= SMAT(1)                                                   MAT01270
      SE(66)= SMAT(2)                                                   MAT01280
      CALL HYST09(IOPT,SE(1),0 ,0.,0.,0 ,SE(3),SE(4),SE(5),SE(6),SE(7), MAT01290
     &     SE(8),SE(9),SE(10),SE(11),SE(12),SE(13),SE(14),SE(15),SE(16), MAT01300
     &     SE(17),SE(65),SE(66),SE(18),SE(19),SE(20),SE(21),SE(22),     MAT01310
     &     SE(23),SE(24),SE(25),SE(26),SE(27),SE(28),SE(29),SE(30),     MAT01320
     &     SE(31),SE(32),SE(33),SE(34),SE(35),SE(36),SE(37),SE(38),     MAT01330
     &     SE(39),SE(41),SE(60),SE(62),SE(63),SE(64),SMAT(3) )          MAT01340
      SE(56)=SE(50)*SMAT(2)/2                                           MAT01350
      SE(57)=SE(22)*SMAT(3)                                             MAT01360
      SE(58)=0                                                          MAT01370
      SE(59)=0                                                          MAT01380
      SE(61)=SE(1)                                                      MAT01390
C                                                                       MAT01400
      RETURN                                                            MAT01410
C------------------------ GET STIFFNESS TERMS ------------------------- MAT01420
  300 CONTINUE                                                          MAT01430
      SE(2)=P-PL                                                        MAT01440
      SE(3)=D-DL                                                        MAT01450
C                                                                       MAT01460
C-- CALL HYST09 TO CALC NEW STIFFNESS STIF                              MAT01470
      CALL HYST09(IOPT,SE(1),P,D,PL,DL,SE(3),SE(4),SE(5),SE(6),SE(7),   MAT01480
     &     SE(8),SE(9),SE(10),SE(11),SE(12),SE(13),SE(14),SE(15),SE(16), MAT01490
     &     SE(17),SE(65),SE(66),SE(18),SE(19),SE(20),SE(21),SE(22),     MAT01500
     &     SE(23),SE(24),SE(25),SE(26),SE(27),SE(28),SE(29),SE(30),     MAT01510
     &     SE(31),SE(32),SE(33),SE(34),SE(35),SE(36),SE(37),SE(38),     MAT01520
     &     SE(39),SE(41),SE(60),SE(62),SE(63),SE(64),SMAT(3) )          MAT01530
C                                                                       MAT01540
C --- STORE MAXIMUM DISPLACEMENT AND CORRESPONDING LOAD                 MAT01550
      IF(ABS(D) .GT. ABS(SE(58))) THEN                                  MAT01560
            SE(58)=D                                                    MAT01570
            SE(59)=P                                                    MAT01580
      ENDIF                                                             MAT01590
C                                                                       MAT01600
C--- GET ENERGIES, DUCTILITY AND EXCURSION FOR BOTH IOPT-3 AND 4 -----  MAT01610
  400 CONTINUE                                                          MAT01620
C                                                                       MAT01630
C --- TRANSFER ENERGIES FOR BOTH IOPT-3 AND 4                           MAT01640
      EESE=SE(51)                                                       MAT01650
      EPSE=SE(54)                                                       MAT01660
C                                                                       MAT01670
C --- TRANSFER DUCTILITIES AND EXCURSION RATIOS FOR BOTH IOPT-3 AND 4.  MAT01680
      DO 410 I=1,3                                                      MAT01690
        DUCT(I)=SE(40+I)                                                MAT01700
        EXCR(I+3)=SE(43+I+3)                                            MAT01710
  410   EXCR(I)=SE(43+I)                                                MAT01720
C                                                                       MAT01730
      RETURN                                                            MAT01740
C------------------- DAMAGE INDEX ------------------------------------ MAT01750
  500 CONTINUE                                                          MAT01760
C...DAMAGE INDEX IS NOT AVAILABLE                                       MAT01770
      DAMAGE=0.                                                         MAT01780
C                                                                       MAT01790
      RETURN                                                            MAT01800
      END                                                               MAT01810
C                                                                       MAT10010
C-----------------------------------------------------------------------MAT10020
C                                                                       MAT10030
      SUBROUTINE MAT10(IOPT,FIRST,MAT,SE,P,PP,D,DL,LELEM,LMAT,          MAT10040
     &                 SMAT,EESE,EPSE,DUCT,DAMAGE)                      MAT10050
C                                                                       MAT10060
      LOGICAL FIRST                                                     MAT10070
      DIMENSION SE(100),SMAT(9),DUCT(3)                                 MAT10080
      CHARACTER*80 TYPE                                                 MAT10090
C                                                                       MAT10100
C------- RESERVE STORAGE FOR ELEMENT                                    MAT10110
      LELEM=15                                                          MAT10120
      IF (IOPT.EQ.0) RETURN                                             MAT10130
C                                                                       MAT10140
C-----------------------------------------------------------------------MAT10150
C VARIABLES:                                                            MAT10160
C-----------------------------------------------------------------------MAT10170
C---------- GLOBAL VARIABLES                                           MAT10180
C IOPT   = 1, INITIALIZE MATERIAL                                       MAT10190
C        = 2, INITIALIZE STIFFNESS                                      MAT10200
C        = 3, GET STIFFNESS TERM                                        MAT10210
C        = 4, GET ENERGIES, DUCTILITY AND EXCURSION                     MAT10220
C        = 5, CALCULATE DAMAGE INDEX                                    MAT10230
C RINPUT - INPUT DATA                                                   MAT10240
C SMAT   - INTERNAL STORAGE                                             MAT10250
C SE     - OUTPUT STIFFNESS                                             MAT10260
C-----------------------------------------------------------------------MAT10270
      GO TO (100,200,300,400,500) IOPT                                  MAT10280
C                                                                       MAT10290
  100 CONTINUE                                                          MAT10300
C                                                                       MAT10310
C------ INTERPERTATE INPUT DATA                                         MAT10320
C     ELAS  - SMAT( 1)    : 0 FOR ELASTIC ANALYSIS                      MAT10330
C                         . 1 FOR INELASTIC ANALYSIS                    MAT10340
C                         . 2 FOR SIMULATING PINO-SUAREZ TOWER COLUMN   MAT10350
C                           LOCAL BUCKLING                              MAT10360
C                         : 3 FOR INELASTIC ANALYSIS AND CHECK BOX      MAT10370
C                           COLUMN INTERACTIVE STRENGH CRITERIA         MAT10380
C     SP    - SMAT( 2)    : RATIO FOR INELASTIC STIFFNESS TO INITIAL    MAT10390
C                           STIFFNESS                                   MAT10400
C     E     - SMAT( 3)    . ELASTIC MODULUS                             MAT10410
C     TI    - SMAT( 4)    . SECTION MOMENT OF INERTIA                   MAT10420
C     TMP   - SMAT( 5)    : PLASTIC MOMENT CAPACITY                     MAT10430
C     DUCMAX- SMAT( 6)    . FAILURE DUCTILITY                           MAT10440
C                           .GE. 1 FAILURE DUCTILITY CONSIDERED         MAT10450
C                           LT. 1 FAILURE DUCTILITY NOT CONSIDERED      MAT10460
C     BDAM  - SMAT( 7)    : PARAMETER FOR ANG'S DAMAGE INDEX            MAT10470
C     REDUCE- SMAT( 8)    : RESIDUAL STRENGTH FOR ELAS-2                MAT10480
C             SMAT( 9)    : DUMMY VARIABLE                              MAT10490
C-----------------------------------------------------------------------MAT10500
C                                                                       MAT10510
      READ(105,*) TYPE ,(SMAT(I),I=1,5),DUCMAX,BDAM                     MAT10520
      SMAT(6)=DUCMAX                                                    MAT10530
      SMAT(7)=BDAM                                                      MAT10540
      IF( SMAT(1) .EQ. 2 ) THEN                                         MAT10550
         READ(105,*) REDUCE                                             MAT10560
         SMAT(8)=REDUCE                                                 MAT10570
      ENDIF                                                             MAT10580
      LMAT = 9                                                          MAT10590
      IF (FIRST) WRITE(108,191)                                         MAT10600
      WRITE(108,192) MAT,(SMAT(K),K=1,5),SMAT(6),SMAT(7),REDUCE         MAT10610
  191 FORMAT (///' BILINEAR INTERACTIVE MATERIAL PROPERTIES'/           MAT10620
     &' ------------------------------------'//,'  MAT.',              MAT10630
     &  3X,'  ELAS',6X,'    SP',6X,'     E ',6X,'    TI',6X,          MAT10640
     &  '   MP',6X,'MAX DUC',6X,'  BETA',6X,'REDU F',/)                 MAT10650
  192 FORMAT (1X,I5,8G12.6/)                                            MAT10660
C                                                                       MAT10670
      RETURN                                                            MAT10680
C                                                                       MAT10690
C------------------------ INITIALIZE STIFFNESS TERMS ------------------MAT10700
  200 CONTINUE                                                          MAT10710
C     ---- BILINEAR INTERACTIVE MATERIAL DATA STORAGE FOR MEMBER ----   MAT10720
C     SE( 1)- EI      SECTION RIGIDITY                                  MAT10730
C     SE( 2)- P1      CURRENT YIELDING LOAD FOR PASITIVE LOAD           MAT10740
C     SE( 3)- P2      CURRENT YIELDING LOAD FOR NEGATIVE LOAD           MAT10750
C     SE( 4)- YIELD   0: ELASTIC,  1: INELASTIC                         MAT10760
C     SE( 5)- ELAS                                                      MAT10770
C     SE( 6)- SP                                                        MAT10780
C     SE( 7)- FAILURE DUCTILITY                                         MAT10790
C     SE( 8)- PARAMETER FOR ANG'S DAMAGE INDEX                          MAT10800
C     SE( 9)- RESIDUAL STRENGTH FOR ELAS-2                              MAT10810
C     SE(10)- CURRENT DUCTILITY                                         MAT10820
C     SE(11)- SMAT(4)                                                   MAT10830
C     SE(12)- TMP=SMAT(5)                                               MAT10840
C     SE(13)- DELY  YIELDING DISPLACEMENT                               MAT10850
C             FOR ROTATION SE(13)=TMP*LENGTH/(6EI)                      MAT10860
C             FOR AXIAL    SE(13)=TMP*LENGTH/EI                         MAT10870
C             FOR TORSION  SE(13)=TMP*LENGTH/EI                         MAT10880
C     SE(14)- DEMAX CURRENT MAXIMUM DISPLACEMENT FOR DUCTILITY          MAT10890
C     SE(15)- YIELDING DISPLACEMENT INDEX                               MAT10900
C             0- DELY HAS NOT BEEN DETERMINED                           MAT10910
C             1: DELY HAS BEEN DETERMINED                               MAT10920
C ----------------------------------------------------------------      MAT10930
      SE(1)= SMAT(3)*SMAT(4)                                            MAT10940
      SE(2)= SMAT(5)                                                    MAT10950
      SE(3)= -SMAT(5)                                                   MAT10960
      SE(4)= 0                                                          MAT10970
      SE(5)= SMAT(1)                                                    MAT10980
      SE(6)= SMAT(2)                                                    MAT10990
      SE(7)= SMAT(6)                                                    MAT11000
      SE(8)= SMAT(7)                                                    MAT11010
```

```
      SE(9)= SMAT(8)                                                    MAT11020
      SE(11)= SMAT(4)                                                   MAT11030
      SE(12)= SMAT(5)                                                   MAT11040
C                                                                       MAT11050
      RETURN                                                            MAT11060
C----------------------- GET STIFFNESS TERMS ------------------------  MAT11070
  300 CONTINUE                                                          MAT11080
C                                                                       MAT11090
C..CHECK FAILURE DUCTILITY                                              MAT11100
      DUCT(1)=SE(10)                                                    MAT11110
      IF( DUCT(1) .GE. SMAT(6) .AND. SMAT(6) .GE. 1) THEN               MAT11120
        P=0                                                             MAT11130
        YIELD=1                                                         MAT11140
        SE(4)=YIELD                                                     MAT11150
        RETURN                                                          MAT11160
      ENDIF                                                             MAT11170
      ELAS  = SMAT( 1)                                                  MAT11180
      SP    = SMAT( 2)                                                  MAT11190
      E     = SMAT( 3)                                                  MAT11200
      TI    = SMAT( 4)                                                  MAT11210
      TMP   = SMAT( 5)                                                  MAT11220
      P1    = SE(2)                                                     MAT11230
      P2    = SE(3)                                                     MAT11240
      YIELD = SE(4)                                                     MAT11250
      IF( ELAS .EQ. 0 ) THEN                                            MAT11260
        YIELD = 0                                                       MAT11270
        SE(4)=YIELD                                                     MAT11280
        SE(14)=0                                                        MAT11290
        DUCT(1)=0.                                                      MAT11300
        RETURN                                                          MAT11310
      ENDIF                                                             MAT11320
      IF( ELAS .EQ. 2 ) THEN                                            MAT11330
C ** FOR ELASTIC REGION                                                 MAT11340
        IF( YIELD .EQ. 0 ) THEN                                         MAT11350
          IF( P .GE. P1 .OR. P .LE. P2 ) THEN                           MAT11360
            YIELD=1                                                     MAT11370
            IF( P .GE. P1 ) P=P1                                        MAT11380
            IF( P .LE. P2 ) P=P2                                        MAT11390
C ---------- DETERMINE ELEMENT DUCTILITY UNIT                           MAT11400
C            NOTE: ROTATION DUCTILITY UNIT IS DEFINED WHEN              MAT11410
C            SE(15)=1, IF (P .GE. TMP .AND. D .GE. TMP*LENGTH/(6EI))    MAT11420
C            ELSE SE(15)=0                                              MAT11430
C                                                                       MAT11440
            IF (SE(15).EQ.0.0.AND.ABS(D).GE.ABS(SE(13))) SE(15)=1       MAT11450
          ELSE IF( P .LT. P1 .AND. P .GT. P2 ) THEN                     MAT11460
            YIELD=0                                                     MAT11470
          ENDIF                                                         MAT11480
C ** FOR INELASTIC REGION                                               MAT11490
        ELSE IF( YIELD .EQ. 1 ) THEN                                    MAT11500
          IF( P .GE. P1 .OR. P .LE. P2 ) THEN                           MAT11510
            YIELD=1                                                     MAT11520
            IF( P .GE. P1 ) P=P1                                        MAT11530
            IF( P .LE. P2 ) P=P2                                        MAT11540
C ---------- DETERMINE ELEMENT DUCTILITY UNIT                           MAT11550
C            NOTE: ROTATION DUCTILITY UNIT IS DEFINED WHEN              MAT11560
C            SE(15)=1, IF (P .GE. TMP .AND. D .GE. TMP*LENGTH/(6EI))    MAT11570
C            ELSE SE(15)=0                                              MAT11580
C                                                                       MAT11590
            IF (SE(15).EQ.0.0.AND ABS(D).GE ABS(SE(13))) SE(15)=1       MAT11600
          ELSE IF( P .LT. P1 .AND. PP .GE. 0 ) THEN                     MAT11610
            YIELD=0                                                     MAT11620
C           *** UNBALANCED FORCE ***                                    MAT11630
            IF( (D-DL) .NE. 0 ) THEN                                    MAT11640
              SM2=(P-PP)/(D-DL)                                         MAT11650
              SM1=SM2/SP                                                MAT11660
              UD=D-DL                                                   MAT11670
              UP=SM1*UD                                                 MAT11680
              P=PP+UP                                                   MAT11690
              IF( P .LE. P2 ) THEN                                      MAT11700
                P=P2                                                    MAT11710
                YIELD=1                                                 MAT11720
              ENDIF                                                     MAT11730
            ENDIF                                                       MAT11740
          ELSE IF( P .GT. P2 .AND. PP .LE. 0 ) THEN                     MAT11750
            YIELD=0                                                     MAT11760
C           *** UNBALANCED FORCE ***                                    MAT11770
            IF( (D-DL) .NE. 0 ) THEN                                    MAT11780
              SM2=(P-PP)/(D-DL)                                         MAT11790
              SM1=SM2/SP                                                MAT11800
              UD=D-DL                                                   MAT11810
              UP=SM1*UD                                                 MAT11820
              P=PP-UP                                                   MAT11830
              IF( P .GE. P1 ) THEN                                      MAT11840
                P=P1                                                    MAT11850
                YIELD=1                                                 MAT11860
              ENDIF                                                     MAT11870
            ENDIF                                                       MAT11880
          ENDIF                                                         MAT11890
        ENDIF                                                           MAT11900
        GO TO 399                                                       MAT11910
      ENDIF                                                             MAT11920
C                                                                       MAT11930
C...  FOR ELAS=1 AND 3 CASES                                            MAT11940
C --- ELASTIC REGION                                                    MAT11950
      IF(YIELD .EQ. 0 ) THEN                                            MAT11960
        IF( P .GE. P1 .OR. P .LE. P2 ) THEN                             MAT11970
          YIELD=1                                                       MAT11980
C         *** UNBALANCED FORCE ***                                      MAT11990
          IF( P .GE. P1 ) THEN                                          MAT12000
            IF( (D-DL) .NE. 0 .AND. (P-PP) .NE. 0) THEN                 MAT12010
              SM1=(P-PP)/(D-DL)                                         MAT12020
C             /* SM2 IS ESTIMATED VALUE */                              MAT12030
              SM2=SM1*SP                                                MAT12040
              UP1=P-P1                                                  MAT12050
              UD=UP1/SM1                                                MAT12060
              UP2= UD*SM2                                               MAT12070
              P=P-(UP1+UP2)                                             MAT12080
            ENDIF                                                       MAT12090
          ELSE IF( P .LE. P2 ) THEN                                     MAT12100
            IF( (D DL) .NE. 0 AND (P-PP) NE 0) THEN                     MAT12110
              SM1=(P-PP)/(D-DL)                                         MAT12120
C             /* SM2 IS ESTIMATED VALUE */                              MAT12130
              SM2=SM1*SP                                                MAT12140
              UP1=P-P2                                                  MAT12150
              UD=UP1/SM1                                                MAT12160
              UP2=-UD*SM2                                               MAT12170
              P=P-(UP1+UP2)                                             MAT12180
            ENDIF                                                       MAT12190
          ENDIF                                                         MAT12200
C ---------- DETERMINE ELEMENT DUCTILITY UNIT                           MAT12210
C            NOTE: ROTATION DUCTILITY UNIT IS DEFINED WHEN              MAT12220
C            SE(15)=1, IF (P .GE. TMP .AND. D .GE. TMP*LENGTH/(6EI))    MAT12230
C            ELSE SE(15)=0                                              MAT12240
C                                                                       MAT12250
          IF (SE(15).EQ.0.0.AND ABS(D).GE ABS(SE(13))) SE(15)=1         MAT12260
        ELSE                                                            MAT12270
          YIELD=0                                                       MAT12280
        ENDIF                                                           MAT12290
        GO TO 399                                                       MAT12300
      ENDIF                                                             MAT12310
C                                                                       MAT12320
C ... INELASTIC REGION                                                  MAT12330
      IF(YIELD .EQ. 1 ) THEN                                            MAT12340
        IF( P .GT. P1 ) THEN                                            MAT12350
          IF( P.GE. PP) THEN                                            MAT12360
            YIELD=1                                                     MAT12370
            IF( SE(15) .EQ. 0 ) THEN                                    MAT12380
              IF( ABS(D) .GE. ABS(SE(13))) THEN                         MAT12390
                SE(13)=ABS(D)                                           MAT12400
                SE(15)=1                                                MAT12410
              ENDIF                                                     MAT12420
            ENDIF                                                       MAT12430
          ELSE IF( P LT PP) THEN                                        MAT12440
            YIELD=0                                                     MAT12450
            P1=PP                                                       MAT12460
            P2=PP-2*TMP                                                 MAT12470
C                                                                       MAT12480
C                 *** UNBALANCED FORCE ***                              MAT12480
                  IF( (D-DL) .NE. 0 ) THEN                              MAT12490
                    SM2=(P-PP)/(D-DL)                                   MAT12500
                    SM1=SM2/SP                                          MAT12510
                    UD=D-DL                                             MAT12520
                    UP=SM1*UD                                           MAT12530
                    P=PP+UP                                             MAT12540
                  ENDIF                                                 MAT12550
                ENDIF                                                   MAT12560
          ELSE IF( P .LT. P2 ) THEN                                     MAT12570
            IF( P.LE. PP) THEN                                          MAT12580
              YIELD=1                                                   MAT12590
              IF( SE(15) .EQ. 0 ) THEN                                  MAT12600
                IF( ABS(D) .GE. ABS(SE(13))) THEN                       MAT12610
                  SE(13)=ABS(D)                                         MAT12620
                  SE(15)=1                                              MAT12630
                ENDIF                                                   MAT12640
              ENDIF                                                     MAT12650
            ELSE IF(P.GT. PP) THEN                                      MAT12660
              YIELD=0                                                   MAT12670
              P2=PP                                                     MAT12680
              P1=PP+2*TMP                                               MAT12690
C             *** UNBALANCED FORCE ***                                  MAT12700
              IF( (D-DL) .NE. 0 ) THEN                                  MAT12710
                SM2=(P-PP)/(D-DL)                                       MAT12720
                SM1=SM2/SP                                              MAT12730
                UD=D-DL                                                 MAT12740
                UP=SM1*UD                                               MAT12750
                P=PP+UP                                                 MAT12760
              ENDIF                                                     MAT12770
            ENDIF                                                       MAT12780
          ENDIF                                                         MAT12790
        ENDIF                                                           MAT12800
  399 SE(2)=P1                                                          MAT12810
      SE(3)=P2                                                          MAT12820
      SE(4)=YIELD                                                       MAT12830
      IF( SE(15) .EQ. 1 ) THEN                                          MAT12840
        IF( SE(14) .LE. ABS(D)) THEN                                    MAT12850
          SE(14)=ABS(D)                                                 MAT12860
          DUCT(1)=SE(14)/SE(13)                                         MAT12870
          SE(10)=DUCT(1)                                                MAT12880
        ENDIF                                                           MAT12890
      ELSE IF( SE(15) .EQ. 0 ) THEN                                     MAT12900
        DUCT(1)=0                                                       MAT12910
        SE(10)=DUCT(1)                                                  MAT12920
      ENDIF                                                             MAT12930
C                                                                       MAT12940
C-------------- GET DUCTILITY IOPT=3 AND 4 ---------------------------MAT12950
  400 CONTINUE                                                          MAT12960
C                                                                       MAT12970
C --- TRANSFER ENERGIES FOR BOTH IOPT=3 AND 4.                          MAT12980
      EESE=0                                                            MAT12990
      EPSE=0                                                            MAT13000
      ELAS = SMAT(1)                                                    MAT13010
C                                                                       MAT13020
C --- TRANSFER DUCTILITIES AND EXCURSION RATIOS FOR BOTH IOPT=3 AND 4   MAT13030
C                                                                       MAT13040
      IF( ELAS .EQ. 0 ) THEN                                            MAT13050
          DUCT(1)=0                                                     MAT13060
          SE(10)=DUCT(1)                                                MAT13070
      ELSE IF( ELAS .NE. 0 ) THEN                                       MAT13080
            IF( SE(15) .EQ. 0 ) THEN                                    MAT13090
              DUCT(1)=0                                                 MAT13100
              SE(10)=DUCT(1)                                            MAT13110
            ELSE IF( SE(15) .EQ. 1 ) THEN                               MAT13120
              DUCT(1)=SE(14)/SE(13)                                     MAT13130
              SE(10)=DUCT(1)                                            MAT13140
            ENDIF                                                       MAT13150
      ENDIF                                                             MAT13160
C                                                                       MAT13170
      RETURN                                                            MAT13180
C---------------- DAMAGE INDEX ------------------------------------    MAT13190
  500 CONTINUE                                                          MAT13200
C..DAMAGE INDEX IS NOT AVAILABLE                                        MAT13210
      DAMAGE=0                                                          MAT13220
C                                                                       MAT13230
      RETURN                                                            MAT13240
      END                                                               MAT13250
C                                                                       MAT10010
C-----------------------------------------------------------------------MAT10020
      SUBROUTINE MAT12(IOPT,FIRST,MAT,SE,P,D,DL,LELEM,LMAT,IMAT,        MAT10030
     &                 SMAT,EESE,EPSE,DUCT,EXCR,DAMAGE)                 MAT10040
C                                                                       MAT10050
      LOGICAL FIRST                                                     MAT10060
      DIMENSION SE(100),SMAT(LMAT),DUCT(3),EXCR(6),EESAT(12,12)         MAT10070
      DIMENSION A(12,12),SAT(12,12),P(12),D(12),DL(12),IMAT(LMAT)       MAT10080
      REAL INEB,INEH                                                    MAT10090
      CHARACTER*80 TYPE                                                 MAT10100
C                                                                       MAT10110
C------- RESERVE STORAGE FOR ELEMENT                                    MAT10120
      LELEM= 99                                                         MAT10130
      IF (IOPT EQ 0) RETURN                                             MAT10140
C                                                                       MAT10150
C..................................... ..................................MAT10160
C VARIABLES:                                                            MAT10170
C..... ..... ..... ..... ..... ..... ..... ..... ..... ..... ..... ....MAT10180
C---------- GLOBAL VARIABLES                                           MAT10190
C IOPT   = 1, INITIALIZE MATERIAL                                       MAT10200
C        = 2, INITIALIZE STIFFNESS                                      MAT10210
C        = 3, GET STIFFNESS TERM                                        MAT10220
C        = 4, GET ENERGIES, DUCTILITY AND EXCURSION                     MAT10230
C        = 5, CALCULATE DAMAGE INDEX                                    MAT10240
C RINPUT = INPUT DATA                                                   MAT10250
C SMAT   = INTERNAL STORAGE                                             MAT10260
C SE     = OUTPUT STIFFNESS                                             MAT10270
C-----------------------------------------------------------------------MAT10280
      GO TO (100,200,200,400,400) IOPT                                  MAT10290
C                                                                       MAT10300
  100 CONTINUE                                                          MAT10310
C                                                                       MAT10320
C----- INTERPERTATE INPUT DATA                                          MAT10330
C     NSEG  = SMAT( 1)                                                  MAT10340
C     YS    = SMAT( 2)                                                  MAT10350
C     EM    = SMAT( 3)                                                  MAT10360
C     LIBN  = SMAT( 4)                                                  MAT10370
C     HH    = SMAT( 5)                                                  MAT10380
C     UU    = SMAT( 6)                                                  MAT10390
C     WW    = SMAT( 7)                                                  MAT10400
C     ZZ    = SMAT( 8)                                                  MAT10410
C     INEB  = SMAT( 9)                                                  MAT10420
C     INEH  = SMAT(10)                                                  MAT10430
C     ST    = SMAT(11)                                                  MAT10440
C     IREV1 = SMAT(12)                                                  MAT10450
C     IREV2 = SMAT(13)                                                  MAT10460
C     IREV3 = SMAT(14)                                                  MAT10470
C     IREV4 = SMAT(15)                                                  MAT10480
C     IECOP = SMAT(16)                                                  MAT10490
C     NUMP  = SMAT(17)                                                  MAT10500
C     SMALL = SMAT(18)                                                  MAT10510
C     RATIX0= SMAT(19)                                                  MAT10520
C     RATIY0= SMAT(20)                                                  MAT10530
C     TOTA  = SMAT(21)                                                  MAT10540
C     IAUTO = SMAT(22)                                                  MAT10550
C     IMATER= SMAT(23)                                                  MAT10560
C     RATIOS= SMAT(24)                                                  MAT10570
C     IR    = SMAT(25)                                                  MAT10580
C     G     = SMAT(26)                                                  MAT10590
C     QRNEE = SMAT(27)                                                  MAT10600
C     ISTIF = SMAT(28)                                                  MAT10610
C     LMAT  = SMAT(29)                                                  MAT10620
C                                                                       MAT10630
      READ(105,*) TYPE ,NSEG,SMAT(2),SMAT(3),LIBN,(SMAT(I),I=5,8),      MAT10640
     &            INEB,INEH,SMAT(11),IREV1,IREV2,IREV3,IREV4,IECOP,     MAT10650
     &            (SMAT(I),I=18,21),IAUTO,IMATER,SMAT(24),IR,SMAT(26),  MAT10660
     &            SMAT(27),ISTIF                                        MAT10670
      SMAT(1)=NSEG                                                      MAT10680
```

```
      SMAT(4)=LIBN                                                      MAT10690
      SMAT(9)=INEB                                                      MAT10700
      SMAT(10)=INEH                                                     MAT10710
      SMAT(12)=IREV1                                                    MAT10720
      SMAT(13)=IREV2                                                    MAT10730
      SMAT(14)=IREV3                                                    MAT10740
      SMAT(15)=IREV4                                                    MAT10750
      SMAT(16)=IECOP                                                    MAT10760
      SMAT(22)=IAUTO                                                    MAT10770
      SMAT(23)=IMATER                                                   MAT10780
      SMAT(25)=IR                                                       MAT10790
      SMAT(26)=ISTIF                                                    MAT10800
C                                                                       MAT10810
C                                                                       MAT10820
      IF( FIRST ) WRITE(108,161)                                        MAT10830
        IF( LIBN .EQ. 1 ) THEN                                          MAT10840
            NUMP=4*(INEB-1NEH)                                          MAT10850
            WRITE(108,191) MAT,NSEG,SMAT(2),SMAT(3),LIBN,               MAT10860
     &           (SMAT(I),I=5,8),INEB,INEH,SMAT(11),IREV1,IREV2,        MAT10870
     &           IREV3,IREV4,IECOP,NUMP,(SMAT(I),I=18,21),IAUTO,        MAT10880
     &           IMATER,SMAT(24),IR,SMAT(26),SMAT(27),ISTIF             MAT10890
        ELSE IF( LIBN .EQ. 2 ) THEN                                     MAT10900
            NUMP=6*INEB                                                 MAT10910
            WRITE(108,192) MAT,NSEG,SMAT(2),SMAT(3),LIBN,               MAT10920
     &           (SMAT(I),I=5,8),INEB,INEH,SMAT(11),IREV1,IREV2,        MAT10930
     &           IREV3,IREV4,IECOP,NUMP,(SMAT(I),I=18,21),IAUTO,        MAT10940
     &           IMATER,SMAT(24),IR,SMAT(26),SMAT(27),ISTIF             MAT10950
        ELSE IF( LIBN .EQ. 3 ) THEN                                     MAT10960
            NUMP=4*INEB*INEH                                            MAT10970
            WRITE(108,194) MAT,NSEG,SMAT(2),SMAT(3),LIBN,               MAT10980
     &           (SMAT(I),I=5,8),INEB,INEH,SMAT(11),IREV1,IREV2,        MAT10990
     &           IREV3,IREV4,IECOP,NUMP,(SMAT(I),I=18,21),IAUTO,        MAT11000
     &           IMATER,SMAT(24),IR,SMAT(26),SMAT(27),ISTIF             MAT11010
        ELSE IF( LIBN .EQ. 4 ) THEN                                     MAT11020
            NUMP=IREV1*IREV2*2 + IREV3*IREV4                            MAT11030
            WRITE(108,196) MAT,NSEG,SMAT(2),SMAT(3),LIBN,               MAT11040
     &           (SMAT(I),I=5,8),INEB,INEH,SMAT(11),IREV1,IREV2,        MAT11050
     &           IREV3,IREV4,IECOP,NUMP,(SMAT(I),I=18,21),IAUTO,        MAT11060
     &           IMATER,SMAT(24),IR,SMAT(26),SMAT(27),ISTIF             MAT11070
        ELSE IF( LIBN .EQ. 6 ) THEN                                     MAT11080
            NUMP=8*INEB                                                 MAT11090
            WRITE(108,192) MAT,NSEG,SMAT(2),SMAT(3),LIBN,               MAT11100
     &           (SMAT(I),I=5,8),INEB,INEH,SMAT(11),IREV1,IREV2,        MAT11110
     &           IREV3,IREV4,IECOP,NUMP,(SMAT(I),I=18,21),IAUTO,        MAT11120
     &           IMATER,SMAT(24),IR,SMAT(26),SMAT(27),ISTIF             MAT11130
        ELSE IF( LIBN .EQ. 7 ) THEN                                     MAT11140
            NUMP=IREV1*IREV2 + IREV3*IREV4                              MAT11150
            WRITE(108,193) MAT,NSEG,SMAT(2),SMAT(3),LIBN,               MAT11160
     &           (SMAT(I),I=5,8),INEB,INEH,SMAT(11),IREV1,IREV2,        MAT11170
     &           IREV3,IREV4,IECOP,NUMP,(SMAT(I),I=18,21),IAUTO,        MAT11180
     &           IMATER,SMAT(24),IR,SMAT(26),SMAT(27),ISTIF             MAT11190
        ELSE IF( LIBN .EQ. 8 ) THEN                                     MAT11200
            NUMP=IREV1*IREV2 + IREV3*IREV4                              MAT11210
            WRITE(108,193) MAT,NSEG,SMAT(2),SMAT(3),LIBN,               MAT11220
     &           (SMAT(I),I=5,8),INEB,INEH,SMAT(11),IREV1,IREV2,        MAT11230
     &           IREV3,IREV4,IECOP,NUMP,(SMAT(I),I=18,21),IAUTO,        MAT11240
     &           IMATER,SMAT(24),IR,SMAT(26),SMAT(27),ISTIF             MAT11250
        ELSE IF( LIBN .EQ. 9 ) THEN                                     MAT11260
            NUMP=IREV1*IREV2 + IREV3*IREV4 + INEB*INEH                  MAT11270
            WRITE(108,195) MAT,NSEG,SMAT(2),SMAT(3),LIBN,               MAT11280
     &           (SMAT(I),I=5,8),INEB,INEH,SMAT(11),IREV1,IREV2,        MAT11290
     &           IREV3,IREV4,IECOP,NUMP,(SMAT(I),I=18,21),IAUTO,        MAT11300
     &           IMATER,SMAT(24),IR,SMAT(26),SMAT(27),ISTIF             MAT11310
        ELSE                                                            MAT11320
            WRITE(108,199) MAT,SMAT(4)                                  MAT11330
        ENDIF                                                           MAT11340
        SMAT(17)=NUMP                                                   MAT11350
C                                                                       MAT11360
 181  FORMAT (///' STABILITY ELEMENT'/                                  MAT11370
     &        ' ----------------'//)                                    MAT11380
 191  FORMAT (1X,'MAT  NO      ='',I3,2X,' BOX SECTION'/                MAT11390
     &        1X,'NSEG         ='',1P,I10  ,9X,'YS          ='',1P,G10.3/  MAT11400
     &        1X,'EM           ='',1P,G10.3,9X,'LIBN        ='',1P,I10  /  MAT11410
     &        1X,'BOX HEIGTH   ='',1P,G10.3,9X,'BOX WILTH   ='',1P,G10.3/  MAT11420
     &        1X,'ECCX0        ='',1P,G10.3,9X,'ECCY0       ='',1P,G10.3/  MAT11430
     &        1X,'H.W.SEC.NO.  ='',1P,G10.3,9X,'H.H.SEC.NO. ='',1P,G10.3/  MAT11440
     &        1X,'THICKNESS    ='',1P,G10.3,9X,'L.B. FLAG   ='',1P,I10  /  MAT11450
     &        1X,'IREV2        ='',1P,I10  ,9X,'IREV3       ='',1P,I10  /  MAT11460
     &        1X,'IREV4        ='',1P,I10  ,9X,'IECOP       ='',1P,I10  /  MAT11470
     &        1X,'NUMP         ='',1P,I10  ,9X,'SMALL       ='',1P,G10.3/  MAT11480
     &        1X,'RATIX0       ='',1P,G10.3,9X,'RATIY0      ='',1P,G10.3/  MAT11490
     &        1X,'TOTA         ='',1P,G10.3,9X,'IAUTO       ='',1P,I10  /  MAT11500
     &        1X,'IMATER       ='',1P,I10  ,9X,'RATIO3      ='',1P,G10.3/  MAT11510
     &        1X,'IR           ='',1P,I10  ,9X,'G           ='',1P,G10.3/  MAT11520
     &      1X,'QRNEE       ='',1P,G10 3,9X,'ISTIF       ='',1P,I10  //)  MAT11530
C                                                                       MAT11540
 192  FORMAT (1X,'MAT. NO.     ='',I3,2X,' TUBE SECTION'/                MAT11550
     &        1X,'NSEG         ='',1P,I10  ,9X,'YS          ='',1P,G10 3/  MAT11560
     &        1X,'EM           ='',1P,G10.3,9X,'LIBN        ='',1P,I10  /  MAT11570
     &        1X,'RADIUS       ='',1P,G10 3,9X,'DUMMY VAR.  ='',1P,G10.3/  MAT11580
     &        1X,'ECCX0        ='',1P,G10.3,9X,'ECCY0       ='',1P,G10 3/  MAT11590
     &        1X,'1/4.SEC.NO   ='',1P,G10.3,9X,'DUMMY VAR.  ='',1P,G10 3/  MAT11600
     &        1X,'THICKNESS    ='',1P,G10.3,9X,'IREV1       ='',1P,I10  /  MAT11610
     &        1X,'IREV2        ='',1P,I10  ,9X,'IREV3       ='',1P,I10  /  MAT11620
     &        1X,'IREV4        ='',1P,I10  ,9X,'IECOP       ='',1P,I10  /  MAT11630
     &        1X,'NUMP         ='',1P,I10  ,9X,'SMALL       ='',1P,G10.3/  MAT11640
     &        1X,'RATIX0       ='',1P,G10.3,9X,'RATIY0      ='',1P,G10.3/  MAT11650
     &        1X,'TOTA         ='',1P,G10.3,9X,'IAUTO       ='',1P,I10  /  MAT11660
     &        1X,'IMATER       ='',1P,I10  ,9X,'RATIO3      ='',1P,G10.3/  MAT11670
     &        1X,'IR           ='',1P,I10  ,9X,'G           ='',1P,G10.3/  MAT11680
     &      1X,'QRNEE       ='',1P,G10.3,9X,'ISTIF       ='',1P,I10  //)  MAT11690
C                                                                       MAT11700
 193  FORMAT (1X,'MAT. NO      ='',I3,2X,' ANGLE SECTION'/               MAT11710
     &        1X,'NSEG         ='',1P,I10  ,9X,'YS          ='',1P,G10.3/  MAT11720
     &        1X,'EM           ='',1P,G10 3,9X,'LIBN        ='',1P,I10  /  MAT11730
     &        1X,'LAG LENGTH   ='',1P,G10.3,9X,'DUMMY VAR.  ='',1P,G10 3/  MAT11740
     &        1X,'ECCX0        ='',1P,G10.3,9X,'ECCY0       ='',1P,G10.3/  MAT11750
     &        1X,'LAG.SEC NO.  ='',1P,G10.3,9X,'DUMMY VAR.  ='',1P,G10.3/  MAT11760
     &        1X,'THICKNESS    ='',1P,G10.3,9X,'N.R.IN U DIR='',1P,I10  /  MAT11770
     &        1X,'N.C IN U DIR='',1P,I10  ,9X,'N.R .IN V DIR='',1P,I10  /  MAT11780
     &        1X,'N C.IN V DIR='',1P,I10  ,9X,'IECOP       ='',1P,I10  /  MAT11790
     &        1X,'NUMP         ='',1P,I10  ,9X,'SMALL       ='',1P,G10.3/  MAT11800
     &        1X,'RATIX0       ='',1P,G10.3,9X,'RATIY0      ='',1P,G10.3/  MAT11810
     &        1X,'TOTA         ='',1P,G10 3,9X,'IAUTO       ='',1P,I10  /  MAT11820
     &        1X,'IMATER       ='',1P,I10  ,9X,'RATIO3      ='',1P,G10 3/  MAT11830
     &        1X,'IR           ='',1P,I10  ,9X,'G           ='',1P,G10.3/  MAT11840
     &      1X,'QRNEE       ='',1P,G10.3,9X,'ISTIF       ='',1P,I10  //)  MAT11850
C                                                                       MAT11860
 194  FORMAT (1X,'MAT  NO      ='',I3,2X,' RECTANGULAR SECTION'/         MAT11870
     &        1X,'NSEG         ='',1P,I10  ,9X,'YS          ='',1P,G10.3/  MAT11880
     &        1X,'EM           ='',1P,G10.3,9X,'LIBN        ='',1P,I10  /  MAT11890
     &        1X,'SEC HEIGTH   ='',1P,G10 3,9X,'SEC WILTH   ='',1P,G10 3/  MAT11900
     &        1X,'ECCX0        ='',1P,G10.3,9X,'ECCY0       ='',1P,G10.3/  MAT11910
     &        1X,'UU SEC.NO.   ='',1P,G10.3,9X,'HH SEC.NO.  ='',1P,G10.3/  MAT11920
     &        1X,'DUmmy VAR.   ='',1P,G10 3,9X,'IREV1       ='',1P,I10  /  MAT11930
     &        1X,'IREV2        ='',1P,I10  ,9X,'IREV3       ='',1P,I10  /  MAT11940
     &        1X,'IREV4        ='',1P,I10  ,9X,'IECOP       ='',1P,I10  /  MAT11950
     &        1X,'NUMP         ='',1P,I10  ,9X,'SMALL       ='',1P,G10.3/  MAT11960
     &        1X,'RATIX0       ='',1P,G10.3,9X,'RATIY0      ='',1P,G10.3/  MAT11970
     &        1X,'TOTA         ='',1P,G10.3,9X,'IAUTO       ='',1P,I10  /  MAT11980
     &        1X,'IMATER       ='',1P,I10  ,9X,'RATIO3      ='',1P,G10.3/  MAT11990
     &        1X,'IR           ='',1P,I10  ,9X,'G           ='',1P,G10.3/  MAT12000
     &      1X,'QRNEE       ='',1P,G10.3,9X,'ISTIF       ='',1P,I10  //)  MAT12010
C                                                                       MAT12020
 195  FORMAT (1X,'MAT. NO      ='',I3,2X,' ANGLE WITH COVER PL '/        MAT12030
     &        1X,'NSEG         ='',1P,I10  ,9X,'YS          ='',1P,G10.3/  MAT12040
     &        1X,'EM           ='',1P,G10.3,9X,'LIBN        ='',1P,I10  /  MAT12050
     &        1X,'LAG LENGTH   ='',1P,G10 3,9X,'COV WIDTH   ='',1P,G10 3/  MAT12060
     &        1X,'COV.THICKNES='',1P,G10.3,9X,'DUMMY VAR    ='',1P,G10 3/  MAT12070
     &        1X,'N.R. COV.    ='',1P,G10 3,9X,'N C. COV    ='',1P,G10 3/  MAT12080
     &        1X,'THICKNESS    ='',1P,G10.3,9X,'N.R IN U DIR='',1P,I10  /  MAT12090
     &        1X,'N.C.IN U DIR='',1P,I10  ,9X,'N.R .IN V DIR='',1P,I10  /  MAT12100
     &        1X,'N C IN V DIR='',1P,I10  ,9X,'IECOP       ='',1P,I10  /  MAT12110
     &        1X,'NUMP         ='',1P,I10  ,9X,'SMALL       ='',1P,G10.3/  MAT12120
     &        1X,'RATIX0       ='',1P,G10.3,9X,'RATIY0      ='',1P,G10 3/  MAT12130
     &        1X,'TOTA         ='',1P,G10 3,9X,'IAUTO       ='',1P,I10  /  MAT12140
     &        1X,'IMATER       ='',1P,I10  ,9X,'RATIO3      ='',1P,G10.3/  MAT12150
     &        1X,'IR           ='',1P,I10  ,9X,'G           ='',1P,G10.3/  MAT12160
     &      1X,'QRNEE       ='',1P,G10 3,9X,'ISTIF       ='',1P,I10  //)  MAT12170
C                                                                       MAT12180
 196  FORMAT (1X,'MAT  NO.     ='',I3,2X,' WIDE-FLANGE SECTION'/         MAT12190
     &        1X,'NSEG         ='',1P,I10  ,9X,'YS          ='',1P,G10.3/  MAT12200
     &        1X,'EM           ='',1P,G10.3,9X,'LIBN        ='',1P,I10  /  MAT12210
     &        1X,'BOX HEIGTH   ='',1P,G10.3,9X,'BOX WILTH   ='',1P,G10.3/  MAT12220
     &        1X,'ECCX0        ='',1P,G10.3,9X,'ECCY0       ='',1P,G10.3/  MAT12230
     &        1X,'WEB THICK.   ='',1P,G10.3,9X,'DUMMY VAR   ='',1P,G10.3/  MAT12240
     &        1X,'FLA THICK.   ='',1P,G10.3,9X,'IREV1       ='',1P,I10  /  MAT12250
     &        1X,'IREV2        ='',1P,I10  ,9X,'IREV3       ='',1P,I10  /  MAT12260
     &        1X,'IREV4        ='',1P,I10  ,9X,'IECOP       ='',1P,I10  /  MAT12270
     &        1X,'NUMP         ='',1P,I10  ,9X,'SMALL       ='',1P,G10.3/  MAT12280
     &        1X,'RATIX0       ='',1P,G10.3,9X,'RATIY0      ='',1P,G10.3/  MAT12290
     &        1X,'TOTA         ='',1P,G10.3,9X,'IAUTO       ='',1P,I10  /  MAT12300
     &        1X,'IMATER       ='',1P,I10  ,9X,'RATIO3      ='',1P,G10.3/  MAT12310
     &        1X,'IR           ='',1P,I10  ,9X,'G           ='',1P,G10 3/  MAT12320
     &      1X,'QRNEE       ='',1P,G10.3,9X,'ISTIF       ='',1P,I10  //)  MAT12330
C                                                                       MAT12340
 199  FORMAT (1X,'MAT. NO.     ='',I3,' IS NOT AVAILABLE',               MAT12350
     &        ' LIBN- ',G10.3,' CHECK LIBN '/)                          MAT12360
C                                                                       MAT12370
      K00=1                                                             MAT12380
110   I1=29*1                                                           MAT12390
      I2=I1+NSEG*12                                                     MAT12400
      I3=I2+(NSEG+1)*6                                                  MAT12410
      I4=I3+NSEG                                                        MAT12420
      I5=I4+2                                                           MAT12430
      I6=I5*2                                                           MAT12440
      I7=I6+NSEG*NUMP                                                   MAT12450
      I8=I7+NSEG*NUMP                                                   MAT12460
      I9=I8+NUMP                                                        MAT12470
      I10=I9+NUMP                                                       MAT12480
      I11=I10+NSEG*2*3                                                  MAT12490
      I12=I11+NSEG                                                      MAT12500
      I13=I12+(NSEG+1)*6                                                MAT12510
      I14=I13+(NSEG+1)*6                                                MAT12520
      I15=I14+(NSEG+1)*6                                                MAT12530
      I16=I15+NSEG                                                      MAT12540
      I17=I16+NSEG*12                                                   MAT12550
      I18=I17+NSEG*12                                                   MAT12560
      I19=I18+NSEG*12                                                   MAT12570
      I20=I19+NSEG*12                                                   MAT12580
      I21=I20+NSEG*NUMP                                                 MAT12590
      I22=I21+NSEG*NUMP                                                 MAT12600
      I23=I22+NSEG*NUMP                                                 MAT12610
      I24=I23+NUMP                                                      MAT12620
      I25=I24*NSEG*9                                                    MAT12630
      I26=I25+NSEG*9                                                    MAT12640
      I27=I26+((NSEG+1)*6)**2                                           MAT12650
      I28=I27+((NSEG+1)*6*((NSEG-1)*6+1))/2                             MAT12660
      I29=I28+(NSEG+1)*6 - 1                                            MAT12670
      I30=I29+144                                                       MAT12680
      I31=I30+144                                                       MAT12690
      I32=I31+(NSEG+1)*6                                                MAT12700
      I33=I32+((NSEG+1)*6)**2                                           MAT12710
      I34=I33+NSEG*144                                                  MAT12720
      I35=I34+2                                                         MAT12730
      I36=I35+2                                                         MAT12740
      I37=I36+(NSEG+1)*6                                                MAT12750
      I38=I37+(NSEG+1)*6                                                MAT12760
      I39=I38+(NSEG+1)*6                                                MAT12770
      I40=I39+((NSEG+1)*6)**2                                           MAT12780
      I41=I40+NUMP                                                      MAT12790
      I42=I41+NSEG*NUMP                                                 MAT12800
      I43=I42+NUMP                                                      MAT12810
      I44=I43+NUMP                                                      MAT12820
      I45=I44+NSEG*NUMP                                                 MAT12830
      I46=I45+NSEG*NUMP                                                 MAT12840
      I47=I46+NSEG*NUMP                                                 MAT12850
      I48=I47+NSEG*NUMP                                                 MAT12860
      I49=I49+NSEG*NUMP                                                 MAT12870
      I50=I49+NSEG*NUMP                                                 MAT12880
      I51=I50+NSEG*NUMP                                                 MAT12890
      IF (X00.EQ.2) GOTO 205                                            MAT12900
C                                                                       MAT12910
      LMAT= I51 + NUMP - 1                                              MAT12920
      SMAT(29)=LMAT                                                     MAT12930
C                                                                       MAT12940
      RETURN                                                            MAT12950
C                                                                       MAT12960
C-- IOPT-2  ------------- INITIALIZE STIFFNESS TERMS -----------------*MAT12970
C-- IOPT-3  ------------- GET STIFFNESS TERMS -----------------------*MAT12980
  200 CONTINUE                                                          MAT12990
C     ---- STABILITY ELEMENT MATERIAL DATA STORAGE FOR MEMBER ----      MAT13000
C     SE(1- 78)- ERASAT                                                 MAT13010
C     SE(  79)-ELS                                                      MAT13020
C     SE(  80)-IND                                                      MAT13030
C     SE(  81)-ISPE                                                     MAT13040
C     SE(  82)-DELTP2                                                   MAT13050
C     SE(  83)-S2                                                       MAT13060
C     SE(  84)-SP                                                       MAT13070
C     SE(  85)-ISTART                                                   MAT13080
C     SE(  86)-RNEEE                                                    MAT13090
C                                                                       MAT13100
C     SE(  87)-EESE                                                     MAT13110
C     SE(  88)-EPSE                                                     MAT13120
C     SE(  89)-DUCT(1)                                                  MAT13130
C     SE(  90)-DUCT(2)                                                  MAT13140
C     SE(  91)-DUCT(3)                                                  MAT13150
C     SE(  92)-EXCR(1)                                                  MAT13160
C     SE(  93)-EXCR(2)                                                  MAT13170
C     SE(  94)-EXCR(3)                                                  MAT13180
C     SE(  95)-EXCR(4)                                                  MAT13190
C     SE(  96)-EXCR(5)                                                  MAT13200
C     SE(  97)-EXCR(6)                                                  MAT13210
C     SE(  98)-FLP FLAG FOR JUDGING LOCAL BUCKLING                      MAT13220
C                0   :  NO LOCAL BUCKLING OCCURRED                     MAT13230
C                1-4 :  LOCAL BUCKLING OCCURRED                       MAT13240
C     SE(  99)-FLAG FOR JUDGING LOCAL BUCKLING HAS OCCURRED OR NOT      MAT13250
C                0   :  NO LOCAL BUCKLING HAS OCCURRED                 MAT13260
C                1   :  LOCAL BUCKLING HAS OCCURRED                    MAT13270
C                       SET COMPRESSION STIFFNESS=0.001                MAT13280
C-------------------------------------------------                     MAT13290
C                                                                       MAT13300
      NSEG  = SMAT( 1)                                                  MAT13310
      YS    = SMAT( 2)                                                  MAT13320
      EM    = SMAT( 3)                                                  MAT13330
      LIBN  = SMAT( 4)                                                  MAT13340
      HH    = SMAT( 5)                                                  MAT13350
      UU    = SMAT( 6)                                                  MAT13360
      WW    = SMAT( 7)                                                  MAT13370
      ZZ    = SMAT( 8)                                                  MAT13380
      INEB  = SMAT( 9)                                                  MAT13390
      INEH  = SMAT(10)                                                  MAT13400
      ST    = SMAT(11)                                                  MAT13410
      IREV1 = SMAT(12)                                                  MAT13420
      IREV2 = SMAT(13)                                                  MAT13430
      IREV3 = SMAT(14)                                                  MAT13440
      IREV4 = SMAT(15)                                                  MAT13450
      IECOP = SMAT(16)                                                  MAT13460
      NUMP  = SMAT(17)                                                  MAT13470
      SMALL = SMAT(18)                                                  MAT13480
      RATIX0= SMAT(19)                                                  MAT13490
      RATIY0= SMAT(20)                                                  MAT13500
      TOTA  = SMAT(21)                                                  MAT13510
      IAUTO = SMAT(22)                                                  MAT13520
      IMATER= SMAT(23)                                                  MAT13530
      RATIO3= SMAT(24)                                                  MAT13540
      IR    = SMAT(25)                                                  MAT13550
      G     = SMAT(26)                                                  MAT13560
      QRNEE = SMAT(27)                                                  MAT13570
      ISTIF = SMAT(28)                                                  MAT13580
      ISPE=0                                                            MAT13590
      ELS=SE( 79)                                                       MAT13600
```

```
      K00=2                                                             MAT13610
      GOTO 110                                                          MAT13620
C                                                                       MAT13630
  205 IF (IOPT.EQ.2) THEN                                               MAT13640
        ISTART=0                                                        MAT13650
        DO 210 I=1,78                                                   MAT13660
  210   SE(I)=0                                                         MAT13670
        SE(99)=0                                                        MAT13680
        DO 215 I=II,LMAT                                                MAT13690
          SMAT(I)=0.                                                    MAT13700
  215   CONTINUE                                                        MAT13710
      ELSE                                                              MAT13720
        IMD   = SE(  80)                                                MAT13730
        DELTP2= SE(  82)                                                MAT13740
        S2    = SE(  83)                                                MAT13750
        SP    = SE(  84)                                                MAT13760
        ISTART= SE(  85)                                                MAT13770
        RNEEE = SE(  86)                                                MAT13780
      ENDIF                                                             MAT13790
C                                                                       MAT13800
C----CALIBRATE TWO-END DISPLACEMENT INCREMENT BEFORE CALL HYST12         MAT13810
      DO 66 I=1,10,3                                                    MAT13820
        SMAT(I13+I-1)=D(2*I-1)-DL(2*I-1)                                MAT13830
        SMAT(I13+I  )=D(3*I-1)-DL(3*I-1)                                MAT13840
        SMAT(I13+I+1)=D(1*I-1)-DL(1*I-1)                                MAT13850
   66 CONTINUE                                                          MAT13860
C                                                                       MAT13870
      IDOF=(NSEG+1)*6                                                   MAT13880
      JSTOR=IDOF*(IDOF+1)/2                                             MAT13890
      CALL HYST12( IOPT , NSEG , YS    , EM , LIBN ,  HH , UU    ,      MAT13900
     &       JSTOR, IDOF , WW    , ZZ    , INEB , INEH ,  ST , IREV1,  MAT13910
     &       IREV2, IREV3, IREV4, IECOP, ECCX0, ECCY0, NUMP , ELS  ,    MAT13920
     &       SMALL,RATIX0,RATIY0, TOTA , IMD  , ISPE ,DELTP2, S2   ,    MAT13930
     &       SP   ,ISTART, IR    , G    , ISTIF, IAUTO,IMATER,RATIO3,  MAT13940
     &       QRNEE, RNEEE,                                              MAT13950
     & SMAT(I1),SMAT(I2),SMAT(I3),SMAT(I4),SMAT(I5),SMAT(I6),SMAT(I7),  MAT13960
     & SMAT(I8),SMAT( I9),SMAT(I10),SMAT(I11),SMAT(I12),SMAT(I13),      MAT13970
     & SMAT(I14),SMAT(I15),SMAT(I16),SMAT(I17),SMAT(I18),SMAT(I19),     MAT13980
     & SMAT(I20),SMAT(I21),SMAT(I22),SMAT(I23),SMAT(I24),SMAT(I25),     MAT13990
     & SMAT(I26),SMAT(I27),SMAT(I28),SMAT(I29),SMAT(I30),SMAT(I31),     MAT14000
     & SMAT(I32),SMAT(I33),SMAT(I34),SMAT(I35),SMAT(I36),SMAT(I37),     MAT14010
     & SMAT(I38),SMAT(I39),SMAT(I40),SMAT(I41),SMAT(I42),SMAT(I43),     MAT14020
     & SMAT(I44),SMAT(I45),SMAT(I46),SMAT(I47),SMAT(I48),SMAT(I49),     MAT14030
     & SMAT(I50),SMAT(I51),SE(98) )                                     MAT14040
C                                                                       MAT14050
      K=0                                                               MAT14060
      DO 60 I=1,12                                                      MAT14070
      DO 60 J=1,12                                                      MAT14080
        K=K+1                                                           MAT14090
   60   EEASAT(I,J)=SMAT(I29+K-1)                                       MAT14100
C                                                                       MAT14110
C--- -CONVERT STABILITY ELEMENT TWO-END STIFFNESS INTO STANDARD DOF     MAT14120
C     DIRECTION                                                         MAT14130
      DO 68 I=1,12                                                      MAT14140
      DO 68 J=1,12                                                      MAT14150
   68 A(I,J)=0                                                          MAT14160
C                                                                       MAT14170
      DO 69 I=1,4                                                       MAT14180
        A(1+(I-1)*3,3+(I-1)*3)=1                                        MAT14190
        A(2+(I-1)*3,1+(I-1)*3)=1                                        MAT14200
        A(3+(I-1)*3,2+(I-1)*3)=1                                        MAT14210
   69 CONTINUE                                                          MAT14220
C                                                                       MAT14230
      CALL MULTM(3,12,SAT,EEASAT,A)                                     MAT14240
      CALL MULTM(0,12,EEASAT,A,SAT)                                     MAT14250
C                                                                       MAT14260
      IF (IOPT.EQ.3) THEN                                               MAT14270
        IF( SE(99) .EQ. 1 .AND. P(7) .LE. 0 ) THEN                      MAT14280
            XLL=DL(7)-DL(1)                                             MAT14290
            XCC=D(7)-D(1)                                               MAT14300
            XDD=XCC-XLL                                                 MAT14310
            IF( XDD .LT. 0 ) THEN                                       MAT14320
              DO 339 J=1,12                                             MAT14330
                EEASAT(1,J)=0                                           MAT14340
                EEASAT(7,J)=0                                           MAT14350
                EEASAT(J,1)=0                                           MAT14360
                EEASAT(J,7)=0                                           MAT14370
  339         CONTINUE                                                  MAT14380
              EEASAT(1,1)=0.001                                         MAT14390
              EEASAT(7,7)=0.001                                         MAT14400
              EEASAT(1,7)=-0.001                                        MAT14410
              EEASAT(7,1)=-0.001                                        MAT14420
            ENDIF                                                       MAT14430
        ENDIF                                                           MAT14440
        IF ( SE(99) .EQ. 0 .AND. SE(98) .GT. 0 ) SE(99)=1               MAT14450
      ENDIF                                                             MAT14460
      K=0                                                               MAT14470
      DO 80 I=1,12                                                      MAT14480
      DO 80 J=1,12                                                      MAT14490
      K=K+1                                                             MAT14500
   80 SE(K)=EEASAT(I,J)                                                 MAT14510
      SE(  79)=ELS                                                      MAT14520
      SE(  80)=IMD                                                      MAT14530
      SE(  81)=ISPE                                                     MAT14540
      SE(  82)=DELTP2                                                   MAT14550
      SE(  83)=S2                                                       MAT14560
      SE(  84)=SP                                                       MAT14570
      SE(  85)=ISTART                                                   MAT14580
      SE(  86)=RNEEE                                                    MAT14590
C                                                                       MAT14600
      RETURN                                                            MAT14610
C--- GET ENERGIES, DUCTILITY AND EXCURSION FOR BOTH IOPT=3 AND 4 ----   MAT14620
  400 CONTINUE                                                          MAT14630
C                                                                       MAT14640
C --- TRANSFER ENERGIES FOR BOTH IOPT=3 AND 4                           MAT14650
      EESE=0                                                            MAT14660
      EPSE=0                                                            MAT14670
C                                                                       MAT14680
      IF (IOPT.EQ.4) THEN                                               MAT14690
C --- TRANSFER DUCTILITIES AND EXCURSION RATIOS FOR BOTH IOPT=3 AND 4   MAT14700
      DO 410 I=1,3                                                      MAT14710
        DUCT(I)=0                                                       MAT14720
        EXCR(I+3)=0                                                     MAT14730
  410   EXCR(I)=0                                                       MAT14740
C                                                                       MAT14750
      ELSE                                                              MAT14760
C---------------- DAMAGE INDEX ----------------------------------------MAT14770
        DAMAGE=0                                                        MAT14780
      ENDIF                                                             MAT14790
C                                                                       MAT14800
      RETURN                                                            MAT14810
      END                                                               MAT14820
C-----------------------------------------------------------------------MATL0010
      SUBROUTINE MATLIB(IOPT,LSTTYP,HEAD,EESE,EPSE,DUCT,EXCR,           MATL0020
     &       P,PL,D,DL,V,VL,LELEM,LMAT,MAT,MATTYP,SE,IELNO,DAMAGE)      MATL0030
C                                                                       MATL0040
      LOGICAL KSAME,FIRST,HEAD                                          MATL0050
      DIMENSION SE(15),DUCT(3),EXCR(6),P(12),PL(12),D(12),DL(12)        MATL0060
$INCLUDE:'ZCOMN'                                                        MATL0070
C                                                                       MATL0080
C ----CHECK FOR AN INVALID MATERIAL NUMBER . .                          MATL0090
      IF (MAT.LE.0 .OR. MAT.GT.NMAT) THEN                               MATL0100
        WRITE(108,189) MAT                                              MATL0110
  189   FORMAT (5X,20('*'),' ERROR, INVALID MATERIAL PROP #',I5/        MATL0120
     &          5X,'CORRECT INPUT  . . . . . ')                         MATL0130
        STOP                                                            MATL0140
      ENDIF                                                             MATL0150
C                                                                       MATL0160
      IC    = NZ(IZMAT+MAT   )                                          MATL0170
      IF (IOPT.NE.1) THEN                                               MATL0180
            MATTYP = NZ(IC)                                             MATL0190
      ELSE                                                              MATL0200
            NZ(IC) = MATTYP                                             MATL0210
      ENDIF                                                             MATL0220
      IF (LSTTYP.EQ.MATTYP) THEN                                        MATL0230
          FIRST=.FALSE.                                                 MATL0240
      ELSE                                                              MATL0250
          FIRST=.TRUE.                                                  MATL0260
          IF (HEAD)  WRITE(108,305) TITLE(1),TITLE(2)                   MATL0270
  305     FORMAT ('1 STRUCTURE.....: ',A,/                              MATL0280
     &          '  SOLUTION..... ',A,/)                                 MATL0290
      ENDIF                                                             MATL0300
C                                                                       MATL0310
      DAMAGE=0                                                          MATL0320
      IF (MATTYP.EQ.1)  THEN                                            MATL0330
        CALL MAT01(IOPT,FIRST,MAT,SE,P(1),PL(1),D(1),DL(1),LELEM,       MATL0340
     &          LMAT,Z(IC+1),EESE,EPSE,DUCT,EXCR)                       MATL0350
      ELSE IF (MATTYP.EQ.2)  THEN                                       MATL0360
        CALL MAT02(IOPT,IELNO,FIRST,MAT,SE,P,PL,D,DL,V,VL,              MATL0370
     &          LELEM,LMAT,Z(IC+1),EESE,EPSE,DUCT,EXCR,DAMAGE)          MATL0380
      ELSE IF (MATTYP.EQ.3)  THEN                                       MATL0390
        CALL MAT03(IOPT,IELNO,FIRST,MAT,SE,P,PL,D,DL,V,LELEM,           MATL0400
     &          LMAT,Z(IC+1),EESE,EPSE,DUCT,EXCR,DAMAGE,NSOLN)          MATL0410
      ELSE IF (MATTYP.EQ.4)  THEN                                       MATL0420
        CALL MAT04(IOPT,IELNO,FIRST,MAT,SE,P,PL,D,DL,V,VL,LELEM,        MATL0430
     &          LMAT,Z(IC+1),EESE,EPSE,DUCT,EXCR,DAMAGE,NSOLN)          MATL0440
      ELSE IF (MATTYP.EQ.6)  THEN                                       MATL0450
        CALL MAT06(IOPT,IELNO,FIRST,MAT,SE,P,PL,D,DL,V,VL,              MATL0460
     &          LELEM,LMAT,Z(IC+1),EESE,EPSE,DUCT,EXCR,DAMAGE)          MATL0470
      ELSE IF (MATTYP.EQ.7)  THEN                                       MATL0480
        CALL MAT07(IOPT,FIRST,MAT,SE,P,PL,D,DL,V,VL,LELEM,              MATL0490
     &          LMAT,Z(IC+1),EESE,EPSE,DUCT,EXCR,DAMAGE)                MATL0500
      ELSE IF (MATTYP.EQ.8)  THEN                                       MATL0510
        CALL MAT08(IOPT,FIRST,MAT,SE,P,PL,D,DL,LELEM,                   MATL0520
     &          LMAT,Z(IC+1),EESE,EPSE,DUCT,EXCR,DAMAGE)                MATL0530
      ELSE IF (MATTYP.EQ.9)  THEN                                       MATL0540
        CALL MAT09(IOPT,FIRST,MAT,SE,P,PL,D,DL,LELEM,LMAT,              MATL0550
     &          Z(IC+1),EESE,EPSE,DUCT,EXCR,DAMAGE)                     MATL0560
      ELSE IF (MATTYP.EQ.10)  THEN                                      MATL0570
        CALL MAT10(IOPT,FIRST,MAT,SE,P,PL,D,DL,LELEM,LMAT,              MATL0580
     &          Z(IC+1),EESE,EPSE,DUCT,DAMAGE)                          MATL0590
      ELSE IF (MATTYP.EQ.11)  THEN                                      MATL0600
        CALL MAT11(IOPT,FIRST,MAT,SE,P,PL,D,DL,LELEM,LMAT,              MATL0610
     &          Z(IC+1),EESE,EPSE,DUCT,EXCR,DAMAGE)                     MATL0620
      ELSE IF (MATTYP.EQ.12)  THEN                                      MATL0630
        LMAT=Z(IC+29)                                                   MATL0640
        CALL MAT12(IOPT,FIRST,MAT,SE,P,D,DL,LELEM,LMAT,NZ(IC+1),        MATL0650
     &          Z(IC+1),EESE,EPSE,DUCT,EXCR,DAMAGE)                     MATL0660
        LMAT=Z(IC+29)                                                   MATL0670
      ELSE IF (MATTYP.EQ.13)  THEN                                      MATL0680
        CALL MAT13(IOPT,FIRST,MAT,SE,P,PL,D,DL,LELEM,LMAT,              MATL0690
     &          Z(IC+1),EESE,EPSE,DUCT,EXCR,DAMAGE)                     MATL0700
      ELSE IF (MATTYP.EQ.14)  THEN                                      MATL0710
        CALL MAT14(IOPT,FIRST,MAT,SE,P,PL,D,DL,LELEM,                   MATL0720
     &          LMAT,Z(IC+1),EESE,EPSE,DUCT,EXCR,DAMAGE)                MATL0730
      ELSE                                                              MATL0740
        WRITE(108,10) IELNO                                             MATL0750
   10   FORMAT(' ELEMENT ',I6,' IS NOT AVAILABLE' )                     MATL0760
      ENDIF                                                             MATL0770
      LSTTYP=MATTYP                                                     MATL0780
C                                                                       MATL0790
      RETURN                                                            MATL0800
      END                                                               MATL0810
C                                                                       DIRE0010
C - DIRECTION OF LOADING .. ..... ..... ..... ..... ..... ..... ..... DIRE0020
C                                                                       DIRE0030
      SUBROUTINE DIRECT(DIR,DIRL,P,PL,VEL)                              DIRE0040
      INTEGER DIR,DIRL                                                  DIRE0050
C                                                                       DIRE0060
C     P       - CURRENT LOAD                                            DIRE0070
C     PL      - LAST LOAD                                               DIRE0080
C     D       - CURRENT DISPLACEMENT                                    DIRE0090
C     DIR     - CURRENT DIRECTION                                       DIRE0100
C               DIR=1, POSITIVE LOADING                                 DIRE0110
C               DIR=2, POSITIVE UNLOADING                               DIRE0120
C               DIR=3, NEGATIVE LOADING                                 DIRE0130
C               DIR=4, NEGATIVE UNLOADING                               DIRE0140
C     DIRL    - LAST DIRECTION                                          DIRE0150
C     VEL     - PRODUCT OF LAST AND CURRENT VELOCITIES                  DIRE0160
      IF (P.GT.0) THEN                                                  DIRE0170
        IF (P.GT.PL .AND. PL.GT.0) THEN                                 DIRE0180
            DIRL=1                                                      DIRE0190
        ELSE IF (P.GT.PL  AND. PL LT.0) THEN                            DIRE0200
            DIRL=4                                                      DIRE0210
        ELSE IF (P.LT.PL) THEN                                          DIRE0220
            DIRL=2                                                      DIRE0230
        ELSE                                                            DIRE0240
            DIRL=DIR                                                    DIRE0250
        ENDIF                                                           DIRE0260
      ELSE IF (P.LT.0) THEN                                             DIRE0270
        IF (P.LT.PL .AND. PL LT.0) THEN                                 DIRE0280
            DIRL=3                                                      DIRE0290
        ELSE IF (P.LT PL .AND. PL GT.0) THEN                            DIRE0300
            DIRL=2                                                      DIRE0310
        ELSE IF (P.GT.PL) THEN                                          DIRE0320
            DIRL=4                                                      DIRE0330
        ELSE                                                            DIRE0340
            DIRL=DIR                                                    DIRE0350
        ENDIF                                                           DIRE0360
      ELSE                                                              DIRE0370
        IF (PL.LT.0) THEN                                               DIRE0380
            DIRL=4                                                      DIRE0390
        ELSE IF (PL.GT.0) THEN                                          DIRE0400
            DIRL=2                                                      DIRE0410
        ELSE                                                            DIRE0420
            DIRL=DIR                                                    DIRE0430
        ENDIF                                                           DIRE0440
      ENDIF                                                             DIRE0450
      IF (DIRL.EQ.1 .OR.DIRL.EQ.0) THEN                                 DIRE0460
        IF (VEL.GT.0) THEN                                              DIRE0470
            DIR=1                                                       DIRE0480
        ELSE                                                            DIRE0490
            DIR=2                                                       DIRE0500
        ENDIF                                                           DIRE0510
      ELSE IF (DIRL.EQ 2) THEN                                          DIRE0520
        IF (VEL.GT.0) THEN                                              DIRE0530
            IF (P.GT.0) THEN                                            DIRE0540
                DIR=2                                                   DIRE0550
            ELSE                                                        DIRE0560
                DIR=3                                                   DIRE0570
            ENDIF                                                       DIRE0580
        ELSE                                                            DIRE0590
            DIR=1                                                       DIRE0600
        ENDIF                                                           DIRE0610
      ELSE IF (DIRL.EQ.3) THEN                                          DIRE0620
        IF (VEL.GT 0) THEN                                              DIRE0630
            DIR=3                                                       DIRE0640
        ELSE                                                            DIRE0650
            DIR=4                                                       DIRE0660
        ENDIF                                                           DIRE0670
      ELSE IF (DIRL EQ 4) THEN                                          DIRE0680
        IF (VEL GT 0) THEN                                              DIRE0690
            IF (P.LT.0) THEN                                            DIRE0700
                DIR=4                                                   DIRE0710
            ELSE                                                        DIRE0720
                DIR=1                                                   DIRE0730
            ENDIF                                                       DIRE0740
        ELSE                                                            DIRE0750
            DIR=3                                                       DIRE0760
        ENDIF                                                           DIRE0770
      ELSE                                                              DIRE0780
        WRITE(108,5)                                                    DIRE0790
    5   FORMAT(5X,'ER','ROR IN SUBROUTINE DIRECT, DIRL NE. 1,2,3,OR 4') DIRE0800
        WRITE(108,10) DIR,DIRL,P,PL,VEL                                 DIRE0810
      ENDIF                                                             DIRE0820
      IF (DIR.EQ.0) DIR=1                                               DIRE0830
   10 FORMAT (5X,'*** IN DIRECT ',1X,'  DIR=',I4,'  DIRL=',I4,          DIRE0840
     &          '  P=',F8.3,'  PL=',F8.3,'  VEL=',F10.3)                DIRE0850
      RETURN                                                            DIRE0860
      END                                                               DIRE0870
C                                                                       FUNC0010
C - XMIN - CALCULATES STIFFNESS BASED ON DX AND DY . ... . . .          FUNC0020
```

```
C                                                                       FUNC0030
      REAL FUNCTION XMIN(DD,DP,SI)                                      FUNC0040
C     XMIN    =                                                         FUNC0050
C     DD      - CHANGE IN DISPLACMENT                                   FUNC0060
C     DP      - CHANGE IN LOAD                                          FUNC0070
C     SI      - MAXIMUN AND DEFULT STIFFNESS                            FUNC0080
            IF ((DD*DP) .GT. 0.00)  THEN                                FUNC0090
               XMIN=MIN((DP/DD),SI)                                     FUNC0100
            ELSE                                                        FUNC0110
               XMIN=SI                                                  FUNC0120
            ENDIF                                                       FUNC0130
      RETURN                                                            FUNC0140
      END                                                               FUNC0150
C                                                                       INTS0010
C  - INTERSECTION OF TWO LINES IN POINT SLOPE FORM ..... .  ... ....    INTS0020
C                                                                       INTS0030
      SUBROUTINE INTSCT(X,Y,X1,Y1,S1,X2,Y2,S2)                          INTS0040
C     X       - X COORDINATE OF INTERSECTION                            INTS0050
C     Y       - Y COORDINATE OF INTERSECTION                            INTS0060
C     X1      - X COORDINATE OF LINE 1                                  INTS0070
C     Y1      - Y COORDINATE OF LINE 1                                  INTS0080
C     S1      - SLOPE OF LINE 1                                         INTS0090
C     X2      - X COORDINATE OF LINE 2                                  INTS0100
C     Y2      - Y COORDINATE OF LINE 2                                  INTS0110
C     S2      - SLOPE OF LINE 2                                         INTS0120
      IF (S1.NE.S2) THEN                                                INTS0130
            B1= Y1 - S1*X1                                              INTS0140
            B2= Y2 - S2*X2                                              INTS0150
            X=(B1-B2)/(S2-S1)                                           INTS0160
            Y=(B1*S2-B2*S1)/(S2-S1)                                     INTS0170
      ELSE                                                              INTS0180
        WRITE(108,10) X1,Y1,S1,X2,Y2,S2                                 INTS0190
   10   FORMAT(/' ******* ERROR IN INTSCT.... S1=S2, NO SOLN,  THUS',   INTS0200
     &        ' SET X=Y=0.',/'   X1=',1P,G15.5,'  Y1=',G15.5,'  S1=',  INTS0210
     &   G15.5/'   X2=',1P,G15 5,'  Y2=',G15.5,'  S2=',G15.5/)          INTS0220
        STOP                                                            INTS0230
      ENDIF                                                             INTS0240
      RETURN                                                            INTS0250
      END                                                               INTS0260
C                                                                       IDRE0010
C ------------------------------ ... ------------------------          IDRE0020
C                                                                       IDRE0030
      SUBROUTINE IDRECT (IDR,IDRV,IDRVO,PI,PS,VI,VS)                    IDRE0040
C                                                                       IDRE0050
      LOGICAL A,NA,B,NB,C,NC,D,ND,E                                     IDRE0060
C                                                                       IDRE0070
  1   A = ABS(PI).GE.ABS(PS)                                            IDRE0080
      B = PI*PS .GE.0                                                   IDRE0090
      C = PI.EQ.0                                                       IDRE0100
      D = VS*VI.GE.0                                                    IDRE0110
      NA= .NOT. A                                                       IDRE0120
      NB= .NOT. B                                                       IDRE0130
      NC= .NOT. C                                                       IDRE0140
      ND= .NOT. D                                                       IDRE0150
C                                                                       IDRE0160
      IF ( A .AND.  B .AND. NC .AND.  D ) IDR=1                         IDRE0170
      IF (          NB .AND. NC .AND. ND ) IDR=1                        IDRE0180
      IF (NA .AND.  B .AND. NC .AND. ND ) IDR=1                         IDRE0190
      IF (            C .AND.  D ) IDR=0                              IDRE0200
      IF (          NB .AND. NC .AND.  D ) IDR=0                      IDRE0210
      IF ( A .AND.  B .AND. NC .AND. ND ) IDR=-1                        IDRE0220
      IF (            C .AND. ND ) IDR=-1                             IDRE0230
      IF (NA .AND   B .AND. NC .AND.  D ) IDR=-1                        IDRE0240
          E = PI .GE  PS                                                IDRE0250
          IDRV=1                                                        IDRE0260
          IDRVO=2                                                       IDRE0270
          IF (E) IDRV=2                                                 IDRE0280
          IF (E) IDRVO=1                                                IDRE0290
      RETURN                                                            IDRE0300
      END                                                               IDRE0310
C                                                                       DNEI0010
C-----------------------------------------------------------------------DNEI0020
C  DUCTILITIES AND EXCURSION RATIOS                                     DNEI0030
C                                                                       DNEI0040
      SUBROUTINE DNE(IOPT,U,E,DY,D,ESE,PSE,PSEOLD,CSE)                  DNEI0050
      DIMENSION U(3),E(6),UC(3)                                         DNEI0060
C                                                                       DNEI0070
C------- VARIABLES ------------------------------------------------------DNEI0080
C     U(I)    - DUCTILITY   (I)                                         DNEI0090
C     E(I)    - EXCURSION RATIO (I)                                     DNEI0100
C     E(1+3)  - DUCTILITY (I) FOR THE CURRENT HALF CYCLE                DNEI0110
C     DY      - YEILD DISPLACMENT                                       DNEI0120
C     D       - CURRENT DISPL                                           DNEI0130
C     TSE     - TOTAL STRAIN ENERGY                                     DNEI0140
C     PSE     - PLASTIC STRAIN ENERGY                                   DNEI0150
C     ESE     - ELASTIC STRAIN ENERGY                                   DNEI0160
C     CSE     - CONSTANT STRAIN ENERGY AT YEILD .  .                    DNEI0170
C     UC      - CURRENT DUCTILITY                                       DNEI0180
C-------------------------------- -- ... --  ------------------------ ... DNEI0190
C                                                                       DNEI0200
      IF (IOPT.EQ.1) THEN                                               DNEI0210
C             ..... CALCULATE CURRENT DUCTILITIES  ..  ..               DNEI0220
            PSEO=PSE-PSEOLD                                             DNEI0230
            IF (DY.NE.0)  UC(1)=D/DY                                    DNEI0240
            IF (ESE.NE.0) UC(2)=(PSEO/ESE +1 )                          DNEI0250
            IF (CSE.NE.0) UC(3)=(PSEO/CSE +1 )                          DNEI0260
            U(1)=MAX( U(1),UC(1) )                                      DNEI0270
            U(2)=MAX( U(2),UC(2) )                                      DNEI0280
            U(3)=MAX( U(3),UC(3) )                                      DNEI0290
            E(4)=MAX( E(4),UC(1) )                                      DNEI0300
            E(5)=MAX( E(5),UC(2) )                                      DNEI0310
            E(6)=MAX( E(6),UC(3) )                                      DNEI0320
      ELSE                                                              DNEI0330
C             ..... CALCULATE EXCURSION RATIO'S ...   .                 DNEI0340
            DO 20 I=1,3                                                 DNEI0350
              J=I+3                                                     DNEI0360
              IF (E(J) GT 1) E(I) = E(I) + E(J)  1                      DNEI0370
   20         E(J)=0                                                   DNEI0380
      ENDIF                                                             DNEI0390
C                                                                       DNEI0400
      RETURN                                                            DNEI0410
      END                                                               DNEI0420
C                                                                       STRE0010
C-----------------------------------------------------------------------STRE0020
C                                                                       STRE0030
      SUBROUTINE STRENG (ESE,PSE,PSEOLD,P,PL,D,DL,SE)                   STRE0040
      DIMENSION ESE(3)                                                  STRE0050
C-- ------------------------------------------------------------------- STRE0060
C VARIABLES:                                                            STRE0070
C------------------- ------------------  ------------------------------ STRE0080
C TSE     - TOTAL ELEMENT STRAIN ENERGY                                 STRE0090
C ESE(1)  - ELASTIC ELEMENT STRAIN ENERGY                               STRE0100
C ESE(2)  - MAXIMUM ELASTIC ELEMENT STRAIN ENERGY, POSITIVE LOADING     STRE0110
C ESE(3)  - MAXIMUM ELASTIC ELEMENT STRAIN ENERGY, NEGATIVE LOADING     STRE0120
C PSE     - PLASTIC ELEMENT STRAIN ENERGY                               STRE0130
C PSEOLD  - PLASTIC ELEMENT STRAIN ENERGY AT LAST ZERO CROSSING         STRE0140
C P       - CURRENT LOAD                                                STRE0150
C PL      - PREVIOUS LOAD                                               STRE0160
C D       - CURRENT DISPLACEMENT                                        STRE0170
C DL      - PREVIOUS DISPLACEMENT                                       STRE0180
C SE      - UNLOADING STIFFNESS                                         STRE0190
C--------- ------------------------------------------------------------ STRE0200
C                                                                       STRE0210
      PSEL  = PSE                                                       STRE0220
C             PREVENT UNDERFLOW FOR VERY SMALL PSUM OR DDIF             STRE0230
            PSUM=P+PL                                                   STRE0240
            DDIF=D-DL                                                   STRE0250
            TOL=1 00E-25                                                STRE0260
            IF (ABS(PSUM).LE.TOL .OR  ABS(DDIF) LE TOL) THEN            STRE0270
               DTSE=0                                                   STRE0280
            ELSE                                                        STRE0290
               DTSE  = PSUM*DDIF/2 0D                                   STRE0300
            ENDIF                                                       STRE0310
      TSE   = ESE(1)+ PSE + DTSE                                        STRE0320
C             PREVENT UNDERFLOW FOR VERY SMALL P, OR  SE=0              STRE0330
            IF (ABS(P).LE.TOL) THEN                                     STRE0340
               ESE(1)=0.                                                STRE0350
            ELSE IF (SE.EQ.0) THEN                                      STRE0360
               ESE(1)=TSE                                               STRE0370
            ELSE                                                        STRE0380
               ESE(1)= P*P/(2*SE)                                       STRE0390
            ENDIF                                                       STRE0400
      ESE(1)= MIN(TSE,ESE(1))                                           STRE0410
      PSE   = TSE-ESE(1)                                                STRE0420
C                                                                       STRE0430
C.....  .. CALC PLACTIC STRAIN ENERGY AT ZERO CROSSING ..               STRE0440
      IF (P*PL.LE.0) THEN                                               STRE0450
         DP=PL-P                                                        STRE0460
         IF (DP.NE.0) THEN                                              STRE0470
           IF( (PSEL - PSE) .LT. TOL) THEN                              STRE0480
             PSEOLD=PSE                                                 STRE0490
           ELSE                                                         STRE0500
             PSEOLD=PSE+ (PSEL-PSE)*P/(PL-P)                            STRE0510
           ENDIF                                                        STRE0520
         ELSE                                                           STRE0530
             PSEOLD=PSE                                                 STRE0540
         ENDIF                                                          STRE0550
      ENDIF                                                             STRE0560
C                                                                       STRE0570
C........ DETERMINE MAXIMUM ELASTIC STRAIN ENERGY/SIDE...               STRE0580
      IF (P.GE.0) THEN                                                  STRE0590
         ESE(2)=MAX(ESE(2),ESE(1))                                      STRE0600
      ELSE                                                              STRE0610
         ESE(3)=MAX(ESE(3),ESE(1))                                      STRE0620
      ENDIF                                                             STRE0630
C                                                                       STRE0640
      RETURN                                                            STRE0650
      END                                                               STRE0660
C                                                                       HYST0010
C     SU BROUTINE AXLMOD - AXIAL HYSTERESIS MODEL                       HYST0020
C     REFERANCE: ANALYSIS OF THE FULL SCALE SEVEN                       HYST0030
C                STORY R/C TEST STRUCTURE.                              HYST0040
      SUBROUTINE HYST02(P,D,PL,DL,V,VL,S,HRLN,XC,KT1,KT2,PY,ALPHA,BETA, HYST0050
     &          DYT,DYC,CSE,PMAX,DMAX,FX,DX,PP,DP,KR,KS,KT,NRL,A,       HYST0060
     &          SEQ,ESE,PSE,PSEOLD,UPOS,EXCR,IDR)                       HYST0070
C                                                                       HYST0080
      REAL*4 KC,KT1,KT2,KR,KS,KT,NRL                                    HYST0090
      INTEGER RULEI                                                     HYST0100
      DIMENSION UPOS(3),EXCR(6),ESE(3)                                  HYST0110
      LOGICAL PGTPL,PLTPL                                               HYST0120
C                                                                       HYST0130
C     P       - CURRENT LOAD, POSITIVE IS TENSION                       HYST0140
C     D       - CURRENT DISPLACMENT, POSITIVE IS ELONGATION             HYST0150
C     S       - STIFFNESS                                               HYST0160
C     XC      - VIRGIN COMPRESSION STIFFNESS, EC                        HYST0170
C     KT1     - PRE-YEILD VIRGIN TENSILE STIFFNESS, 0.9*KC              HYST0180
C     KT2     - POST-YEILD VIRGIN TENSILE STIFFNESS, 0 001*KC           HYST0190
C     PY      - TENSILE YEILD POINT, PHO*FY                             HYST0200
C     PMAX    - MAXIMUN TENSILE FORCE BEFORE UNLOADING                  HYST0210
C     DMAX    - MAXIMUN TENSILE DISPLACMENT BEFORE UNLOADING            HYST0220
C     FX      - FORCE AT STIFFNESS CHANGE POINT ON THE UNLOADING CURVE  HYST0230
C     DX      - DISPL AT STIFFNESS CHANGE POINT ON THE UNLOADING CURVE  HYST0240
C     PI,DI   - INTERSECTION OF TWO LINES                               HYST0250
C     DYT     - TENSILE YEILD DISPLACMENT = PY/KT1                      HYST0260
C     DYC     - COMPRESSION YEILD DISPLACMENT = PY/KC                   HYST0270
C     PP      - FORCE AT COMPRESSION STIFFNESS CHANGE POINT             HYST0280
C     DP      - COMPRESSION  AT COMPRESSION STIFFNESS CHANGE POINT      HYST0290
C     KR      - UNLOADING STIFFNESS = KC/(DMAX/DYT)**(1.20)             HYST0300
C     KS      - LOADING STIFFNESS, LINE BETWEEN (FX,DX) AND (PP,DP)     HYST0310
C     KT      - LOADING STIFFNESS, LINE BETWEEN (PP,DP) AND (-2PY,-2DY) HYST0320
C                                                                       HYST0330
C....... SET INITIAL STIFFNESS AND RETURN                               HYST0340
 8000 IF (NRL.EQ.0) THEN                                                HYST0350
            IF (P.LE.0) THEN                                            HYST0360
                 NRL=1                                                  HYST0370
                 HRLN=1.1                                               HYST0380
                 S=XC                                                   HYST0390
                 SEQ=XC                                                 HYST0400
                 A=P                                                    HYST0410
            ELSE                                                        HYST0420
                 NRL=2                                                  HYST0430
                 HRLN=2.1                                               HYST0440
                 S=KT1                                                  HYST0450
                 SEQ=KC                                                 HYST0460
                 A=P                                                    HYST0470
            ENDIF                                                       HYST0480
            RETURN                                                      HYST0490
      ENDIF                                                             HYST0500
C                                                                       HYST0510
C.... ... RETURN IF NO STEP HAS BEEN TAKEN                              HYST0520
      IF (P.EQ PL .OR. D.EQ.DL ) RETURN                                 HYST0530
C                                                                       HYST0540
C....... SET LOADING FLAGS                                              HYST0550
      PGTPL=P.GT.PL                                                     HYST0560
      PLTPL=P.LT.PL                                                     HYST0570
C                                                                       HYST0580
C. ....  CALC EXCURSION RATIO AT EVERY ZERO CROSSING...                 HYST0590
      IF (P*PL.LE.0 .AND. PL.GT.0)                                      HYST0600
     &    CALL DNE(0,UPOS,EXCR,DYT,D,ESE,PSE,PSEOLD,CSE)                HYST0610
      IF (P*PL.LE.0 .AND. PL.LE.0) THEN                                 HYST0620
          EXCR(4)=0                                                     HYST0630
          EXCR(5)=0                                                     HYST0640
          EXCR(6)=0                                                     HYST0650
      ENDIF                                                             HYST0660
C                                                                       HYST0670
C..... CALC DUCTILITY IF LOAD GT PAST LOAD                              HYST0680
C                                                                       HYST0690
C     DUCTILITY FOR POSITIVE LOADS ONLY                                 HYST0700
      IF (P.GT 0 .AND. PGTPL)                                           HYST0710
     &    CALL DNE(1,UPOS,EXCR,DYT,D ,ESE(2),PSE,PSEOLD,CSE)            HYST0720
C                                                                       HYST0730
 999  CALL IDRECT(IDR,IDRV,IDRVO,P,PL,V,VL)                             HYST0740
      IDR=IDR                                                           HYST0750
C                                                                       HYST0760
   1  IF (NRL.EQ.0) NRL=1                                               HYST0770
C                                                                       HYST0780
      RULEI=NRL                                                         HYST0790
C                                                                       HYST0800
      GO TO (100,200,300,400,500,600,700,800,900,                       HYST0810
     &                   1000,1100,1200,1300) RULEI                     HYST0820
C                                                                       HYST0830
C RULE 1   LOADING AND UNLOADING ON THE COMPRESSION BRANCH BEFORE YEILD HYST0840
C                                                                       HYST0850
 100  SEQ=KC                                                            HYST0860
      IF (P GT 0) GO TO 200                                             HYST0870
      S=KC                                                              HYST0880
      NRL=1                                                             HYST0890
      HRLN=1.1                                                          HYST0900
      IF (PLTPL) A=MIN(10 *P,.2 *PY)                                    HYST0910
      IF (PGTPL) A=0.                                                   HYST0920
      RETURN                                                            HYST0930
C                                                                       HYST0940
C RULE 2 - TENSION LOADING BEFORE YEILD                                 HYST0950
C                                                                       HYST0960
 200  IF (IDR) 230,230,210                                              HYST0970
 210  IF (P GE.PY) GO TO 600                                            HYST0980
      S=KT1                                                             HYST0990
      NRL=2                                                             HYST1000
      HRLN=2.1                                                          HYST1010
      A=PY                                                              HYST1020
      RETURN                                                            HYST1030
 230  IF (PL.LE.0) GO TO 210                                            HYST1040
      FMAX=AMAX1(P,FMAX)                                                HYST1050
      DMAX=AMAX1(D,DMAX)                                                HYST1060
      FX  =FMAX-PY                                                      HYST1070
      DX  =DMAX-PY/KC                                                   HYST1080
      S=KC                                                              HYST1090
      NRL=3                                                             HYST1100
      HRLN=2.2                                                          HYST1110
      A=FX                                                              HYST1120
      RETURN                                                            HYST1130
C                                                                       HYST1140
```

```
C RULE 3 - UNLOADING ON TENSION BEFORE YEILD                          HYST1150
C                                                                     HYST1160
 300  IF (PGTPL) GO TO 500                                           HYST1170
      IF (P.GT.0) GO TO 330                                          HYST1180
 310  IF (KC.NE.KT1) THEN                                            HYST1190
         CALL INTSCT(D1,PI,D,P,KC,DX,FX,KT1)                         HYST1200
      ELSE                                                           HYST1210
         PI=P                                                        HYST1220
         DI=D                                                        HYST1230
      ENDIF                                                          HYST1240
      IF (P.LE.PI) GO TO 320                                         HYST1250
      S=KC                                                           HYST1260
      NRL=3                                                          HYST1270
      HRLN=3.11                                                      HYST1280
      A=PI                                                           HYST1290
      RETURN                                                         HYST1300
 320  S=KT1                                                          HYST1310
      NRL=4                                                          HYST1320
      HRLN=3.12                                                      HYST1330
      A=PY                                                           HYST1340
      RETURN                                                         HYST1350
 330  IF (P.GE.FMAX) GO TO 200                                       HYST1360
      S=KC                                                           HYST1370
      NRL=3                                                          HYST1380
      HRLN=3.2                                                       HYST1390
      A=FX                                                           HYST1400
      RETURN                                                         HYST1410
C                                                                    HYST1420
C RULE 4 - LOADING TOWARDS Y' FROM FX.                               HYST1430
C                                                                    HYST1440
 400  IF (IDR) 500,500,410                                           HYST1450
 410  IF (ABS(P).GE.PY) GO TO 100                                    HYST1460
      S=KT1                                                          HYST1470
      NRL=4                                                          HYST1480
      HRLN=4.1                                                       HYST1490
      A=-PY                                                          HYST1500
      RETURN                                                         HYST1510
C                                                                    HYST1520
C RULE 5 - LOADING AND UNLOADING BETWEEN BRANCHES O-Y AND FX-Y'      HYST1530
C                                                                    HYST1540
 500  IF (XC.NE.KT1) THEN                                            HYST1550
         CALL INTSCT(DITENS,PITENS,D,P,KC,DYT,PY,KT1)                HYST1560
         CALL INTSCT(DICOMP,PICOMP,D,P,KC,DX ,FX,KT1)                HYST1570
      ELSE                                                           HYST1580
         PITENS=P                                                    HYST1590
         DITENS=D                                                    HYST1600
         PICOMP=P                                                    HYST1610
         DICOMP=D                                                    HYST1620
      ENDIF                                                          HYST1630
      IF (P.LE.PICOMP) GO TO 330                                     HYST1640
      IF (P.GE.PITENS) GO TO 210                                     HYST1650
      S=KC                                                           HYST1660
      NRL=5                                                          HYST1670
      HRLN=5.1                                                       HYST1680
      IF (PGTPL ) A=PITENS                                           HYST1690
      IF (PLTPL ) A=PICOMP                                           HYST1700
      RETURN                                                         HYST1710
C                                                                    HYST1720
C RULE 6 - LOADING AFTER TENSION YEILD                               HYST1730
C                                                                    HYST1740
 600  IF (IDR) 630,630,610                                           HYST1750
 610  S=KT2                                                          HYST1760
      NRL=6                                                          HYST1770
      HRLN=6.1                                                       HYST1780
      A=5*P                                                          HYST1790
C... ..... CALC UNLOADING CURVE FOR ENERGY INFO                      HYST1800
          FMAX=AMAX1(P,FMAX)                                         HYST1810
          DMAX=AMAX1(D,DMAX)                                         HYST1820
          KR=XC*(DMAX/DYT)**(-1. * ALPHA)                            HYST1830
          FX  =FMAX-PY                                               HYST1840
          DX  =DMAX-PY/KR                                            HYST1850
          KS  =(PY+FX)/(DX-DYC)                                      HYST1860
C........ FORCE KR GE XS, TO PREVENT UNSTABLE LOOPS...               HYST1870
          IF (KS GT.KR)  KR=(FMAX+PY)/(DMAX-DYC)                     HYST1880
          SEQ=KR                                                     HYST1890
      RETURN                                                         HYST1900
 630  FMAX=AMAX1(P,FMAX)                                             HYST1910
      DMAX=AMAX1(D,DMAX)                                             HYST1920
      KR=XC*(DMAX/DYT)**(-1. * ALPHA)                                HYST1930
      FX  =FMAX-PY                                                   HYST1940
      DX  =DMAX-PY/KR                                                HYST1950
      KS  =(PY-FX)/(DX-DYC)                                          HYST1960
C........ FORCE KR GE XS, TO PREVENT UNSTABLE LOOPS...               HYST1970
      IF (KS GT.KR) THEN                                             HYST1980
        KR=(FMAX+PY)/(DMAX-DYC)                                      HYST1990
        KS=KR                                                        HYST2000
        DX=DMAX-PY/KR                                                HYST2010
      ENDIF                                                          HYST2020
      DP  =DYC + BETA*(DX-DYC)                                       HYST2030
      FP  =FX - (DX-DP)*KS                                           HYST2040
      KT  =(2*PY+FP)/(DP-2*DYC)                                      HYST2050
      DQ  =DYC + PY/KR                                               HYST2060
      S=KR                                                           HYST2070
      NRL=7                                                          HYST2080
      HRLN=6.2                                                       HYST2090
      A=FX                                                           HYST2100
      RETURN                                                         HYST2110
C                                                                    HYST2120
C RULE 7 - UNLOADING AFTER YEILD                                     HYST2130
C                                                                    HYST2140
 700  SEQ=KR                                                         HYST2150
      IF (IDR) 730,730,710                                           HYST2160
 710  IF (P GE.FMAX) GO TO 600                                       HYST2170
      S=KR                                                           HYST2180
      NRL=7                                                          HYST2190
      HRLN=7.1                                                       HYST2200
      A=FMAX                                                         HYST2210
      RETURN                                                         HYST2220
 730  IF (P.LE.FX) GO TO 800                                         HYST2230
      S=KR                                                           HYST2240
      NRL=7                                                          HYST2250
      HRLN=7.2                                                       HYST2260
      A=FX                                                           HYST2270
      RETURN                                                         HYST2280
C                                                                    HYST2290
C RULE 8 - ON BRANCH BETWEEN DX & DP                                 HYST2300
C                                                                    HYST2310
 800  IF (PGTPL) THEN                                                HYST2320
        IF (KR.EQ.XS) GO TO 1200                                     HYST2330
        GO TO 1300                                                   HYST2340
      ENDIF                                                          HYST2350
      IF (P.LE.FP) GO TO 900                                         HYST2360
      S=XS                                                           HYST2370
      NRL=8                                                          HYST2380
      HRLN=8.1                                                       HYST2390
      A=FP                                                           HYST2400
      RETURN                                                         HYST2410
C                                                                    HYST2420
C RULE 9 - ON BRANCH FP-Y"                                           HYST2430
C                                                                    HYST2440
 900  IF (PGTPL) GO TO 1300                                          HYST2450
      IF (P.LE.(-2*PY) ) GO TO 1000                                  HYST2460
      S=KT                                                           HYST2470
      NRL=9                                                          HYST2480
      HRLN=9.1                                                       HYST2490
      A=-2*PY                                                        HYST2500
      RETURN                                                         HYST2510
C                                                                    HYST2520
C RULE 10 - ON THE COMPRESSION CURVE AFTER YELDING                   HYST2530
C                                                                    HYST2540
 1000 SEQ=  1  / ( 1./KC + (4./KR - 4 /KC)*(PY/P)**2 )              HYST2550
C                                                                    HYST2560
      IF (IDR) 1030,1030,1010                                        HYST2570
 1010 S=KC                                                           HYST2580
      NRL=10                                                         HYST2590
      HRLN=10.1                                                      HYST2600
      A=10*P                                                         HYST2610
      RETURN                                                         HYST2620
 1030 IF (ABS(P).LE.PY) GO TO 1100                                   HYST2630
      S=KC                                                           HYST2640
      NRL=10                                                         HYST2650
      HRLN=10.2                                                      HYST2660
      A=-PY                                                          HYST2670
      RETURN                                                         HYST2680
C                                                                    HYST2690
C RULE 11 - UNLOADING BETWEEN Y' & Q                                 HYST2700
C                                                                    HYST2710
 1100 SEQ=KR                                                         HYST2720
      IF (ABS(P).GT.PY ) GO TO 1000                                  HYST2730
      IF (    P .GE. 0.) GO TO 1200                                  HYST2740
      S=KR                                                           HYST2750
      NRL=11                                                         HYST2760
      HRLN=11.1                                                      HYST2770
      IF (PGTPL) A=0                                                 HYST2780
      IF (PLTPL) A=-PY                                               HYST2790
      RETURN                                                         HYST2800
C                                                                    HYST2810
C RULE 12 - LOADING FORM Q TO M                                      HYST2820
C                                                                    HYST2830
 1200 IF (PLTPL) THEN                                                HYST2840
        IF (XS.EQ.KR .AND. P.GT.FP) GO TO 800                        HYST2850
        GO TO 1300                                                   HYST2860
      ENDIF                                                          HYST2870
      IF ( P .GE.FMAX) GO TO 600                                     HYST2880
      S=XS                                                           HYST2890
      NRL=12                                                         HYST2900
      HRLN=12.1                                                      HYST2910
      A=FMAX                                                         HYST2920
      RETURN                                                         HYST2930
C                                                                    HYST2940
C RULE 13 - LOADING AND UNLOADING INSIDE Q-M-D-P-Y"-Y'               HYST2950
C                                                                    HYST2960
1300  IF (KR.NE.XS) THEN                                             HYST2970
         CALL INTSCT(DITENS,PITENS,D,P,KR,DMAX,FMAX,XS)              HYST2980
         CALL INTSCT(DICOMP,PICOMP,D,P,KR,DX ,FX,XS )                HYST2990
      ELSE                                                           HYST3000
        IF (PGTPL) GO TO 1200                                        HYST3010
        PITENS=P                                                     HYST3020
        DITENS=D                                                     HYST3030
        PICOMP=P                                                     HYST3040
        DICOMP=D                                                     HYST3050
      ENDIF                                                          HYST3060
C--------IS POINT WITHIN PY'-PP-PY" ?                                HYST3080
      IF (PICOMP.LE.PP)  THEN                                        HYST3090
        IF (KC.NE.KT) THEN                                           HYST3100
           CALL INTSCT(DICOMP,PICOMP,D,P,XC,DP,PP,KT )               HYST3110
        ELSE                                                         HYST3120
           IF (PLTPL) GO TO  900                                     HYST3130
           PICOMP=P                                                  HYST3140
           DICOMP=D                                                  HYST3150
        ENDIF                                                        HYST3160
        IF (KR EQ.XS .AND. PLTPL) GO TO  800                         HYST3170
        S2=XT                                                        HYST3180
      ELSE                                                           HYST3190
        S2=KR                                                        HYST3200
      ENDIF                                                          HYST3210
       IF (PGTPL) THEN                                               HYST3220
          IF (P.GT.PITENS) GO TO 1200                                HYST3230
          A=PITENS                                                   HYST3240
          HRLN=13.1                                                  HYST3250
       ENDIF                                                         HYST3260
       IF (PLTPL) THEN                                               HYST3270
          IF (P.LT.PICOMP) GO TO 800                                 HYST3280
          A=PICOMP                                                   HYST3290
          HRLN=13.2                                                  HYST3300
       ENDIF                                                         HYST3310
      S=MAX(KR,S2)                                                   HYST3320
      NRL=13                                                         HYST3330
C                                                                    HYST3340
      RETURN                                                         HYST3350
      END                                                            HYST3360
C                                                                    HYST0010
C--------------------------------------------------------------------HYST0020
C--  BENDING HYSTERESIS MODEL..... HYST03 ..... ....................HYST0030
C--------------------------------------------------------------------HYST0040
C                                                                    HYST0050
      SUBROUTINE HYST03(P,D,PL,DL,VEL,NB,NI,PC,DC,PB,DB,SI,CSE,      HYST0060
     &                  K,RULE,RDIR,A,PMG,PMH,DMG,DMH,BACKB,ALPHA1,ALPHA2,  HYST0070
     &            IR,PR,DR,FR,SE,DY,ESE,PSE,PSEOLD,UPOS,UNEG,EXCR)  HYST0080
C                                                                    HYST0090
      IMPLICIT REAL(A-H,K,O-Z)                                       HYST0100
      IMPLICIT INTEGER(I,J,L-N)                                      HYST0110
      INTEGER DIR,DIRL                                               HYST0120
      CHARACTER*4  FR(1)                                             HYST0130
      LOGICAL  BACKB,REVRSL                                          HYST0140
      DIMENSION PB(1),DB(1),PR(1),DR(1),UPOS(3),UNEG(3),EXCR(6),ESE(3)  HYST0150
      DATA ALPHAA/1.129/,ALPHAB/1.029/                               HYST0160
C                                                                    HYST0170
C     INPUT VARIABLES                                                HYST0180
C     P       - CURRENT LOAD                                         HYST0190
C     D       - CURRENT DISPLACEMENT                                 HYST0200
C     PL      - LAST LOAD                                            HYST0210
C     VEL     - PRODUCT OF LAST AND CURRENT VELOCITY                 HYST0220
C     NB      - NUMBER OF BACKBONE CURVE POINTS                      HYST0230
C     NI      - MAXIMUN NUMBER OF SECONDARY LOOPS                    HYST0240
C     PC      - CRACKING LOAD                                        HYST0250
C     DC      - CRACKING DISPLACEMENT                                HYST0260
C     PB(J)   - BACKBONE CURVE LOADING POINT J, J=1,NB               HYST0270
C     DB(J)   - BACKBONE CURVE DISPLACEMENT POINT J,1,NB             HYST0280
C     SI      - INITIAL STIFFNESS, PC/DC                             HYST0290
C                                                                    HYST0300
C     OUTPUT VARIABLES                                               HYST0310
C     ....AN INITIAL VALUE OF ZERO , OR FALSE                        HYST0320
C                 IS REQUIRED FOR ALL OUTPUT VARIABLES...            HYST0330
C     K       - STIFFNESS                                            HYST0340
C     RULE    - RULE NUMBER, STORED IN THE FORMAT RN.DIR UPON EXIT OF  HYST0350
C               THIS SU_BROU_TINE                                    HYST0360
C     A       - LOAD LIMIT BEFORE NEXT RULE CHANGE, IN THE CURRENT DIR  HYST0370
C     PMG     - MAXIMUN PAST LOAD IN THE POSITIVE DIRECTION          HYST0380
C     PMH     - MAXIMUN PAST LOAD IN THE NEGATIVE DIRECTION          HYST0390
C     DMG     - MAXIMUN PAST DISPLACEMENT IN THE POSITIVE DIRECTION  HYST0400
C     DMH     - MAXIMUN PAST DISPLACEMENT IN THE NEGATIVE DIRECTION  HYST0410
C     FR(I)   - SECONDARY LOOP LOADING POINT, I                      HYST0420
C     DR(I)   - SECONDARY LOOP DISPLACEMENT POINT, I                 HYST0430
C     FR(I)   - SECONDARY LOOP FLAG                                  HYST0440
C     IF      - NUMBER OF SECONDARY LOOPS ACTIVE                     HYST0450
C     ALPHA1  - POSITIVE LOAD DEGRADING STIFFNESS FACTOR             HYST0460
C     ALPHA2  - NEGATIVE LOAD DEGRADING STIFFNESS FACTOR             HYST0470
C     BACKB   - BACKBONE LOADING FLAG                                HYST0480
C                                                                    HYST0490
C     INTERNAL VARIABLES                                             HYST0500
C     J       - BACKBONE CURVE POINT # IN THE CURRENT DIRECTION      HYST0510
C     J0      - TEMPORARY VARIABLE                                   HYST0520
C     I,IT    - SECONDARY LOOP NUMBER                                HYST0530
C     DIR     - CURRENT DIRECTION                                    HYST0540
C               DIR=1, POSITIVE LOADING                              HYST0550
C               DIR=2, POSITIVE UNLOADING                            HYST0560
C               DIR=3, NEGATIVE LOADING                              HYST0570
C               DIR=4, NEGATIVE UNLOADING                            HYST0580
C     DIRL    - LAST DIRECTION                                       HYST0590
C     IRULE   - INTEGER VALUE OF RULE #                              HYST0600
C     LRULE   - INTEGER VALUE OF LAST RULE #                         HYST0610
C     REVRSL  - LOAD REVERSAL FLAG                                   HYST0620
C     AP      - ABSOLUTE VALUE OF THE LOAD, P                        HYST0630
C     PM      - MAXIMUN PAST LOAD IN THE CURRENT DIRECTION           HYST0640
C     DM      - MAXIMUN PAST DISPLACEMENT IN THE CURRENT DIRECTION   HYST0650
C     DMAX    - MAXIMUN PAST DISPLACEMENT IN BOTH DIRECTIONS         HYST0660
C     Q1      - ONE QUARTER OF PM, USED BY UNLOADING CURVES          HYST0670
C     Q3      - THREE QUARTERS OF PM, USED BY UNLOADING CURVES       HYST0680
C     DQ1     - DISPLACEMENT AT Q1 ON UNLOADING BRANCH               HYST0690
C     DQ3     - DISPLACEMENT AT Q3 ON UNLOADING BRANCH               HYST0700
```

```
C      D01     - INTERMEDIATE DISPLACEMENT INTERCEPT OF UNLOADING BRANCH HYST0710
C      D02     - INTERMEDIATE DISPLACEMENT INTERCEPT OF UNLOADING BRANCH HYST0720
C      D0      - DISPLACEMENT INTERCEPT OF UNLOADING BRANCH              HYST0730
C      DP      - DIFFERANCE IN LOAD, USED FOR CALCULATING K             HYST0740
C      DD      - DIFFERANCE IN DISPLACEMENT, USED FOR CALCULATING K     HYST0750
C      PX      - LOAD INTERCEPT OF LOADING AND BACKBONE CURVE           HYST0760
C      DX      - DISPLACEMENT INTERCEPT OF LOADING AND BACKBONE CURVE   HYST0770
C      PC/3    - PC/3                                                   HYST0780
C      DC3     - DISPLACEMENT AT PC3                                    HYST0790
C      P2,D2   - INTERSECTION OF LOADING AND UNLOADING CURVES           HYST0800
C      PI,DI   - SMALL LOOP UNLOADING POINTS                            HYST0810
C      X       - MAXIMUN DISPLACEMENT                                   HYST0820
C      K8      - STIFFNESS FROM RULE 8                                  HYST0830
C      FLAG1   - UPPER SECONDARY LOOP FLAG                              HYST0840
C      FLAG2   - LOWER SECONDARY LOOP FLAG                              HYST0850
C      ALPHAA  - DEGRADING FACTOR FOR FIRST CYCLE                       HYST0860
C      ALPHAB  - DEGRADING FACTOR FOR SUBSEQUENT CYCLES                 HYST0870
C      S       - TEMPORARY STIFFNESS                                    HYST0880
C      S1      - UNLOADING STIFFNESS FROM FS1                           HYST0890
C      S2      - UNLOADING STIFFNESS FROM FS2                           HYST0900
C      S3      - UNLOADING STIFFNESS FROM FS3                           HYST0910
C      SU      - STIFFNESS BETWEEN CURRENT AND RELOADING POINT          HYST0920
C      S02     - STIFFNESS BETWEEN PA AND D0                            HYST0930
C      S03     - STIFFNESS BETWEEN PB AND D0                            HYST0940
C                                                                       HYST0950
C      INTERNAL FUNCTIONS                                               HYST0960
C      FD2     - DISPLACEMENT INTERCEPT OF LOADING AND UNLOADING CURVE  HYST0970
C      FS1     - UNLOADING STIFFNESS WHEN P>.75 PMAX                    HYST0980
C      FS2     - UNLOADING STIFFNESS WHEN  75 PMAX > P > .25 PMAX       HYST0990
C      FS3     - UNLOADING STIFFNESS WHEN .25 PMAX > P                  HYST1000
C      FSL     - LOADING STIFFNESS WHEN  P < PC/3 WITHOUT REVERSAL...   HYST1010
C                                                                       HYST1020
C      SU_BROU_TINES CALLED                                             HYST1030
C      DIRECT  - DETERMINES THE VALUES OF DIR AND DIRL                  HYST1040
C      INTSCT  - DETERMINES THE INTERSECTION OF TWO LINES IN POINT-SLOPE HYST1050
C                FORM                                                   HYST1060
C      KMIN    - DETERMINES THE STIFFNESS BASED ON DP,DD AND K_DEFULT   HYST1070
C                                                                       HYST1080
      FS1(X)= SI *(DC/X)**.294                                          HYST1090
      FS2(X)= SI *(0.8344*(DC/X-1) +1)                                  HYST1100
      FS3(X)= SI *(0.9092*(DC/X-1) +1)                                  HYST1110
      FSL(X)= SI *(DC/X)**.285                                          HYST1120
      FD2(X)= DM-.05*PM/(FS1(X))                                        HYST1130
C                                                                       HYST1140
 1000 IF (IR GT.0) WRITE(108,1020) (I,PR(I),DR(I),FR(I),I=1,IR)         HYST1150
 1020 FORMAT(' I=',G13.5,'    PR=',G13.5,'    DR=',G13.5,               HYST1160
     &       '    FR=',A)                                              HYST1170
C                                                                       HYST1180
      AP=ABS(P)                                                         HYST1190
C.... . DETERMINE IF ELASTIC                                            HYST1200
      IF (RULE EQ.0 .AND. AP.LT.PC ) THEN                               HYST1210
            K=SI                                                        HYST1220
            SE=X                                                        HYST1230
            RETURN                                                      HYST1240
      ENDIF                                                             HYST1250
C                                                                       HYST1260
C..... DETERMINE IF WALL IS INACTIVE, IF SO RETURN                      HYST1270
      IF (RULE.LT.0.0) RETURN                                           HYST1280
C                                                                       HYST1290
C..... DETERMINE CURRENT DIRECTION                                      HYST1300
      LRULE=INT(RULE)                                                   HYST1310
      DIR =RDIR                                                         HYST1320
      CALL DIRECT(DIR,DIRL,P,PL,VEL)                                    HYST1330
      RDIR =DIR                                                         HYST1340
C                                                                       HYST1350
C.. ... DETERMINE MAXIMUN AND MINIMUN VALUES                            HYST1360
            PMG=MAX(P,PMG, PC)                                          HYST1370
            DMG=MAX(D,DMG, DC)                                          HYST1380
            PMH=MIN(P,PMH,-PC)                                          HYST1390
            DMH=MIN(D,DMH,-DC)                                          HYST1400
            DMAX=MAX(DMG,-DMH)                                          HYST1410
            PMAX=MAX(PMG,-PMH)                                          HYST1420
C                                                                       HYST1430
C..  . SET MAXIMUN AND MINIMUN VALUES...                                HYST1440
      IF (DIR EQ 1 .OR  DIR.EQ.2 ) THEN                                 HYST1450
            PM=PMG                                                      HYST1460
            DM=DMG                                                      HYST1470
      ELSE IF (DIR.EQ.3 .OR. DIR.EQ 4) THEN                             HYST1480
            PM=PMH                                                      HYST1490
            DM=DMH                                                      HYST1500
      ENDIF                                                             HYST1510
C                                                                       HYST1520
C... .. CALCULATE EQUIVALENT UNLOADING STIFFNESS FOR ENERGY..           HYST1530
        IF (BACKB .OR  RULE.EQ.1 0) THEN                                HYST1540
            S1=FS1(DMAX)                                                HYST1550
            S2=FS2(DMAX)                                                HYST1560
            S3=FS3(DMAX)                                                HYST1570
            D0 =DM - PM*(DMH-DMG)/(PMH-PMG)                             HYST1580
            Q1=PM/4                                                     HYST1590
            Q3=3*Q1                                                     HYST1600
            DQ3=DM -0.25*PM/S1                                          HYST1610
            S02=KMIN((DQ3-D0),Q3,S1)                                    HYST1620
            DQ1=DQ3-0.50*PM/MAX(S2,S02)                                 HYST1630
            S03=KMIN((DQ1-D0),Q1,S1)                                    HYST1640
            SE=0.5*PM**2/((PM-Q3)*0.5*(DM-DQ3)+(Q3+Q1)*0.5*(DQ3-DQ1)    HYST1650
     &           +(Q1)*0.5*(Q1)/MAX(S3,S03) )                           HYST1660
        ENDIF                                                           HYST1670
C                                                                       HYST1680
C.... . CHECK TO SEE IF LAST RULE IS STILL IN EFFECT                    HYST1690
      IF (DIR.EQ.DIRL .AND.                                             HYST1700
     &   (DIR.EQ.1 .AND.  P.LT.A) .OR.  (DIR.EQ.2 .AND. P.GT.A)  .OR.   HYST1710
     &   (DIR.EQ.3  AND  P.GT.A) .OR. (DIR.EQ.4 .AND. P.LT.A))) THEN    HYST1720
          RETURN                                                        HYST1730
       ENDIF                                                            HYST1740
C                                                                       HYST1750
C     . CALC EXCURSION RATIO AT EVERY ZERO CROSSING...                  HYST1760
      IF (P*PL.LE.0) CALL DNE(0,UPOS,EXCR, DY,DLI,ESE,PSE,PSEOLD,CSE)   HYST1770
C                                                                       HYST1780
C . . . ERASE SMALL LOOP FLAGS                                          HYST1790
      IF (IR.GT.0) THEN                                                 HYST1800
        IT=IR                                                           HYST1810
        DO 10 I=1,IT                                                    HYST1820
          IF ((FR(I).EQ.'L' .AND.  ( P.LE.PR(I)  .OR.  D.LE.DR(I) ))  .OR. HYST1830
     &        (FR(I).EQ.'7' .AND   ( P.GE.PR(I)  OR.  D GE DR(I) ))) THEN HYST1840
            IR=I-1                                                      HYST1850
            GO TO 15                                                    HYST1860
          ENDIF                                                         HYST1870
   10   CONTINUE                                                        HYST1880
        ENDIF                                                           HYST1890
   15 I=IR+1                                                            HYST1900
C                                                                       HYST1910
      IF (DIR.EQ.1 .OR. DIR.EQ 3 )  THEN                                HYST1920
C ..    LOADING RULES .... ... ... ..... .... .  ... .... RULES 1,6 11 HYST1930
C                                                                       HYST1940
C ..... LOADING ON BACKBONE                                    RULE 1   HYST1950
    1     IF (BACKB  OR. RULE.EQ 0.0) THEN                              HYST1960
            X=0                                                         HYST1970
            DO 18 J=2,NB                                                HYST1980
              DD=DB(J)-ABS(D)                                           HYST1990
              DP=PB(J)-AP                                               HYST2000
              IF (AP.LT.PB(J) .AND  (DD*DP).GT.0 ) THEN                 HYST2010
                S=DP/DD                                                 HYST2020
                IF (S.GT X) THEN                                        HYST2030
                  K=S                                                   HYST2040
                  A=SIGN(PB(J),P)                                       HYST2050
                ENDIF                                                   HYST2060
              ENDIF                                                     HYST2070
   18       CONTINUE                                                    HYST2080
            RULE=1                                                      HYST2090
            BACKB=.TRUE                                                 HYST2100
            IR=0                                                        HYST2110
            IF (DIR.EQ.1) THEN                                          HYST2120
                ALPHA1=ALPHAA                                           HYST2130
            ELSE                                                        HYST2140
                ALPHA2=ALPHAA                                           HYST2150
            ENDIF                                                       HYST2160
C                                                                       HYST2170
C  .... LOADING AFTER UNLOADING                                         HYST2180
          ELSE                                                          HYST2190
C.............  RELOADING INSIDE SMALL LOOPS                  RULE 11   HYST2200
   20       I=I-1                                                       HYST2210
            IF (I.GT.0 ) THEN                                           HYST2220
                IF (.NOT.(( FR(I) EQ.'7' .AND.  DIR.EQ.1 )  .OR.        HYST2230
     &              ( FR(I).EQ.'L' .AND   DIR EQ 3 ))  GO TO 20         HYST2240
                K=KMIN((D-DR(I)),(P-PR(I)),SI)                          HYST2250
                A=PR(I)                                                 HYST2260
                RULE=11-I/1000.                                         HYST2270
C                                                                       HYST2280
C. ..... ... INITIAL RELOADING BELOW PC/3                               HYST2290
C..... ... ... ... WITH REVERSAL                              RULE 8    HYST2300
            ELSE IF (AP.LT.(PC/3) )THEN                                 HYST2310
              BACKB=.FALSE.                                             HYST2320
              IF ((DIR.EQ.1 .AND  DIRL EQ.4) .OR.                       HYST2330
     &            (DIR.EQ.3 .AND. DIRL.EQ.2)) THEN                      HYST2340
                P2=.95*PM                                               HYST2350
                D2=FD2(DMAX)                                            HYST2360
                K=KMIN((D-D2),(P P2),SI)                                HYST2370
                K8=K                                                    HYST2380
                A= P2                                                   HYST2390
                RULE=8                                                  HYST2400
C                                                                       HYST2410
                PC3=SIGN(PC/3,PM)                                       HYST2420
                D01=DM-PM*(1./FS1(DMAX)+2./FS2(DMAX)+1./FS3(DMAX))/6.   HYST2430
                D02=DM - PM*(DMH-DMG)/(PMH-PMG)                         HYST2440
                IF (DIR.EQ.1) D0=MAX(D01,D02)                           HYST2450
                IF (DIR.EQ.3) D0=MIN(D01,D02)                           HYST2460
                DC3=D0+PC3/FSL(DMAX)                                    HYST2470
                S=KMIN((D-DC3),(P-PC3),SI)                              HYST2480
                IF (S GT.K) THEN                                        HYST2490
                    K=S                                                 HYST2500
                    A= PC3                                              HYST2510
                    RULE=8.1                                            HYST2520
                ENDIF                                                   HYST2530
C                                                                       HYST2540
C.. ...  . .. ..... WITH REVERSAL, FAR SIDE LOADING    RULE 9           HYST2550
              IF ((DIR.EQ.1 .AND.  D.LT.0) .OR                          HYST2560
     &            (DIR.EQ.3 .AND. D.GT.0)) THEN                         HYST2570
                K=K8                                                    HYST2580
                DO 40 J=2,NB                                            HYST2590
                    IF ( PB(J) .LT.ABS(PM) ) GO TO 40                   HYST2600
                    DD= -SIGN(DB(J),D)*D                                HYST2610
                    DP= -SIGN(PB(J),D)*P                                HYST2620
                    IF ( (DD*DP).LE.0.) GO TO 40                        HYST2630
                      S=DP/DD                                           HYST2640
                      IF (S.GT.K  AND. S.LE.SI) THEN                    HYST2650
                        K=S                                             HYST2660
                        A=SIGN(PB(J),PM)                                HYST2670
                        RULE=9                                          HYST2680
                        BACKB=.TRUE.                                    HYST2690
                      ENDIF                                             HYST2700
   40               CONTINUE                                           HYST2710
                ENDIF                                                   HYST2720
C                                                                       HYST2730
C.... ... ... ..... ..... WITHOUT REVERSAL                    RULE 6   HYST2740
              ELSE                                                      HYST2750
                K=FSL(DMAX)                                             HYST2760
                A= SIGN((PC/3),PM)                                      HYST2770
                RULE=6                                                  HYST2780
              ENDIF                                                     HYST2790
C                                                                       HYST2800
C.... ... ... MIDRANGE RELOADING CURVE                        RULE 7    HYST2810
              ELSE IF (AP.LT.ABS(.95*PM) ) THEN                         HYST2820
                BACKB=.FALSE.                                           HYST2830
                P2=.95*PM                                               HYST2840
                D2=FD2(DMAX)                                            HYST2850
                K=KMIN((D-D2),(P-P2),SI)                                HYST2860
                A=P2                                                    HYST2870
                RULE=7                                                  HYST2880
C                                                                       HYST2890
C.. ..... . .  LOADING TOWARDS THE BACKBONE CURVE             RULE 10   HYST2900
              ELSE                                                      HYST2910
                BACKB=.TRUE.                                            HYST2920
                IF (AP.GE.ABS(PM) .OR. (ABS(DM).LE.DC) GO TO 1          HYST2930
                IR=0                                                    HYST2940
                IF (DIR.EQ.1) THEN                                      HYST2950
                    IF (ALPHA1.NE.ALPHAA .AND  ALPHA1.NE.ALPHAB)        HYST2960
     &                        ALPHA1=ALPHAA                             HYST2970
                    DD=ALPHA1*DM-D                                      HYST2980
                ELSE                                                    HYST2990
                    IF (ALPHA2.NE.ALPHAA .AND. ALPHA2.NE.ALPHAB)        HYST3000
     &                        ALPHA2=ALPHAA                             HYST3010
                    DD=ALPHA2*DM-D                                      HYST3020
                ENDIF                                                   HYST3030
                DP=PM-P                                                 HYST3040
                IF ((DP*DD).GT.0) THEN                                  HYST3050
                  K=MIN( (DP/DD),SI)                                    HYST3060
                ELSE                                                    HYST3070
                  K=SI                                                  HYST3080
                ENDIF                                                   HYST3090
                DO 50 J=2,NB                                            HYST3100
                  IF ( PB(J).LE.ABS(PM) ) GO TO 50                      HYST3110
                  J0=J-1                                                HYST3120
                  DD=DB(J)-DB(J0)                                       HYST3130
                  DP=PB(J)-PB(J0)                                       HYST3140
                  IF ((DP*DD).LE.0.) GO TO 50                           HYST3150
                   CALL INTSCT(DX,PX,D,P,K,SIGN(DB(J),P),               HYST3160
     &                         SIGN(PB(J),P),(DP/DD) )                  HYST3170
                   PX=PX                                                HYST3180
                   DX=DX                                                HYST3190
                   IF (ABS(DX).GE.DB(J0)  AND. ABS(DX).LE.DB(J).AND.    HYST3200
     &                ABS(PX).GT.ABS(PM)  .AND.  PM*PX GT.0) THEN       HYST3210
                     IF (AP.GT.ABS(PX)) GO TO 1                         HYST3220
                     A=-PX                                              HYST3230
                     RULE=10                                            HYST3240
                     IF (DIR.EQ.1)  THEN                                HYST3250
                         ALPHA1=ALPHAB                                  HYST3260
                     ELSE                                               HYST3270
                         ALPHA2=ALPHAB                                  HYST3280
                     ENDIF                                              HYST3290
                     GO TO 100                                          HYST3300
                   ENDIF                                                HYST3310
   50           CONTINUE                                               HYST3320
                GO TO 1                                                 HYST3330
              ENDIF                                                     HYST3340
        ENDIF                                                           HYST3350
C                                                                       HYST3360
C                                                                       HYST3370
      ELSE                                                              HYST3380
C .... UNLOADING RULES .  .... . . . . . . . . . . . . .  RULES 2-5     HYST3390
C                                                                       HYST3400
C..... INITILIZE VALUES                                                 HYST3410
            BACKB=.FALSE.                                               HYST3420
            S1=FS1(DMAX)                                                HYST3430
            S2=FS2(DMAX)                                                HYST3440
            S3=FS3(DMAX)                                                HYST3450
            D0 =DM - PM*(DMH-DMG)/(PMH-PMG)                             HYST3460
            Q1=PM/4                                                     HYST3470
            Q3=3*Q1                                                     HYST3480
            DQ3=DM -0.25*PM/S1                                          HYST3490
            S02=KMIN((DQ3-D0),Q3,SI)                                    HYST3500
            DQ1=DQ3-0.50*PM/MAX(S2,S02)                                 HYST3510
            S03=KMIN((DQ1-D0),Q1,SI)                                    HYST3520
            SU=0.                                                       HYST3530
C                                                                       HYST3540
C                                                                       HYST3550
C .... . ... UNLOADING INSIDE SMALL POSITIVE LOOPS             RULE 5   HYST3560
            I=IR+1                                                      HYST3570
   70       I=I-1                                                       HYST3580
            IF (I.GT.0) THEN                                            HYST3590
              IF ( NOT ((DIR.EQ 2 .AND. FR(I).EQ 'L')   .OR             HYST3600
     &                  (DIR.EQ.4  AND. FR(I).EQ '7'))) ) GO TO 70      HYST3610
              IF ((DIR EQ.2  AND ( (P GT Q3  AND  PR(I) GT.Q3) .OR      HYST3620
```

```
     &                (Q3.GE.P .AND.   P.GT.Q1 .AND. PR(I).GT.Q1) .OR.   HYST3630
     &                (Q1.GE.P .AND.   PR(I).GT.0 ))) .OR.               HYST3640
     &               (DIR.EQ.4 .AND.( (P.LT.Q3 .AND. PR(I).LT.Q3) .OR.   HYST3650
     &                (Q3.LE.P .AND.   P.LT.Q1 .AND. PR(I).LT.Q1)  OR.   HYST3660
     &                (Q1.LE.P .AND.   PR(I).LT.0 ))) ) THEN             HYST3670
                     X=(PR(I)-P)/(DR(I)-D)                              HYST3680
                     IF (X.LE.0 .OR. X.GT.SI) GO TO 70                  HYST3690
                     A=PR(I)                                            HYST3700
                     RULE=5+I/1000.                                     HYST3710
                     GO TO 100                                          HYST3720
                  ELSE                                                  HYST3730
                     SU=KMIN((DR(I)-D),(PR(I)-P),SI)                    HYST3740
                  ENDIF                                                 HYST3750
               ENDIF                                                    HYST3760
C                                                                       HYST3770
C..... ... CALC DUCTILITY AND EXCURSION RATIO...                        HYST3780
               IF  (RULE.EQ.1  OR. RULE.GT.5)  THEN                     HYST3790
                 IF (DIR.EQ 2) THEN                                     HYST3800
                   DLI= MAX( D , DL )                                   HYST3810
                   CALL DNE(1,UPOS,EXCR, DY,DLI,ESE(2),PSE,PSEOLD,CSE)  HYST3820
                 ELSE IF (DIR.EQ.4) THEN                                HYST3830
                   DLI= MIN( D , DL )                                   HYST3840
                   CALL DNE(1,UNEG,EXCR,-DY,DLI,ESE(3),PSE,PSEOLD,CSE)  HYST3850
                 ENDIF                                                  HYST3860
               ENDIF                                                    HYST3870
C                                                                       HYST3880
C. ..... UNLOADING ON THE TOP SEGMENT                          RULE 2   HYST3890
               IF(AP.GT.ABS(Q1)) THEN                                   HYST3900
                  X=MAX(S1,SU)                                          HYST3910
                  A=Q1                                                  HYST3920
                  RULE=2                                                HYST3930
C.. ..... ... UNLOADING ON THE MIDDLE SEGMENT                  RULE 3   HYST3940
               ELSE IF(AP.GT.ABS(Q1)) THEN,                             HYST3950
                  X=MAX(S2,SC2,SU)                                      HYST3960
                  A=Q1                                                  HYST3970
                  RULE=3                                                HYST3980
C.. ..... ... UNLOADING ON THE BOTTOM SEGMENT                  RULE 4   HYST3990
               ELSE                                                     HYST4000
                  X=MAX(S3,SC3,SU)                                      HYST4010
                  A=0.                                                  HYST4020
                  RULE=4                                                HYST4030
               ENDIF                                                    HYST4040
            ENDIF                                                       HYST4050
C..... SAVE REVERSAL POINTS FOR SMALL LOOPS..... ..... ... ....        HYST4060
 100        IRULE=INT(RULE)                                            HYST4070
            REVRSL=IRULE.EQ.1.OR.IRULE.EQ.10.OR.LRULE.EQ.1.OR.LRULE.EQ.10  HYST4080
            IF ((DIRL.EQ.1 .AND. DIR.EQ.2 .AND. .NOT. REVRSL).OR.       HYST4090
           &    (DIRL.EQ.4 .AND. DIR.EQ.3) ) THEN                       HYST4100
                IF (IR.LT.NI)  THEN                                     HYST4110
                     IR=IR+1                                            HYST4120
                     PR(IR)=P                                           HYST4130
                     DR(IR)=D                                           HYST4140
                     FR(IR)='7'                                         HYST4150
                ENDIF                                                   HYST4160
            ELSE IF ((DIRL.EQ.3 .AND. DIR.EQ.4 .AND. .NOT. REVRSL)      HYST4170
           &                  .OR. (DIRL.EQ.2 .AND. DIR.EQ.1) ) THEN   HYST4180
                IF (IR.LT.NI)  THEN                                     HYST4190
                     IR=IR+1                                            HYST4200
                     PR(IR)=P                                           HYST4210
                     DR(IR)=D                                           HYST4220
                     FR(IR)='L'                                         HYST4230
                ENDIF                                                   HYST4240
            ENDIF                                                       HYST4250
            IF ((1.05*SI).GT.K .AND. K.GT.0 ) RETURN                    HYST4260
C                                                                       HYST4270
            K=0                                                         HYST4280
            RULE=-RULE                                                  HYST4290
C                                                                       HYST4300
            RETURN                                                      HYST4310
            END                                                         HYST4320
C                                                                       HYST0010
C-----------------------------------------------------------------------HYST0020
C--- SHEAR HYSTERESIS MODEL.. .. HYST04 ... ..... .. .. ..             HYST0030
C-----------------------------------------------------------------------HYST0040
C                                                                       HYST0050
      SUBROUTINE HYST04(P,D,PL,DL,VEL,NS,NI,                            HYST0060
     &    PC    ,DC    ,PS    ,DS    ,SI    ,CSE    ,DY    ,          HYST0070
     &    X     ,RULE  ,RDIR  ,A     ,PMG   ,PMH    ,                 HYST0080
     &    DMG   ,DMH   ,BACKB ,SR1   ,SR2   ,SR3    ,                 HYST0090
     &    SRM   ,IR    ,SE    ,UPOS  ,UNEG  ,EXCR   ,                 HYST0100
     &    ESE   ,PSE ,PSEOLD,FR     ,DR     ,FR     )                 HYST0110
C                                                                       HYST0120
      IMPLICIT REAL(A-H,K,O-Z)                                          HYST0130
      IMPLICIT INTEGER(I,J,L-N)                                         HYST0140
      INTEGER DIR,DIRL                                                  HYST0150
      LOGICAL  BACKB,CRACKD,REVRSL                                      HYST0160
      CHARACTER*4  FR(NI)                                               HYST0170
      DIMENSION PS(NS),DS(NS),PR(NI),DR(NI)                             HYST0180
      DIMENSION UPOS(3),UNEG(3),EXCR(6),ESE(3)                          HYST0190
      DATA ALPHA/1.04/                                                  HYST0200
C     INPUT VARIABLES                                                   HYST0210
C     P       - CURRENT LOAD                                            HYST0220
C     D       - CURRENT DISPLACEMENT                                    HYST0230
C     PL      - LAST LOAD                                               HYST0240
C     VEL     - PRODUCT OF LAST AND CURRENT VELOCITY                    HYST0250
C     NS      - NUMBER OF BACKBONE CURVE POINTS                         HYST0260
C     NI      - MAXIMUN NUMBER OF SECONDARY LOOPS                       HYST0270
C     PC      - CRACKING LOAD                                           HYST0280
C     DC      - CRACKING DISPLACEMENT                                   HYST0290
C     SI      - INITIAL STIFFNESS PC/DC                                 HYST0300
C     PS      - BACKBONE CURVE LOADING POINTS                           HYST0310
C     DS      - BACKBONE CURVE DISPLACEMENT POINTS                      HYST0320
C                                                                       HYST0330
C     OUTPUT VARIABLES                                                  HYST0340
C      AN INITIAL VALUE OF ZERO IS REQUIRED FOR ALL OUTPUT VARIABLES..  HYST0350
C     K       - STIFFNESS                                               HYST0360
C     RULE    - RULE NUMBER, STORED IN THE FOR_MAT RN.DIR UPON EXIT OF  HYST0370
C               THIS SU_BROU_TINE                                       HYST0380
C     A       - LOAD LIMIT BEFORE NEXT RULE CHANGE                      HYST0390
C     PMG     - MAXIMUN PAST LOAD IN THE POSITIVE DIRECTION             HYST0400
C     PMH     - MAXIMUN PAST LOAD IN THE NEGATIVE DIRECTION             HYST0410
C     DMG     - MAXIMUN PAST DISPLACEMENT IN THE POSITIVE DIRECTION     HYST0420
C     DMH     - MAXIMUN PAST DISPLACEMENT IN THE NEGATIVE DIRECTION     HYST0430
C     PR      - SECONDARY LOOP LOADING POINT                            HYST0440
C     DP      - SECONDARY LOOP DISPLACEMENT POINT                       HYST0450
C     FR      - SECONDARY LOOP FLAG                                     HYST0460
C     IR      - SECONDARY LOOP NUMBER                                   HYST0470
C     BACKB   - BACKBONE LOADING FLAG                                   HYST0480
C     SR1     - RELOADING STIFFNESS FROM RULE 8 - USED BY RULE 11       HYST0490
C     SR2     - RELOADING STIFFNESS FROM RULE 9 - USED BY RULE 11       HYST0500
C     SR3     - RELOADING STIFFNESS FROM RULE 7 - USED BY RULE 11       HYST0510
C     SRM     - RELOADING STIFFNESS FROM RULES 8 & 9   USED BY RULE 11  HYST0520
C                                                                       HYST0530
C     INTERNAL VARIABLES                                                HYST0540
C     J       - BACKBONE CURVE POINT # IN THE CURRENT DIRECTION         HYST0550
C     DIR     - CURRENT DIRECTION                                       HYST0560
C               DIR-1, POSITIVE LOADING                                 HYST0570
C               DIR-2, POSITIVE UNLOADING                               HYST0580
C               DIR-3, NEGATIVE LOADING                                 HYST0590
C               DIR-4, NEGATIVE UNLOADING                               HYST0600
C     DIRL    - LAST DIRECTION                                          HYST0610
C     IRULE   - INTEGER VALUE OF RULE                                   HYST0620
C     LRULE   - INTEGER VALUE OF LAST RULE #                            HYST0630
C     CRACKD  - CRACKED WALL FLAG                                       HYST0640
C     REVRSL  - LOAD REVERSAL FLAG                                      HYST0650
C     PC2     - PC/2, USED BY LOADING CURVES                            HYST0660
C     DC2     - DISPLACEMENT CORRESPONDING TO PC2 ON THE LOADING BRANCH  HYST0670
C     S       - INTERMEDIATE STIFFNESS                                  HYST0680
C     SR      - RELOADING STIFFNESS                                     HYST0690
C     SRP     - MINIMUN RELOADING STIFFNESS                             HYST0700
C     DC      - DISPLACEMENT INTERSEPT OF PEAK DISPLACEMENTS            HYST0710
C     DCP     - DISPLACEMENT INTERSEPT OF THE UNLOADING BRANCH          HYST0720
C     AP      - ABSOLUTE VALUE OF CURRENT LOAD P                        HYST0730
C     PM      - MAXIMUN PAST LOAD IN THE CURRENT DIRECTION              HYST0740
C     DM      - MAXIMUN PAST DISPLACEMENT IN THE CURRENT DIRECTION      HYST0750
C     DMAX    - MAXIMUN PAST DISPLACEMENT IN EITHER DIRECTION           HYST0760
C     PMAX    - MAXIMUN PAST LOAD IN EITHER DIRECTION                   HYST0770
C     PA      - PM-PC, USED BY UNLOADING CURVES                         HYST0780
C     PB      - PC/2, USED BY UNLOADING CURVES                          HYST0790
C     DA      - DISPLACEMENT AT PA, USED BY UNLOADING CURVES            HYST0800
C     DB      - DISPLACEMENT AT PB, USED BY UNLOADING CURVES            HYST0810
C     DP      - DIFFERANCE IN LOAD, USED FOR CALCULATING X              HYST0820
C     DD      - DIFFERANCE IN DISPLACEMENT, USED FOR CALCULATING X      HYST0830
C     PX      - LOAD INTERCEPT OF LOADING AND BACKBONE CURVE            HYST0840
C     DX      - DISPLACEMENT INTERCEPT OF LOADING AND BACKBONE CURVE    HYST0850
C     P2,D2,  - POINT ON THE LOADING CURVE BETWEEN RULES 7 AND 10       HYST0860
C     X       - MAXIMUN DISPLACEMENT                                    HYST0870
C               - DISPLACEMENT INTERCEPT FOR RULE 11                    HYST0880
C     X1      - 75% PC DISPLACEMENT  FOR RULE 11                        HYST0890
C     X2      - 25% PC DISPLACEMENT  FOR RULE 11                        HYST0900
C     I,IT    - TEMPORARY LABEL FOR SMALL LOOP LOADING AND UNLOADING PTSHYST0910
C     J0      - J-1                                                     HYST0920
C     ALPHA   - DEGRADING STIFFNESS FACTOR                              HYST0930
C     ALPHAP  - DEGRADING STIFFNESS FACTOR USED BY RULE 10              HYST0940
C     S1      - STIFFNESS FROM FUNCTION FS1                             HYST0950
C     S2      - STIFFNESS FROM FUNCTION FS2                             HYST0960
C     S3      - STIFFNESS FROM FUNCTION FS3                             HYST0970
C     SU      - STIFFNESS BETWEEN CURRENT AND RELOADING POINT           HYST0980
C     SC2     - STIFFNESS BETWEEN PA AND D0                             HYST0990
C     SC3     - STIFFNESS BETWEEN PB AND D0                             HYST1000
C                                                                       HYST1010
C     INTERNAL FUNCTIONS                                                HYST1020
C     FD2     - DISPLACEMENT INTERCEPT OF LOADING AND UNLOADING CURVE   HYST1030
C     FS1     - UNLOADING STIFFNESS WHEN P>(PM-PC)-PA                   HYST1040
C     FS2     - UNLOADING STIFFNESS WHEN PA>P)(PC/2]-PB                 HYST1050
C     FS3     - UNLOADING STIFFNESS WHEN    P)PB                        HYST1060
C     FSR     - LOADING BELOW PC/2, WITH PINCHING                       HYST1070
C                                                                       HYST1080
C     SU_BROU_TINES CALLED                                              HYST1090
C     DIRECT  - DETERMINES THE VALUES OF DIR AND DIRL                   HYST1100
C     INTSCT  - DETERMINES THE INTERSECTION OF TWO LINES IN POINT-SLOPE  HYST1110
C               FORM                                                    HYST1120
C     XMIN    - DETERMINES THE STIFFNESS BASED ON DP,DD AND K_DEPULT    HYST1130
C                                                                       HYST1140
C                                                                       HYST1150
C UNSYMMETRIC COIEFFICIENTS                                             HYST1160
            FS1(X)= SI *MIN((1.4675*(DC/X)**.343 ),1.)                  HYST1170
            FS2(X)= SI *MIN((0.7761*(DC/X)**.3195),1.)                  HYST1180
            FS3(X)= SI *MIN((0.0707*(DC/X)*1.369 ),1.)                  HYST1190
      FSR(X)= SI *MIN((   (ABS(DC/X))**1.02),1.)                        HYST1200
      FD2(X)= SIGN(DMAX,PM)-.05*SIGN(PMAX,PM)/(FS1(X))                  HYST1210
C                                                                       HYST1220
 1000 IF (IR.GT.0) WRITE(108,1020) (I,PR(I),DR(I),FR(I),I=1,IR)         HYST1230
 1020 FORMAT(5X,'I=',G13.5,'     PR=',G13.5,'     DR=',G13.5,'     FR=',A) HYST1240
C                                                                       HYST1250
      AP=ABS(P)                                                         HYST1260
C                                                                       HYST1270
C ..... DETERMINE IF ELASTIC                                            HYST1280
      IF (RULE.EQ.0 .AND. AP .LT.PC ) THEN                              HYST1290
            K=SI                                                        HYST1300
            SE=K                                                        HYST1310
            RETURN                                                      HYST1320
      ENDIF                                                             HYST1330
C                                                                       HYST1340
C..... DETERMINE IF WALL IS INACTIVE, IF SO RE_TURN                     HYST1350
      IF (RULE.LT 0.0) RETURN                                           HYST1360
C                                                                       HYST1370
C. ...  DETERMINE CURRENT DIRECTION                                     HYST1380
      LRULE=INT(RULE)                                                   HYST1390
      DIR =RDIR                                                         HYST1400
      CALL DIRECT(DIR,DIRL,P,PL,VEL)                                    HYST1410
      RDIR =DIR                                                         HYST1420
C                                                                       HYST1430
C. . .. DETERMINE MAXIMUN AND MINIMUN VALUES                            HYST1440
      PMG=MAX(PMG, PC,P)                                                HYST1450
      DMG=MAX(DMG, DC,D)                                                HYST1460
      PMH=MIN(PMH,-PC,P)                                                HYST1470
      DMH=MIN(DMH,-DC,D)                                                HYST1480
      DMAX=MAX(DMG,-DMH)                                                HYST1490
      PMAX=MAX(PMG,-PMH)                                                HYST1500
C                                                                       HYST1510
C ..... SET MAXIMUN AND MINIMUN VALUES...                               HYST1520
      IF (DIR.EQ.1  .OR.  DIR.EQ.2 ) THEN                               HYST1530
            PM=PMG                                                      HYST1540
            DM=DMG                                                      HYST1550
      ELSE IF (DIR.EQ.3 .OR. DIR.EQ.4) THEN                             HYST1560
            PM=PMH                                                      HYST1570
            DM=DMH                                                      HYST1580
      ENDIF                                                             HYST1590
C                                                                       HYST1600
C.. ... CALCULATE EQUIVALENT UNLOADING STIFFNESS FOR ENERGY...          HYST1610
         IF (BACKB .OR. RULE.EQ 0.0) THEN                               HYST1620
                  S1=FS1(DMAX)                                          HYST1630
                  S2=FS2(DMAX)                                          HYST1640
                  S3=FS3(DMAX)                                          HYST1650
                  D0=DM-PM*(DMG-DMH)/(PMG-PMH)                          HYST1660
                  PA=PM-SIGN(PC,PM)                                     HYST1670
                  DA=DM-SIGN(PC,PM)/S1                                  HYST1680
                  PB=SIGN( (MIN(ABS(PA),(PC/2)) ),PM)                   HYST1690
                  S02=KMIN((DA-D0),PA,SI)                               HYST1700
                  DB=DA-(PA-PB)/MAX(S2,S02)                             HYST1710
                  S03=KMIN((DB-D0),PB,SI)                               HYST1720
                  SE=0.5*PM**2/((PM+PA)*0.5*(DM-DA)                     HYST1730
     &               +(PA+PB)*0.5*(DA-DB)+(PB)*0.5*(PB)/MAX(S3,S03))    HYST1740
         ENDIF                                                          HYST1750
C                                                                       HYST1760
C,..... CHECK TO SEE IF LAST RULE IS STILL IN EFFECT                    HYST1770
      IF (DIR.EQ.DIRL .AND.                                             HYST1780
     &  ((DIR.EQ.1 .AND. P.LT.A) .OR. (DIR.EQ.2 .AND. P.GT.A)  .OR.     HYST1790
     &  (DIR.EQ.3 .AND. P.GT.A) .OR. (DIR.EQ.4 .AND. P.LT.A))) THEN     HYST1800
         RETURN                                                         HYST1810
      ENDIF                                                             HYST1820
C                                                                       HYST1830
C.. ... CALC EXCURSION RATIO AT EVERY ZERO CROSSING. .                  HYST1840
      IF (P*PL.LE.0) CALL DNE(0,UPOS,EXCR, DY,DLI,ESE,PSE,PSEOLD,CSE)   HYST1850
C                                                                       HYST1860
C . .   ERASE SMALL LOOP FLAGS                                          HYST1870
      IF (IR.GT.0) THEN                                                 HYST1880
        IT-IR                                                           HYST1890
        DO 10 I-1,IT                                                    HYST1900
          IF ((FR(I) EQ.'L' .AND  ( P.LE.PR(I) .OR  D LE.DR(I) ))  .OR. HYST1910
     &         (FR(I).EQ '7' .AND  ( P.GE PR(I) .OR. D.GE DR(I) )) THEN HYST1920
            IR=I-1                                                      HYST1930
            GO TO 15                                                    HYST1940
          ENDIF                                                         HYST1950
   10   CONTINUE                                                        HYST1960
        ENDIF                                                           HYST1970
   15 I-IR+1                                                            HYST1980
C                                                                       HYST1990
      IF (DIR.EQ 1  OR. DIR.EQ 3.)  THEN                                HYST2000
C. .   LOADING RULES  . . . . . .. .. . ..... .. .. .. ..... RULES 1,5-11HYST2010
        PC2-SIGN((PC/2),PM)                                             HYST2020
C                                                                       HYST2030
C. . .  LOADING ON BACKBONE                                  RULE 1 HYST2040
   30   IF (BACKB  OR. RULE.EQ.0.0) THEN                                HYST2050
          K=0.                                                          HYST2060
          DO 40 J=2,NS                                                  HYST2070
            DD=DS(J)-ABS(D)                                             HYST2080
            DP=PS(J)-AP                                                 HYST2090
            IF (AP.LT PS(J) .AND  (DD*DP) GT.0.) THEN                   HYST2100
              S=DP/DD                                                   HYST2110
              IF (S.GT.K .AND  S.LE.SI) THEN                            HYST2120
                K=S                                                     HYST2130
                A-SIGN(PS(J),P)                                         HYST2140
              ENDIF                                                     HYST2150
            ENDIF                                                       HYST2160
   40     CONTINUE                                                     HYST2170
          RULE=1                                                        HYST2180
          BACKB- TRUE                                                   HYST2190
          IR=0                                                          HYST2200
C                                                                       HYST2210
C. .   LOADING AFTER UNLOADING                            RULES 6-11HYST2220
```

```
      ELSE                                                              HYST2230
           CRACKD=(ABS(DM) GT.DC)                                       HYST2240
           REVRSL=((DIR.EQ.1 .AND. DIRL.EQ.4) .OR. LRULE.EQ 8           HYST2250
     &        .OR  (DIR.EQ.3 .AND. DIRL.EQ.2) .OR. LRULE.EQ.9          HYST2260
     &        .OR. LRULE.EQ.5  .OR. LRULE EQ.11)                       HYST2270
           IF (CRACKD) THEN                                             HYST2280
              P2=.95*SIGN(PMAX,PM)                                      HYST2290
              D2=FD2(DMAX)                                              HYST2300
           ELSE                                                         HYST2310
              P2=PM                                                     HYST2320
              D2=DM                                                     HYST2330
           ENDIF                                                        HYST2340
           S=KMIN((D2-D),(P2-P),SI)                                     HYST2350
   20      I=I+1                                                        HYST2360
C                                                                       HYST2370
C... . ..... .. RELOADING INSIDE SMALL LOOPS                  RULE 11HYST2380
           IF (I.GT.0 ) THEN                                            HYST2390
              IF (.NOT (( FR(I).EQ.'7' .AND  DIR.EQ.1 )   OR.          HYST2400
     &                  ( FR(I) EQ 'L' .AND  DIR.EQ 3 ))  GO TO 20HYST2410
              IF (SR1.LE.0 .OR. SR2.LE.0  OR. SR3.LE.0  OR.            HYST2420
     &            SRM.LE.0 ) THEN                                     HYST2430
                 I=1                                                  HYST2440
                 GO TO 20                                             HYST2450
              ENDIF                                                   HYST2460
              IF (AP.LT.ABS(PC2) .AND. S.GT.SRM) THEN                 HYST2470
                 REVRSL=.FALSE.                                       HYST2480
                 I=1                                                  HYST2490
                 GO TO 20                                             HYST2500
              ENDIF                                                   HYST2510
              IF (ABS(PR(I)).LE  PC/4) THEN                           HYST2520
                 K=KMIN((D-DR(I)),(P-PR(I)),SI)                       HYST2530
                 A=PR(I)                                              HYST2540
                 RULE=11.1+I/1000.                                    HYST2550
              ELSE IF (ABS(PR(I)).LE.3*PC/4) THEN                     HYST2560
                 X2=D2*(P2-1.5*PC2)/SR3                               HYST2570
                 X1=X2*PC2/SR2                                        HYST2580
                 X=D*(PC2/2-P) /SR2                                   HYST2590
                 IF (AP.LT.PC/4 .AND. ((DIR.EQ.1 .AND  X.LT.X1).OR HYST2600
     &               (DIR.EQ.3 .AND. X.GT.X1))) THEN                  HYST2610
                    K=KMIN((X1-D),(PC2/2-P),SI)                       HYST2620
                    A=SIGN((PC/4),PM)                                 HYST2630
                    RULE=11.2+I/1000.                                 HYST2640
                 ELSE                                                 HYST2650
                    K=KMIN((DR(I)-D),(PR(I)-P),SI)                    HYST2660
                    A=PR(I)                                           HYST2670
                    RULE=11.3+I/1000.                                 HYST2680
                 ENDIF                                                HYST2690
              ELSE                                                    HYST2700
                 X2=DR(I)-(PR(I)-1.5*PC2)/SRM                         HYST2710
                 X1=X2*PC2/SR2                                        HYST2720
                 X=D*(PC2/2-P) /SR2                                   HYST2730
                 IF (AP.LT.PC/4 .AND. ((DIR.EQ.1  AND  X.LT.X1).OR.HYST2740
     &               (DIR.EQ.3 .AND. X.GT.X1))) THEN                  HYST2750
                    X=KMIN((X1-D),(PC2/2-P),SI)                       HYST2760
                    A=SIGN((PC/4),PM)                                 HYST2770
                    RULE=11.4+I/1000.                                 HYST2780
                 ELSE                                                 HYST2790
                    X=D-(1.5*PC2-P) /SR3                              HYST2800
                    IF (AP.LT. 3*PC/4  .AND.                          HYST2810
     &                  ((DIR.EQ.1 .AND. X LT.X2)                     HYST2820
     &                .OR.  (DIR.EQ.3 .AND. X.GT.X2))) THEN           HYST2830
                       X=KMIN((X2-D),(1.5*PC2-P),SI)                  HYST2840
                       A=SIGN((3*PC/4),PM)                            HYST2850
                       RULE=11 5+I/1000.                              HYST2860
                    ELSE                                              HYST2870
                       SR3=SRM                                        HYST2880
                       X=KMIN((DR(I)-D),(PR(I)-P),SI)                 HYST2890
                       A=PR(I)                                        HYST2900
                       RULE=11.6+I/1000                               HYST2910
                    ENDIF                                             HYST2920
                 ENDIF                                                HYST2930
              ENDIF                                                   HYST2940
C                                                                     HYST2950
C. . . .    ... INITIAL RELOADING BELOW PC/2  W/O REVERSAL      RULE 6 HYST2960
           ELSE IF (AF.LT.ABS(PC2)  AND. .NOT REVRSL ) THEN           HYST2970
              X=PS1(DMAX)                                             HYST2980
              A=PC2                                                   HYST2990
              RULE=6                                                  HYST3000
C                                                                     HYST3010
C.............. RELOADING TO P2                                RULE 7 HYST3020
           ELSE IF (AP.LT.ABS(P2)  AND                                HYST3030
     &          (.NOT.REVRSL .OR. AP.GE.PC*0.75)) THEN                HYST3040
                 X=S                                                  HYST3050
                 A=P2                                                 HYST3060
                 RULE=7                                               HYST3070
C                                                                     HYST3080
C  ......... . RELOADING BELOW PC WITH REVERSAL          RULES 8 & 9 HYST3090
           ELSE IF (AP.LT.PC*0.75 ) THEN                              HYST3100
                SR=FSR(DMAX)                                          HYST3110
                 S1=FS1(DMAX)                                         HYST3120
                 S2=FS2(DMAX)                                         HYST3130
                 S3=FS3(DMAX)                                         HYST3140
                 PA=SIGN(PMAX,PM)-SIGN(PC,PM)                         HYST3150
                 DA=SIGN(DMAX,PM)-SIGN(PC,PM)/S1                      HYST3160
                 PB=SIGN( (MIN(ABS(PA),(PC/2)) ),PM)                  HYST3170
                 DB=DA-(PA-PB)/S2                                     HYST3180
                 D0P=DB-(PB)/S3                                       HYST3190
                 D0=DM-PM*(DMG-DMH)/(PMG-PMH)                         HYST3200
                 IF (DIR.EQ.1) D0P=MAX(D0P,D0)                        HYST3210
                 IF (DIR.EQ 3) D0P=MIN(D0P,D0)                        HYST3220
              DC2=D0P-PC2/S1                                          HYST3230
              SRM=KMIN((DC2-D2),(PC2-P2),S1)                          HYST3240
              IF (S .GT.SRM) THEN                                     HYST3250
                 REVRSL=.FALSE.                                       HYST3260
                 GO TO 20                                             HYST3270
              ENDIF                                                   HYST3280
              SRP=KMIN((DC2-D),(PC2-P),S1)                            HYST3290
              SR1=MAX(SRP,MIN(SR,S))                                  HYST3300
              DC2P=D+(PC2-P)/SR1                                      HYST3310
              SR3=KMIN((D2-DC2P),(P2-PC2),SI)                         HYST3320
              SR2=2.00/( 1./SR1 + 1./SR3 )                            HYST3330
              IF (AP.LT.(PC/4) ) THEN                                 HYST3340
                 X=SR1                                                HYST3350
                 A=SIGN((PC/4),PM)                                    HYST3360
                 RULE=8                                               HYST3370
              ELSE                                                    HYST3380
                 X=SR2                                                HYST3390
                 A=SIGN(PC,PM)*0.75                                   HYST3400
                 RULE=9                                               HYST3410
              ENDIF                                                   HYST3420
C                                                                     HYST3430
C  . ..... .. . LOADING TOWARDS THE BACKBONE CURVE             RULE 10HYST3440
           ELSE                                                       HYST3450
              BACKB=.TRUE.                                            HYST3460
              IF (.NOT  CRACKD) GO TO 30                              HYST3470
              IR=0                                                    HYST3480
              ALPHAP=1.0                                              HYST3490
              IF (ABS(DM).EQ.ABS(DMAX))  ALPHAP=ALPHA                 HYST3500
              K=KMIN((ALPHAP*SIGN(DMAX,D2)-D2),(SIGN(PMAX,P2)-P2)     HYST3510
     &             ,SI)                                             HYST3520
              DO 50 J=2,NS                                            HYST3530
                IF ( PS(J).LE ABS(PMAX) ) GO TO 50                    HYST3540
                JC=J-1                                                HYST3550
                DD=DS(J)-DS(JC)                                       HYST3560
                DP=PS(J)-PS(JC)                                       HYST3570
                IF ((DP*DD) LE.0 ) GO TO 50                           HYST3580
                 CALL INTSCT(DX,PX,D,P,K,SIGN(DS(J),P),               HYST3590
     &                       SIGN(PS(J),P),(DP/DD) )                  HYST3600
                 IF (ABS(DX) GE.DS(JC)  AND  ABS(DX) LE DS(J) AND HYST3610
     &              ABS(PX).GT.ABS(PMAX).AND   PM*PX GT.0) THEN       HYST3620
                    IF (AP.GE.ABS(PX)) GO TO 30                       HYST3630
                    K=KMIN((DX-D),(PX-P),SI)                          HYST3640
                    A= PX                                             HYST3650
                    RULE=10                                           HYST3660
                    GO TO 100                                         HYST3670
                 ENDIF                                                HYST3680
   50             CONTINUE                                            HYST3690
                  GO TO 30                                            HYST3700
              ENDIF                                                   HYST3710
           ENDIF                                                      HYST3720
C                                                                     HYST3730
C                                                                     HYST3740
      ELSE                                                            HYST3750
C..... UNLOADING RULES  ....  . ...  ........  . ......... .. ....    HYST3760
C                                                                     HYST3770
C.... INITILIZE VALUES                                      RULES 2-5HYST3780
           BACKB= FALSE.                                              HYST3790
           S1=FS1(DMAX)                                               HYST3800
           S2=FS2(DMAX)                                               HYST3810
           S3=FS3(DMAX)                                               HYST3820
           D0=DM-PM*(DMG-DMH)/(PMG PMH)                               HYST3830
           FA=PM-SIGN(PC,PM)                                          HYST3840
           DA=DM-SIGN(PC,PM)/S1                                       HYST3850
           PB=SIGN( (MIN(ABS(FA),(PC/2)) ),PM)                        HYST3860
           S02=KMIN((DA-D0),PA,SI)                                    HYST3870
           DB=DA-(PA-PB)/MAX(S2,S02)                                  HYST3880
           S03=KMIN((DB-D0),PB,SI)                                    HYST3890
           SU=0.                                                      HYST3900
C                                                                     HYST3910
C.............. UNLOADING INSIDE SMALL POSITIVE LOOPS          RULE 5 HYST3920
         I=IR+1                                                       HYST3930
   70    I=I+1                                                        HYST3940
         IF (I.GT.0) THEN                                             HYST3950
           IF (.NOT.((DIR.EQ.2 .AND. FR(I).EQ.'L')    .OR.            HYST3960
     &              (DIR.EQ.4 .AND. FR(I).EQ.'7')) ) GO TO 70         HYST3970
C                                                                     HYST3980
           IF ((DIR.EQ.2 .AND.( (P.GT.PA .AND. PR(I).GT.PA) .OR      HYST3990
     &          (PA.GE.P .AND.   P.GT.PB .AND. PR(I).GT.PB) .OR      HYST4000
     &          (PB.GE.P .AND    PR(I).GT.0 )) .OR.                  HYST4010
     &         (DIR.EQ.4 .AND.( (P.LT.PA .AND. PR(I).LT.PA)  OR.     HYST4020
     &          (PA.LE.P .AND.   P.LT.PB .AND. PR(I).LT.PB) .OR.     HYST4030
     &          (PB.LE.P .AND.   PR(I).LT.0 )) )) THEN               HYST4040
             K=(PR(I)-P)/(DR(I)-D)                                    HYST4050
             IF (K.LE.0 .OR. K.GT.SI) GO TO 70                        HYST4060
             A=PR(I)                                                  HYST4070
             RULE=5+I/1000.                                           HYST4080
             GO TO 100                                                HYST4090
           ELSE                                                       HYST4100
             SU=KMIN((DR(I)-D),(PR(I)-P),SI)                          HYST4110
           ENDIF                                                      HYST4120
C                                                                     HYST4130
         ENDIF                                                        HYST4140
C                                                                     HYST4150
C.............. CALC DUCTILITY AND EXCURSION RATIO. .               HYST4160
         IF  (RULE.EQ.1 .OR. RULE.GT.5)  THEN                         HYST4170
           IF (DIR.EQ.2) THEN                                         HYST4180
             DLI= MAX( D , DL )                                       HYST4190
             CALL DNE(1,UPOS,EXCR, DY,DLI,ESE(2),PSE,PSEOLD,CSE)      HYST4200
           ELSE IF (DIR.EQ.4) THEN                                    HYST4210
             DLI= MIN( D , DL )                                       HYST4220
             CALL DNE(1,UNEG,EXCR,-DY,DLI,ESE(3),PSE,PSEOLD,CSE)      HYST4230
           ENDIF                                                      HYST4240
         ENDIF                                                        HYST4250
C                                                                     HYST4260
C.......  ..... UNLOADING ON THE TOP SEGMENT                 RULE 2 HYST4270
         IF(AP.GT.ABS(PA)) THEN                                       HYST4280
           K=MAX(S1,SU)                                               HYST4290
           A=PA                                                       HYST4300
           RULE=2                                                     HYST4310
C..... ..... UNLOADING ON THE MIDDLE SEGMENT '             RULE 3 HYST4320
         ELSE IF(AP.GT ABS(PB)) THEN                                  HYST4330
           K=MAX(S2,S02,SU)                                           HYST4340
           A=PB                                                       HYST4350
           RULE=3                                                     HYST4360
C..... ..... UNLOADING ON THE BOTTOM SEGMENT                RULE 4 HYST4370
         ELSE                                                         HYST4380
           K=MAX(S3,S03,SU)                                           HYST4390
           A=0.                                                       HYST4400
           RULE=4                                                     HYST4410
         ENDIF                                                        HYST4420
C                                                                     HYST4430
      ENDIF                                                           HYST4440
C                                                                     HYST4450
C ..... SAVE REVERSAL POINTS FOR SMALL LOOPS..  ..... .. .....       HYST4460
 100  IRULE=INT(RULE)                                                HYST4470
      REVRSL=IRULE.EQ.1.OR.IRULE.EQ.10.OR.LRULE.EQ.1.OR.LRULE.EQ 10  HYST4480
      IF      ((DIRL.EQ.1 .AND  DIR.EQ.2  AND. .NOT. REVRSL)         HYST4490
     &          .OR. (DIRL EQ 4 .AND. DIR.EQ.3) ) THEN               HYST4500
           IF (IR.LT.NI)  THEN                                        HYST4510
             IR=IR+1                                                  HYST4520
             PR(IR)=P                                                 HYST4530
             DR(IR)=D                                                 HYST4540
             FR(IR)='7'                                               HYST4550
           ENDIF                                                      HYST4560
      ELSE IF ((DIRL.EQ.3 .AND  DIR.EQ.4 .AND. .NOT. REVRSL)         HYST4570
     &          .OR. (DIRL EQ 2 .AND. DIR.EQ.1) ) THEN               HYST4580
           IF (IR.LT.NI)  THEN                                        HYST4590
             IR=IR+1                                                  HYST4600
             PR(IR)=P                                                 HYST4610
             DR(IR)=D                                                 HYST4620
             FR(IR)='L'                                               HYST4630
           ENDIF                                                      HYST4640
      ENDIF                                                           HYST4650
      IF ((1.05*SI) GT.K .AND. K.GT.0 ) RETURN                        HYST4660
C                                                                     HYST4670
      K=0                                                             HYST4680
      RULE=-RULE                                                      HYST4690
C                                                                     HYST4700
      RETURN                                                          HYST4710
      END                                                             HYST4720
C                                                                     HYST0010
C-----------------------------------------------------------------    HYST0020
C--- TAKEDA HYSTERESIS MODEL ..... HYST06 ..... .....  .....  ...    HYST0030
C-----------------------------------------------------------------    HYST0040
      SUBROUTINE HYST06(PI,DI,PS,DS,VI,VS,S,HRLN,A,B,IDRO,RF,         HYST0050
     & PC,DC,PY,DY,PU,OC,CY,YU,CPY,CSE,RNRL,UM1,UM2,UM3,UM4,          HYST0060
     & S1,X0 ,X0UM,X0Y,U0D,U0,X1UM,U1,U1D ,X2U0,U2,X3U1,U3,QY,        HYST0070
     & ACHNG,SE,ESE,PSE,PSEOLD,UPOS,UNEG,EXCR)                        HYST0080
C                                                                     HYST0090
      REAL UM(2,2),IDRO,UPOS(3),UNEG(3),EXCR(6),ESE(3)                HYST0100
C                                                                     HYST0110
C   TAKEDA HYSTERISIS MODEL - REPERANCE, S. OTANI, UILU-ENG-74-2029   HYST0120
C                  TAKEDA & SOZEN, J. ASCE ST. DIV. - DEC. 1970       HYST0130
C                                                                     HYST0140
C  PI       CURRENT LOAD                                             HYST0150
C  DI       CURRENT DISPL                                            HYST0160
C  PS       PREVIOUS LOAD                                            HYST0170
C  DS       PREVIOUS DISPL                                           HYST0180
C  VI       CURRENT VELOCITY                                         HYST0190
C  VL       PREVIOUS VELOCITY                                        HYST0200
C  S        CURRENT STIFFNESS                                        HYST0210
C  HRLN     CURRENT RULE NUMBER                                      HYST0220
C  A        UPPER LOAD LIMIT BEFORE RULE CHANGE                      HYST0230
C  B        LOWER LOAD LIMIT BEFORE RULE CHANGE                      HYST0240
C  IDRO  ??PREVIOUS DIRECTION                                        HYST0250
C  RF                                                                HYST0260
C  PC,DC    CRACKING LOAD AND DISPLACEMENT                           HYST0270
C  PY,DY    YEILD LOAD AND DISPLACEMENT                              HYST0280
C  PU       ULTIMATE LOAD AND DISPLACEMENT                           HYST0290
C  OC       ELASTIC STIFFNESS PD/DC                                  HYST0300
C  CY       POST CRACKING STIFFNESS (PY-PC)/(DY-DC)                  HYST0310
C  YU       POST YEILD STIFFNESS (PU-PY)/(DU DY)                     HYST0320
C  CPY      CRACKING', YEILD STIFFNESS (PY-PC)/(DY-DC)               HYST0330
C  CSE                                                               HYST0340
C  RNRL     NEXT RULE NUMBER                                         HYST0350
C  UM1      PEAK DISPLACEMENT IN THE POSITIVE DIRECTION              HYST0360
C  UM2      PEAK LOAD IN THE POSITIVE DIRECTION                      HYST0370
C  UM3      PEAK DISPLACEMENT IN THE NEGATIVE DIRECTION              HYST0380
C  UM4      PEAK LOAD IN THE NEGATIVE DIRECTION                      HYST0390
C  S1       UNLOADING STIFFNESS                                      HYST0400
C  ACHNG    LOAD WHERE RULE CHNGS IF LOADING CONTINUES IN CURR DIRECTION HYST0410
C  SE       ELASTIC UNLOADING CURVE                                  HYST0420
```

```
C  ESE     ELASTIC STRAIN ENERGY                                                HYST0430
C  PSE     PLASTIC STRAIN ENERGY                                                HYST0440
C  PSEOLD  PLASTIC STRAIN ENERGY AT LAST ZERO CROSSING                          HYST0450
C  UPOS    POSITIVE DUCTILITY INPO.                                             HYST0460
C  UNEG    NEGATIVE DUCTILITY INPO.                                             HYST0470
C  EXCR    EXCRUSION RATIO INPO.                                                HYST0480
C                                                                               HYST0490
C  JI      # FOR CURRENT DIRECTION FOR LOADING AND UNLOADING                    HYST0500
C          # FOR NEW DIRECTION FOR REVERSAL                                     HYST0510
C          # = 2 FOR POSITIVE LOAD ,   # = 1 FOR NEGATIVE LOAD                 HYST0520
C  IREVSL  SIGN OF LOAD IN NEW DIRECTION                                        HYST0530
C  ISGN    SIGN OF THE CURRENT LOAD FOR LOADING AND UNLOADING                   HYST0540
C          SIGN OF THE NEW LOAD FOR REVERSAL                                    HYST0550
C                                                                               HYST0560
C ...    RTN IF CURRENT STATE = PREVIOUS STATE                                  HYST0570
 8000 IF ((PI.EQ.PS .OR. DI.EQ.DS) .AND. RNRL.NE 0 ) RETURN                     HYST0580
C                                                                               HYST0590
C.....   GET DIRECTION AND RTN IF CURRENT STATE IS WITHIN LIMITS..              HYST0600
      CALL IDRECT (IDR,IDRVO,IDRVO,PI,PS,VI,VS)                                 HYST0610
      IF (A.GT.B .AND. PI.LE.A .AND. PI.GT.B .AND. IDR.EQ.IDRO) RETURN          HYST0620
      IF (A.LT.B .AND. PI.GT.A .AND. PI.LT.B .AND. IDR.EQ IDRO) RETURN          HYST0630
C                                                                               HYST0640
C. .  . CALC EXCURSION RATIO AT EVERY ZERO CROSSING...                          HYST0650
      IF (PI*PS.LE.0)CALL DNE(0,UPOS,EXCR, DY,DI,ESE,PSE,PSEOLD,CSE)            HYST0660
C                                                                               HYST0670
C..... INITILIZE DATA FOR FIRST CALL TO SUBROUTINE                              HYST0680
  1   IF (RNRL.EQ.0) THEN                                                       HYST0690
          RNRL=1                                                                HYST0700
          UM1=-DC                                                               HYST0710
          UM2=-PC                                                               HYST0720
          UM3= DC                                                               HYST0730
          UM4= PC                                                               HYST0740
      ENDIF                                                                     HYST0750
C                                                                               HYST0760
C..... INITILIZE DATA                                                           HYST0770
      NRL=RNRL                                                                  HYST0780
      UM(1,1)=UM1                                                               HYST0790
      UM(1,2)=UM2                                                               HYST0800
      UM(2,1)=UM3                                                               HYST0810
      UM(2,2)=UM4                                                               HYST0820
C                                                                               HYST0830
      IF (PI.GT.0) THEN                                                         HYST0840
  901     ISGN =1                                                               HYST0850
          JI=2                                                                  HYST0860
      ELSE IF (PI.LT.0) THEN                                                    HYST0870
  902     ISGN =-1                                                              HYST0880
          JI=1                                                                  HYST0890
      ELSE IF (DI.LT.DS) THEN                                                   HYST0900
  903     ISGN =-1                                                              HYST0910
          JI=1                                                                  HYST0920
      ELSE IF (DI.GT.DS) THEN                                                   HYST0930
  904     ISGN =1                                                               HYST0940
          JI=2                                                                  HYST0950
      ELSE                                                                      HYST0960
  905     ISGN =1                                                               HYST0970
          JI=1                                                                  HYST0980
      ENDIF                                                                     HYST0990
C                                                                               HYST1000
      IF (DI.LE.DS) THEN                                                        HYST1010
          IREVSL=-1                                                             HYST1020
      ELSE                                                                      HYST1030
          IREVSL= 1                                                             HYST1040
      ENDIF                                                                     HYST1050
C                                                                               HYST1060
      IF (IDR.LT.0) THEN                                                        HYST1070
       IF (ISGN.GT.0) CALL DNE(1,UPOS,EXCR, DY,DI,ESE(2),PSE,PSEOLD,CSE)HYST1080
       IF (ISGN.LT.0) CALL DNE(1,UNEG,EXCR,-DY,DI,ESE(3),PSE,PSEOLD,CSE)HYST1090
      ENDIF                                                                     HYST1100
C                                                                               HYST1110
      GO TO ( 100, 200, 300, 400, 500, 600, 700, 800, 900,1000,                 HYST1120
     1       1100,1200,1300,1400,1500,1600) NRL                                 HYST1130
C ----- TAKEDA RULE  1.0 -----    ELASTIC STAGE                                 HYST1140
  100 IF (IDR) 130,130,110                                                      HYST1150
  110 IF (ABS(PI).GE.PC) GO TO 200                                              HYST1160
      B=PI                                                                      HYST1170
      A=PC*ISGN                                                                 HYST1180
      S=DC                                                                      HYST1190
      SE=S                                                                      HYST1200
      NRL=1                                                                     HYST1210
      HRLN=1.11                                                                 HYST1220
      GO TO 2000                                                                HYST1230
  130 B=0.                                                                      HYST1240
      A=PC*IREVSL                                                               HYST1250
      IF (PI.NE.0) A=PC*SIGN(1.0,PI)                                            HYST1260
      S=DC                                                                      HYST1270
      SE=S                                                                      HYST1280
      NRL=1                                                                     HYST1290
      HRLN=1.2                                                                  HYST1300
      GO TO 2000                                                                HYST1310
C ----- TAKEDA RULE  2.0 -----  LOADING ON PRIMARY CURVE UP TO YEILD            HYST1320
  200 IF (IDR) 230,230,210                                                      HYST1330
  210 IF (ABS(PI) GE.PY) GO TO 300                                              HYST1340
      B=PI                                                                      HYST1350
      A=PY*ISGN                                                                 HYST1360
      S=CY                                                                      HYST1370
      S1=(PI+(PC*ISGN))/(DI+DC*ISGN)                                            HYST1380
      SE=S1                                                                     HYST1390
      NRL=2                                                                     HYST1400
      HRLN=2.11                                                                 HYST1410
      GO TO 2000                                                                HYST1420
  230 B=0.                                                                      HYST1430
      A=PI                                                                      HYST1440
      S1=(PI+(PC*ISGN))/(DI+DC*ISGN)                                            HYST1450
      S=S1                                                                      HYST1460
      SE=S1                                                                     HYST1470
      NRL=5                                                                     HYST1480
      HRLN=2.2                                                                  HYST1490
      GO TO 2000                                                                HYST1500
C ----- TAKEDA RULE  3.0 -----  LOADING ON PRIMARY CURVE AFTER YEILD            HYST1510
  300     BOTT=MAX(ABS(DI),ABS(UM1),ABS(UM3) )                                  HYST1520
          S1=CPY* SQRT(DY/BOTT)                                                 HYST1530
          SE=S1                                                                 HYST1540
      IF (IDR) 330,330,310                                                      HYST1550
  310 IF (ABS(PI) GE.PU) GO TO 320                                              HYST1560
      A=PU*ISGN                                                                 HYST1570
      B=PI                                                                      HYST1580
      S=YU                                                                      HYST1590
      NRL=3                                                                     HYST1600
      HRLN=3.11                                                                 HYST1610
      GO TO 2000                                                                HYST1620
  320 A=PU*ISGN                                                                 HYST1630
      B=PI                                                                      HYST1640
      S=0.00001                                                                 HYST1650
      NRL=3                                                                     HYST1660
      HRLN=3.12                                                                 HYST1670
      GO TO 2000                                                                HYST1680
  330 B=0.                                                                      HYST1690
      A=PI                                                                      HYST1700
      BOTT=MAX(ABS(DI),ABS(UM1),ABS(UM3) )                                      HYST1710
      S1=CPY* SQRT(DY/BOTT)                                                     HYST1720
      S=S1                                                                      HYST1730
      NRL=4                                                                     HYST1740
      HRLN=3.2                                                                  HYST1750
      GO TO 2000                                                                HYST1760
C ----- TAKEDA RULE  4.0 -----  UNLOADING FROM POINT UM  ON THE PRIMARYHYST1770
C                              CURVE AFTER YEILDING                             HYST1780
  400 IF (IDR) 430,440,410                                                      HYST1790
  410 IF (ABS(PI).GE.ABS(UM(JI,2))) GO TO 420                                   HYST1800
      A=UM(JI,2)                                                                HYST1810
      B=PI                                                                      HYST1820
      S=S1                                                                      HYST1830
      NRL=4                                                                     HYST1840
      HRLN=4.11                                                                 HYST1850
      GO TO 2000                                                                HYST1860
  420 A=PU*ISGN                                                                 HYST1870
      B=PI                                                                      HYST1880
      S=YU                                                                      HYST1890
      NRL=3                                                                     HYST1900
      HRLN=4.12                                                                 HYST1910
      GO TO 2000                                                                HYST1920
  430 A=PI                                                                      HYST1930
      B=0                                                                       HYST1940
      S=S1                                                                      HYST1950
      NRL=4                                                                     HYST1960
      HRLN=4.2                                                                  HYST1970
      GO TO 2000                                                                HYST1980
  440 IF (DI.LE.DS) IDRVO=1                                                     HYST1990
      IF (DI.GT.DS) IDRVO=2                                                     HYST2000
      IF (ABS(UM(IDRVO,1)).GT.DC) GO TO 450                                     HYST2010
      A=PC*IREVSL                                                               HYST2020
      B=0.                                                                      HYST2030
      S=S1                                                                      HYST2040
      NRL=15                                                                    HYST2050
      HRLN=4.31                                                                 HYST2060
      GO TO 2000                                                                HYST2070
  450 X0=DI                                                                     HYST2080
      X0UM=(0.-UM(IDRVO,2))/(X0-UM(IDRVO,1))                                    HYST2090
      X0Y =(0.-PY*ISGN)/(X0-DY*ISGN)                                            HYST2100
      IF (PY.GT.ABS(UM(IDRVO,2)) .AND. X0Y.GT.X0UM) GO TO 460                   HYST2110
      A=UM(IDRVO,2)                                                             HYST2120
      B=0.                                                                      HYST2130
      S2=X0UM                                                                   HYST2140
      S=S2                                                                      HYST2150
      NRL=6                                                                     HYST2160
      HRLN=4.32                                                                 HYST2170
      GO TO 2000                                                                HYST2180
  460 UM(JI,1)=DY*ISGN                                                          HYST2190
      UM(JI,2)=PY*ISGN                                                          HYST2200
      A=UM(JI,2)                                                                HYST2210
      B=0.                                                                      HYST2220
      S2=X0Y                                                                    HYST2230
      S=S2                                                                      HYST2240
      NRL=6                                                                     HYST2250
      HRLN=4.33                                                                 HYST2260
      GO TO 2000                                                                HYST2270
C ----- TAKEDA RULE  5.0 -----   UNLOADING FROM POINT UM ON THE PRIMARYHYST2280
C                                CURVE BEFORE YEILDING                          HYST2290
  500 IF (IDR) 530,540,510                                                      HYST2300
  510 IF (ABS(PI).GE.ABS(UM(JI,2))) GO TO 200                                   HYST2310
      A=UM(JI,2)                                                                HYST2320
      B=PI                                                                      HYST2330
      S=S1                                                                      HYST2340
      NRL=5                                                                     HYST2350
      HRLN=5.11                                                                 HYST2360
      GO TO 2000                                                                HYST2370
  530 A=PI                                                                      HYST2380
      B=0                                                                       HYST2390
      S=S1                                                                      HYST2400
      NRL=5                                                                     HYST2410
      HRLN=5.2                                                                  HYST2420
      GO TO 2000                                                                HYST2430
  540 IF (DI.LE.DS) IDRV=2                                                      HYST2440
      IF (DI.GT.DS) IDRV=1                                                      HYST2450
      IF (ABS(UM(IDRV,1)).GT.DC) GO TO 450                                      HYST2460
      A=PC*IREVSL                                                               HYST2470
      B=0.                                                                      HYST2480
      S=S1                                                                      HYST2490
      NRL=14                                                                    HYST2500
      HRLN=5.31                                                                 HYST2510
      GO TO 2000                                                                HYST2520
C ----- TAKEDA RULE  6.0 -----   LOADING TOWARDS POINT UM ON THE PRIMARHYST2530
C                                CURVE.                                         HYST2540
  600 IF (IDR) 630,630,610                                                      HYST2550
  610 IF (ABS(PI).GE.ABS(UM(JI,2))) GO TO 210                                   HYST2560
      A=UM(JI,2)                                                                HYST2570
      B=PI                                                                      HYST2580
      X0UM=(PI-UM(JI,2))/(DI-UM(JI,1))                                          HYST2590
      S=X0UM                                                                    HYST2600
      NRL=6                                                                     HYST2610
      HRLN=6.11                                                                 HYST2620
      GO TO 2000                                                                HYST2630
  630 U0D=DI                                                                    HYST2640
      U0=PI                                                                     HYST2650
      A=PI                                                                      HYST2660
      B=0.                                                                      HYST2670
      S=S1                                                                      HYST2680
      NRL=7                                                                     HYST2690
      HRLN=6.20                                                                 HYST2700
      GO TO 2000                                                                HYST2710
C ----- TAKEDA RULE  7.0 -----  UNLOADING FROM POINT UM AFTER RULE 6            HYST2720
  700 IF (IDR) 730,740,710                                                      HYST2730
  710 IF (ABS(PI).GE.ABS(U0)) GO TO 720                                         HYST2740
      A=U0                                                                      HYST2750
      B=PI                                                                      HYST2760
      S=S1                                                                      HYST2770
      NRL=7                                                                     HYST2780
      HRLN=7.11                                                                 HYST2790
      GO TO 2000                                                                HYST2800
  720 A=UM(JI,2)                                                                HYST2810
      B=PI                                                                      HYST2820
      X0UM=(0.-UM(JI,2))/(X0-UM(JI,1))                                          HYST2830
      S=X0UM                                                                    HYST2840
      NRL=6                                                                     HYST2850
      HRLN=7.12                                                                 HYST2860
      GO TO 2000                                                                HYST2870
  730 A=PI                                                                      HYST2880
      B=0.                                                                      HYST2890
      S=S1                                                                      HYST2900
      NRL=7                                                                     HYST2910
      HRLN=7.2                                                                  HYST2920
      GO TO 2000                                                                HYST2930
  740 A=UM(JI,2)                                                                HYST2940
      B=0                                                                       HYST2950
      X1UM=(PI-UM(JI,2))/(DI-UM(JI,1))                                          HYST2960
      S=X1UM                                                                    HYST2970
      NRL=8                                                                     HYST2980
      HRLN=7.3                                                                  HYST2990
      GO TO 2000                                                                HYST3000
C ----- TAKEDA RULE  8.0 -----   LOADING TOWARDS POINT UM ON THE PRIMARHYST3010
C                                CURVE                                          HYST3020
  800 IF (IDR) 830,830,810                                                      HYST3030
  810 IF (ABS(PI).GE.ABS(UM(JI,2))) GO TO 210                                   HYST3040
      A=UM(JI,2)                                                                HYST3050
      B=PI                                                                      HYST3060
      X1UM=(PI-UM(JI,2))/(DI-UM(JI,1))                                          HYST3070
      S=X1UM                                                                    HYST3080
      NRL=8                                                                     HYST3090
      HRLN=8.11                                                                 HYST3100
      GO TO 2000                                                                HYST3110
  830 U1=PI                                                                     HYST3120
      U1D=DI                                                                    HYST3130
      A=PI                                                                      HYST3140
      B=0                                                                       HYST3150
      S=S1                                                                      HYST3160
      NRL=9                                                                     HYST3170
      HRLN=8.2                                                                  HYST3180
      GO TO 2000                                                                HYST3190
C ----- TAKEDA RULE  9.0 -----   UNLOADING FROM POINT U1 AFTER RULE 8           HYST3200
  900 IF (IDR) 930,940,910                                                      HYST3210
  910 IF (ABS(PI) GE ABS(U1       )) GO TO 810                                  HYST3220
      A=U1                                                                      HYST3230
      B=PI                                                                      HYST3240
      S=S1                                                                      HYST3250
      NRL=9                                                                     HYST3260
      HRLN=9.11                                                                 HYST3270
      GO TO 2000                                                                HYST3280
  930 A=PI                                                                      HYST3290
      B=0                                                                       HYST3300
      S=S1                                                                      HYST3310
      NRL=9                                                                     HYST3320
      HRLN=9.2                                                                  HYST3330
      GO TO 2000                                                                HYST3340
```

```
  940 A=U0                                                                  HYST3350
      B=0.                                                                  HYST3360
      X2U0=(PI-U0)/(DI-U0D)                                                 HYST3370
      S=X2U0                                                                HYST3380
      NRL=10                                                                HYST3390
      HRLN=9.3                                                              HYST3400
      GO TO 2000                                                            HYST3410
C  ----- TAKEDA RULE 10.0 -----   LOADING TOWARDS POINT U0                  HYST3420
 1000 IF (IDR) 1030,1030,1010                                               HYST3430
 1010 IF (ABS(PI).GE.ABS(U0       )) GO TO 610                             HYST3440
      A=U0                                                                  HYST3450
      B=PI                                                                  HYST3460
      X2U0=(PI-U0)/(DI-U0D)                                                 HYST3470
      S=X2U0                                                                HYST3480
      NRL=10                                                                HYST3490
      HRLN=10.11                                                            HYST3500
      GO TO 2000                                                            HYST3510
 1030 U2=PI                                                                 HYST3520
      A=PI                                                                  HYST3530
      B=0.                                                                  HYST3540
      S=S1                                                                  HYST3550
      NRL=11                                                                HYST3560
      HRLN=10.2                                                             HYST3570
      GO TO 2000                                                            HYST3580
C  ----- TAKEDA RULE 11.0 -----   UNLOADING FROM POINT U2 AFTER RULE 10      HYST3590
 1100 IF (IDR) 1130,1140,1110                                               HYST3600
 1110 IF (ABS(PI).GE.ABS(U2       )) GO TO 1010                            HYST3610
      A=U2                                                                  HYST3620
      B=PI                                                                  HYST3630
      S=S1                                                                  HYST3640
      NRL=11                                                                HYST3650
      HRLN=11.11                                                            HYST3660
      GO TO 2000                                                            HYST3670
 1130 A=PI                                                                  HYST3680
      B=0.                                                                  HYST3690
      S=S1                                                                  HYST3700
      NRL=11                                                                HYST3710
      HRLN=11.2                                                             HYST3720
      GO TO 2000                                                            HYST3730
 1140 A=U1                                                                  HYST3740
      B=0                                                                   HYST3750
      X3U1=(PI-U1)/(DI-U1D)                                                 HYST3760
      S=X3U1                                                                HYST3770
      NRL=12                                                                HYST3780
      HRLN=11.3                                                             HYST3790
      GO TO 2000                                                            HYST3800
C  ----- TAKEDA RULE 12.0 -----   LOADING TOWARDS POINT U1                  HYST3810
 1200 IF (IDR) 1230,1230,1210                                               HYST3820
 1210 IF (ABS(PI).GE.ABS(U1       )) GO TO  810                            HYST3830
      A=U1                                                                  HYST3840
      B=PI                                                                  HYST3850
      X3U1=(PI-U1)/(DI-U1D)                                                 HYST3860
      S=X3U1                                                                HYST3870
      NRL=12                                                                HYST3880
      HRLN=12.11                                                            HYST3890
      GO TO 2000                                                            HYST3900
 1230 U3=PI                                                                 HYST3910
      A=PI                                                                  HYST3920
      B=0.                                                                  HYST3930
      S=S1                                                                  HYST3940
      NRL=13                                                                HYST3950
      HRLN=12.2                                                             HYST3960
      GO TO 2000                                                            HYST3970
C  ----- TAKEDA RULE 13.0 -----  UNLOADING FROM POINT U3 AFTER RULE 12       HYST3980
 1300 IF (IDR) 1330, 940,1310                                               HYST3990
 1310 IF (ABS(PI).GE.ABS(U3       )) GO TO 1210                            HYST4000
      A=U3                                                                  HYST4010
      B=PI                                                                  HYST4020
      S=S1                                                                  HYST4030
      NRL=13                                                                HYST4040
      HRLN=13.11                                                            HYST4050
      GO TO 2000                                                            HYST4060
 1330 A=PI                                                                  HYST4070
      B=0                                                                   HYST4080
      S=S1                                                                  HYST4090
      NRL=13                                                                HYST4100
      HRLN=13.2                                                             HYST4110
      GO TO 2000                                                            HYST4120
C  ----- TAKEDA RULE 14.0 -----  LOADING IN THE UNCRACKED DIRECTION AFTER HYST4130
C                                 CRACKING IN THE OTHER DIRECTION.         HYST4140
 1400 IF (IDR) 1430,1440,1410                                               HYST4150
 1410 IF (ABS(PI).GE.     PC      ) GO TO  210                             HYST4160
      A=PC*ISGN                                                             HYST4170
      B=0.                                                                  HYST4180
      S=S1                                                                  HYST4190
      NRL=14                                                                HYST4200
      HRLN=14.11                                                            HYST4210
      GO TO 2000                                                            HYST4220
 1430 A=PI                                                                  HYST4230
      B=0.                                                                  HYST4240
      S=S1                                                                  HYST4250
      NRL=14                                                                HYST4260
      HRLN=14.2                                                             HYST4270
      GO TO 2000                                                            HYST4280
 1440 A=UM(JI,2)                                                            HYST4290
      B=0                                                                   HYST4300
      S=S1                                                                  HYST4310
      NRL=5                                                                 HYST4320
      HRLN=14.3                                                             HYST4330
      GO TO 2000                                                            HYST4340
C  ----- TAKEDA RULE 15.0 -----  LOADING IN THE UNCRACKED DIRECTION AFTER HYST4350
C                                 YEILDING IN THE OTHER DIRECTION.         HYST4360
 1500 IF (IDR) 1530,1540,1510                                               HYST4370
 1510 IF (ABS(PI).GE.     PC      ) GO TO 1520                             HYST4380
      A=PC*ISGN                                                             HYST4390
      B=PI                                                                  HYST4400
      S=S1                                                                  HYST4410
      NRL=15                                                                HYST4420
      HRLN=15.11                                                            HYST4430
      GO TO 2000                                                            HYST4440
 1520 PQ=PC*ISGN                                                            HYST4450
      DQ=(PQ-PI)/S1+DI                                                      HYST4460
      QY=(PY*ISGN-PQ)/(DY*ISGN-DQ)                                          HYST4470
      A=PY*ISGN                                                             HYST4480
      B=PQ                                                                  HYST4490
      S=QY                                                                  HYST4500
      NRL=16                                                                HYST4510
      HRLN=15.12                                                            HYST4520
      GO TO 2000                                                            HYST4530
 1530 A=PI                                                                  HYST4540
      B=0.                                                                  HYST4550
      S=S1                                                                  HYST4560
      NRL=15                                                                HYST4570
      HRLN=15.2                                                             HYST4580
      GO TO 2000                                                            HYST4590
 1540 IF (ABS(PI)  GE  ABS(UM(JI,2))) GO TO 1550                            HYST4600
      A=UM(JI,2)                                                            HYST4610
      B=0                                                                   HYST4620
      S=S1                                                                  HYST4630
      NRL=4                                                                 HYST4640
      HRLN=15.31                                                            HYST4650
      GO TO 2000                                                            HYST4660
 1550 A=PU*IREVSL                                                           HYST4670
      B=0                                                                   HYST4680
      S=YU                                                                  HYST4690
      NRL=3                                                                 HYST4700
      HRLN=15.32                                                            HYST4710
      GO TO 2000                                                            HYST4720
C  ----- TAKEDA RULE 16.0 -----   LOADING TOWARDS THE YEILD POINT AFTER HYST4730
C                                  RULE 15                                 HYST4740
 1600 IF (IDR) 1630,1630,1610                                               HYST4750
 1610 IF (ABS(PI) GE      PY      ) GO TO 300                              HYST4760
      A=PY*ISGN                                                             HYST4770
      B=PI                                                                  HYST4780
      S=QY                                                                  HYST4790
      NRL=16                                                                HYST4800
      HRLN=16.11                                                            HYST4810
      GO TO 2000                                                            HYST4820
 1630 UM(JI,1)=DY                                                           HYST4830
      UM(JI,2)=PY                                                           HYST4840
      U0=PI                                                                 HYST4850
      U0D=DI                                                                HYST4860
      A=PI                                                                  HYST4870
      B=0.                                                                  HYST4880
      S=S1                                                                  HYST4890
      S2=QY                                                                 HYST4900
      X0=DI-PI/S2                                                           HYST4910
      NRL= 7                                                                HYST4920
      HRLN=16.2                                                             HYST4930
      GO TO 2000                                                            HYST4940
C                                                                           HYST4950
C ------> CALCULATE MAXIMUN PREVIOUS DEFORMATION                           HYST4960
 2000    UM1=MIN(UM(1,1),DI)                                                HYST4970
         UM2=MIN(UM(1,2),PI)                                                HYST4980
         UM3=MAX(UM(2,1),DI)                                                HYST4990
         UM4=MAX(UM(2,2),PI)                                                HYST5000
         RNRL=NRL                                                           HYST5010
         IDRO=IDR                                                           HYST5020
         RF=0.                                                              HYST5030
         IF (IDR.GE.0) ACHNG=A                                              HYST5040
         IF (IDR.LT.0) ACHNG=B                                              HYST5050
      RETURN                                                                HYST5060
      END                                                                   HYST5070
C                                                                           HYST0010
C********** HYST07 ***********************************************************HYST0020
C                                                                           HYST0030
      SUBROUTINE HYST07(PC,DC,PL,DL,K,RULE,PSEOLD,A,B,VI,VL,S,PY,KIE,       HYST0040
     &                  DY,CSE,ESE,PSE,RKEY,UPOS,UNEG,EXCR)                 HYST0050
      REAL    K,KIE                                                         HYST0060
      DIMENSION UPOS(3),UNEG(3),EXCR(6),ESE(3)                              HYST0070
C                                                                           HYST0080
C     SU_BROU_TINE ELSPLS - ELASTO-PLASTIC BILINEAR HYSTERESIS MODEL        HYST0090
C                                                                           HYST0100
C     PC        - CURRENT LOAD                                            HYST0110
C     PEQ       - MODIFIED CURRENT LOAD                                   HYST0120
C     DC        - CURRENT DISPL.                                          HYST0130
C     PL        - LAST LOAD                                               HYST0140
C     DL        - LAST DISPL.                                             HYST0150
C     PY        - YEILD LOAD                                              HYST0160
C     PYEQ      - MODIFIED YEILD LOAD                                     HYST0170
C     S         - ELASTIC STIFFNESS                                       HYST0180
C     K         - CURRENT STIFFNESS (OUTPUT)                              HYST0190
C     RULE      - CURRENT RULE NUMBER, USED TO TRACE MODEL (OUTPUT)       HYST0200
C     RNRL      - NEXT RULE NUMBER (OUTPUT)                               HYST0210
C     VI        - NOT USED                                                HYST0220
C     VL        - NOT USED                                                HYST0230
C     KIE       - INELASTIC STIFFNESS                                     HYST0240
C                                                                           HYST0250
      KEY=RKEY                                                              HYST0260
      PC=K*(DC-DL)+PL                                                       HYST0270
C                                                                           HYST0280
C.....  CALC EXCURSION RATIO AT EVERY ZERO CROSSING ..                      HYST0290
      IF (PC*PL.LE.0) CALL DNE(0,UPOS,EXCR, DY,DL,ESE,PSE,PSEOLD,CSE)       HYST0300
C                                                                           HYST0310
C                                                                           HYST0320
C-----------  ELASTIC REGION                                                HYST0330
      IF (RULE.LE.1 .AND. ABS(PC).LT.PY)  THEN                              HYST0340
         K=S                                                                HYST0350
         RULE=1.                                                            HYST0360
         A= PY                                                              HYST0370
         B=-PY                                                              HYST0380
         PC=S*DC                                                            HYST0390
         CALL STRENG (ESE,PSE,PSEOLD,PC,PL,DC,DL,S )                        HYST0400
         KEY=0                                                              HYST0410
         RKEY=KEY                                                           HYST0420
         GO TO 1000                                                         HYST0430
      ENDIF                                                                 HYST0440
C                                                                           HYST0450
C-----------  INELASTIC REGION                                              HYST0460
      DIR=DC*DL                                                             HYST0470
      IF (DIR.EQ.0) RETURN                                                  HYST0480
      PEQ =PC-KIE*DC                                                        HYST0490
      PEQL=PL-KIE*DL                                                        HYST0500
      PYEQ=PY*(1-KIE/S)                                                     HYST0510
      DVEL=VI*VL                                                            HYST0520
C                                                                           HYST0530
C------  LOAD AT OR ABOVE POSITIVE YEILD                                    HYST0540
      IF (RULE.EQ.1) THEN                                                   HYST0550
         IF (DIR.GT.0) THEN                                                 HYST0560
C           ...  POSITIVE YEILD ...                                         HYST0570
            CALL STRENG (ESE,PSE,PSEOLD, PY,PL, DY,DL,S )                   HYST0580
            PL=PY                                                           HYST0590
            DL=DY                                                           HYST0600
            GO TO 210                                                       HYST0610
         ELSE                                                               HYST0620
C           ... NEGATIVE YEILD ...                                          HYST0630
            CALL STRENG (ESE,PSE,PSEOLD,-PY,PL,-DY,DL,S )                   HYST0640
            PL=-PY                                                          HYST0650
            DL=-DY                                                          HYST0660
            GO TO 220                                                       HYST0670
         ENDIF                                                              HYST0680
      ELSE                                                                  HYST0690
         IF (RULE.EQ.2.10) GO TO 210                                        HYST0700
         IF (RULE.EQ.2.20) GO TO 220                                        HYST0710
         IF (RULE.EQ.2.30) GO TO 230                                        HYST0720
      ENDIF                                                                 HYST0730
      WRITE(108,*) 'INVALID RULE IN HYST07-BILINEAR HYSTERESIS MODEL'       HYST0740
C                                                                           HYST0750
C-----  LOAD AT OR ABOVE POSITIVE YEILD, ON BACKBONE CURVE                  HYST0760
  210 IF (DIR.LE.0) GO TO 230                                               HYST0770
      K=KIE                                                                 HYST0780
      RULE=2.10                                                             HYST0790
      PC=PYEQ+DC*KIE                                                        HYST0800
      B= PC                                                                 HYST0810
      A= PC*10                                                              HYST0820
      CALL STRENG (ESE,PSE,PSEOLD,PC,PL,DC,DL,S )                           HYST0830
      KEY=1                                                                 HYST0840
         RKEY=KEY                                                           HYST0850
      IF (DVEL.LT.0 ) THEN                                                  HYST0860
            K=S                                                             HYST0870
            B= PC                                                           HYST0880
            A= PC                                                           HYST0890
      ENDIF                                                                 HYST0900
      GO TO 1000                                                            HYST0910
C                                                                           HYST0920
C-----  LOAD AT OR BELOW NEGATIVE YEILD, ON BACKBONE CURVE                  HYST0930
  220 IF (DIR.GE.0) GO TO 230                                               HYST0940
      K=KIE                                                                 HYST0950
      RULE=2.20                                                             HYST0960
      PC=-PYEQ+DC*KIE                                                       HYST0970
      B= PC                                                                 HYST0980
      A= PC*10                                                              HYST0990
      CALL STRENG (ESE,PSE,PSEOLD,PC,PL,DC,DL,S )                           HYST1000
      KEY=2                                                                 HYST1010
         RKEY=KEY                                                           HYST1020
      IF (DVEL.LT.0 ) THEN                                                  HYST1030
            K=S                                                             HYST1040
            B= PC                                                           HYST1050
            A= PC                                                           HYST1060
      ENDIF                                                                 HYST1070
      GO TO 1000                                                            HYST1080
C                                                                           HYST1090
C------  LOAD IN LINEAR RANGE BETWEEN BACKBONE CURVES                       HYST1100
  230 K=S                                                                   HYST1110
      RULE=2.3                                                              HYST1120
      PC=S*(DC-DL) + PL                                                     HYST1130
      A= PC + PYEQ - PEQ                                                    HYST1140
      B= PC - PYEQ - PEQ                                                    HYST1150
      IF (PC.GT.A) THEN                                                     HYST1160
         DA=DC-(A-PC)/S                                                     HYST1170
         CALL STRENG (ESE,PSE,PSEOLD, A,PL,DA,DL,S )                        HYST1180
         DL=DA                                                              HYST1190
```

```
        FL=A                                                                    HYST1200
        GO TO 210                                                               HYST1210
      ELSE IF (FC LT.B) THEN                                                    HYST1220
        DB=DC*(B-FC)/S                                                          HYST1230
        CALL STRENG (ESE,PSE,PSEOLD, B,FL,DB,DL,S )                             HYST1240
        DL=DB                                                                   HYST1250
        FL=B                                                                    HYST1260
        GO TO 220                                                               HYST1270
      ENDIF                                                                     HYST1280
C                                                                               HYST1290
C.. ..... CALC D & E'S BASED ON PEAK STRAIN ENERGY                              HYST1300
      IF (KEY EQ.1) CALL DNE(1,UPOS,EXCR, DY,DL,ESE(2),PSE,PSEOLD,CSE)         HYST1310
      IF (KEY.EQ.2) CALL DNE(1,UNEG,EXCR,-DY,DL,ESE(3),PSE,PSEOLD,CSE)         HYST1320
C                                                                               HYST1330
C . .... CALC STRAIN ENERGY FOR THIS STEP..                                     HYST1340
      CALL STRENG (ESE,PSE,PSEOLD,FC,FL,DC,DL,S )                               HYST1350
C                                                                               HYST1360
C........ CALC PLACTIC STRAIN ENERGY AT ZERO CROSSING ...                       HYST1370
C                                                                               HYST1380
      KEY=3                                                                     HYST1390
      RKEY=KEY                                                                  HYST1400
C                                                                               HYST1410
 1000 RETURN                                                                    HYST1420
      END                                                                       HYST1430
C                                                                               HYST0010
C     GOEL TRUSS HYSTERESIS MODEL FOR ANGLE AND BOX SECTION                     HYST0020
C                                                                               HYST0030
      SUBROUTINE HYST08(P,PLA,DE,DELA,EL,SLK,EM,AREA,RADG,YS,PY,SHAPE,          HYST0040
     &       CC,CYCLE,RULE,STIF,IRV,IDUC,HA,VA,HC,VC,REN,DEMAX,REGIN,          HYST0050
     &       PREGIN,PBOT,DBOT,PTOP,DTOP,EKI,DUC,EXCR,DELY,ESE,                  HYST0060
     &       PSE,PSEOLD,CSE)                                                    HYST0070
      DIMENSION DUC(3), EXCR(6), ESE(3)                                         HYST0080
      LOGICAL REVRSC                                                            HYST0090
      REAL*4 IRV, IDUC                                                          HYST0100
C                                                                               HYST0110
      IF( IRV .EQ. 0.) REVRSC=.FALSE.                                           HYST0120
      IF( IRV .EQ. 1.) REVRSC= TRUE.                                            HYST0130
      PI=3.1415926                                                              HYST0140
      ES=AREA*EM/EL                                                             HYST0150
      DELY=PY/ES                                                                HYST0160
      DEL=DE/DELY                                                               HYST0170
      PL=P/PY                                                                   HYST0180
      ESR=SLK*EL/RADG                                                           HYST0190
      SLOP7=STIF/ES                                                             HYST0200
      DA=ABS(DE)                                                                HYST0210
C                                                                               HYST0220
      DP=P-PLA                                                                  HYST0230
      DDE=DE-DELA                                                               HYST0240
      DDEL=DDE/DELY                                                             HYST0250
C ---- CALCULATE EXCURSION RATE AT EVERY ZERO CROSSING ********************** HYST0260
C                                                                               HYST0270
      IF (P*PLA.LE 0.) CALL DNE(0,DUC,EXCR,DELY,DA,ESE,PSE,PSEOLD,CSE)         HYST0280
C                                                                               HYST0290
C*************************************************************************     HYST0300
C   DUC     ---- ELEMENT DUCTILITY RATE                                        HYST0310
C   EXCR    ---- ELEMENT EXCURSION RATE                                        HYST0320
C   DELY    ---- ELEMENT YIELDING DISPLACEMENT                                 HYST0330
C   ESE     ---- ELEMENT ELASTIC STRAIN ENERGY                                 HYST0340
C   PSE     ---- ELEMENT PLASTIC STRAIN ENERGY                                 HYST0350
C   PSEOLD -- - ELEMENT PLASTIC STRAIN ENERGY AT LAST ZERO CROSSING            HYST0360
C   CSE     ---- ELEMENT CONSTANT STRAIN ENERGY                                HYST0370
C                                                                               HYST0380
C   DP      ---- ELEMENT AXIAL LOAD INCREMENT (KIPS)                           HYST0390
C   ESR     ---- EFFECTIVE SLENDER RATIO  (KL/R)                               HYST0400
C   YS      ---- YIELDING STRESS                                               HYST0410
C   AREA    ---- CROSS SECTION AREA (A)                                        HYST0420
C   EM      ---- ELASTIC MODULUS  (E)                                          HYST0430
C   DP      ---- ELEMENT AXIAL LOAD INCREMENT (KIPS)                           HYST0440
C   P       ---- CURRENT AXIAL LOAD (KIPS)                                     HYST0450
C   PL      ---- P - PY RATIO (P/PY)                                           HYST0460
C   PLA     ---- LAST AXIAL LOAD                                               HYST0470
C   DDEL    ---- DISPLACEMENT INCREMENT  (IN)                                  HYST0480
C   DE      ---- CURRENT AXIAL DISPLACEMENT  (IN)                              HYST0490
C   DEMAX   ---- MAXIMUM AXIAL DISPLACEMENT  (IN)                              HYST0500
C   DA       --- CURRENT ABSOLUTE AXIAL DISPLACEMENT  (IN)                     HYST0510
C   DELA    ---- LAST AXIAL DISPLACEMENT                                       HYST0520
C   DDE     ---- ELEMENT DISPLACEMENT INCREMENT                                HYST0530
C   DEL     ---- DEL-DE/DELY                                                   HYST0540
C   DELY    ---- YIELDING DISPLACEMENT                                         HYST0550
C   V       ---- CURRENT VELOCITY                                              HYST0560
C   VLA     ---- LAST VELOCITY                                                 HYST0570
C   ES      ---- ELEMENT ELASTIC STIFFNESS                                     HYST0580
C   EKI     ---- UNLOADING EQUIVALENT STIFFNESS FOR CALCULATING ESE            HYST0590
C   PY      ---- TENSION YIELDING LOAD                                         HYST0600
C   EL      ---- ELEMENT LENGTH  (IN)                                          HYST0610
C   CC      ---- C SUB. C CONSTANT                                             HYST0620
C   PMAX    ---- BUCKING LOAD (KIPS)                                          HYST0630
C   DELMAX ---- BUCKING DISPLACEMENT (IN)                                      HYST0640
C   HA,HB,..,HC1,HD1..,HP ---  P/PY - DE/DELY PLOT'S DE/DELY COORDINATE        HYST0650
C   VA,VB,. ,VC1,VD1 .,VP ---- P/PY - DE/DELY PLOT'S P/PY COORDINATE           HYST0660
C   RSN     ---- RESIDUAL STRAIN                                               HYST0670
C   RULE    ---- RULE NUMBER                                                   HYST0680
C   CYCLE    ---- NO. OF CYCLE                                                 HYST0690
C   STIF    ---- ELEMENT STIFFNESS                                             HYST0700
C   REVRSC ---- LOGICAL INDEX FOR REVERSING CURVE OCCURRED                     HYST0710
C   IDUC    ---- 1. HC LESS THAN 12,  0 HC=12                                  HYST0720
C   REGIN   ---- REGIN 1. DE >  HA                                             HYST0730
C                     2. DE = (HH, HA)                                          HYST0740
C                     3. DE = (HB, HH)                                          HYST0750
C                     4  DE = (HC, HB)                                          HYST0760
C   PREGIN ---- PREVIOUS STEP REGIN ID NUMBER                                  HYST0770
C   SHAPE  --   1. FOR BOX OR ANGLE SECTION                                    HYST0780
C               2. FOR I SHAPE SECTION                                          HYST0790
C               3. FOR BOX SECTION CONSIDERING LOCAL BUCKLING                   HYST0800
C                  NOTE. THIS IS ONLY FOR KL/R LESS THAN 40                     HYST0810
C*************************************************************************     HYST0820
C                                                                               HYST0830
C INITIAL BUCKLING LOAD                                                         HYST0840
C                                                                               HYST0850
      IF( ESR .LT. CC  AND. CYCLE .EQ. 1 ) THEN                                 HYST0860
        PMAX=-((1.-(ESR**2/(2.*CC**2)))*YS*AREA                                 HYST0870
      ELSE IF( ESR .GE. CC .AND  CYCLE .EQ. 1 ) THEN                            HYST0880
        PMAX=-PI*PI*AREA*EM/(ESR**2)                                            HYST0890
      ENDIF                                                                     HYST0900
C                                                                               HYST0910
C RESIDUAL ELOGATION FACTOR RFAC                                                HYST0920
C                                                                               HYST0930
C                 /** FOR BOX AND ANGLE SECTION **/                            HYST0940
      IF(SHAPE .EQ. 1) THEN                                                     HYST0950
        RFAC=1.75                                                               HYST0960
      ELSE IF(SHAPE .EQ. 3) THEN                                                HYST0970
        RFAC=0                                                                  HYST0980
C                 /** FOR I SHAPE SECTION       **/                           HYST0990
      ELSE IF(SHAPE .EQ. 2) THEN                                                HYST1000
        RFAC=1.4                                                                HYST1010
      ENDIF                                                                     HYST1020
C                                                                               HYST1030
C GENERATE CONTROL POINTS COORDINATE                                            HYST1040
C                                                                               HYST1050
C  (1)  POINT A                                                                 HYST1060
C                                                                               HYST1070
      IF( SHAPE .NE. 3 ) THEN                                                   HYST1080
        IF( CYCLE .EQ. 1 ) THEN                                                 HYST1090
          VA=PMAX/PY                                                            HYST1100
          HA=PMAX/(ES*DELY)                                                     HYST1110
        ELSE IF( CYCLE .EQ. 2) THEN                                             HYST1120
          IF( ESR .LE. 40 ) THEN                                                HYST1130
            PMAX=-(23.4/ESR)*PY                                                 HYST1140
          ELSE IF( ESR .GT. 40 ) THEN                                           HYST1150
            PMAX=-(30 /ESR)*PY                                                  HYST1160
          ENDIF                                                                 HYST1170
        ELSE IF( CYCLE .GT. 2) THEN                                             HYST1180
          IF( ESR .LE. 40 ) THEN                                                HYST1190
            PMAX=-(18.5/ESR)*PY                                                 HYST1200
          ELSE IF( ESR .GT. 40 ) THEN                                           HYST1210
            PMAX=-(25./ESR)*PY                                                  HYST1220
          ENDIF                                                                 HYST1230
        ENDIF                                                                   HYST1240
      ELSE IF( SHAPE .EQ. 3 ) THEN                                              HYST1250
        IF( CYCLE .EQ. 1) THEN                                                  HYST1260
          VA=PMAX/PY                                                            HYST1270
          HA=PMAX/(ES*DELY)                                                     HYST1280
        ELSE IF( CYCLE .EQ. 2) THEN                                             HYST1290
          PMAX=PMAX                                                             HYST1300
        ELSE IF( CYCLE .EQ. 3) THEN                                             HYST1310
          PMAX=-(23.4/ESR)*PY                                                   HYST1320
        ELSE IF( CYCLE .GT. 3) THEN                                             HYST1330
          PMAX=-(18.5/ESR)*PY                                                   HYST1340
        ENDIF                                                                   HYST1350
      ENDIF                                                                     HYST1360
C                                                                               HYST1370
C  (2)  POINT B                                                                 HYST1380
C                                                                               HYST1390
      HB=.5                                                                     HYST1400
      IF( ESR .LE. 40 ) THEN                                                    HYST1410
        VB=-17./ESR                                                             HYST1420
      ELSE IF( ESR .GT. 40 ) THEN                                               HYST1430
C               /** FOR BOX AND ANGLE SECTION **/                              HYST1440
        IF( SHAPE .EQ. 1  OR. SHAPE .EQ. 3 ) THEN                               HYST1450
          VB=-18./ESR                                                           HYST1460
C               /** FOR I SHAPE SECTION       **/                            HYST1470
        ELSE IF( SHAPE .EQ. 2 ) THEN                                            HYST1480
          VB=-11.3/ESR                                                          HYST1490
        ENDIF                                                                   HYST1500
      ENDIF                                                                     HYST1510
C                                                                               HYST1520
C  (3)  POINT C                                                                 HYST1530
C                                                                               HYST1540
      IF( IDUC .EQ. 0.) THEN                                                    HYST1550
        HC=-12.                                                                 HYST1560
        IF( ESR .LE. 40 ) THEN                                                  HYST1570
          VC=-15./ESR                                                           HYST1580
        ELSE IF( ESR .GT. 40 ) THEN                                             HYST1590
C               /** FOR BOX AND ANGLE SECTION **/                              HYST1600
          IF( SHAPE .EQ. 1 .OR  SHAPE .EQ. 3 ) THEN                             HYST1610
            VC=-12./ESR                                                         HYST1620
C               /** FOR I SHAPE SECTION       **/                            HYST1630
          ELSE IF( SHAPE .EQ. 2 ) THEN                                          HYST1640
            VC=-8.5/ESR                                                         HYST1650
          ENDIF                                                                 HYST1660
        ENDIF                                                                   HYST1670
      ENDIF                                                                     HYST1680
C                                                                               HYST1690
C  (4)  POINT E                                                                 HYST1700
C                                                                               HYST1710
        HE=1.+(REN*EL/DELY)                                                     HYST1720
        VE=1                                                                    HYST1730
C                                                                               HYST1740
C  (5)  POINT D1                                                                HYST1750
C                                                                               HYST1760
      SLOPE=(VE-VC)/(HE-HC)                                                     HYST1770
      HD1=(SLOPE*HC-VC)/(SLOPE-0.267949)                                        HYST1780
      VD1=-0.267949*HD1                                                         HYST1790
C                                                                               HYST1800
C  (6)  POINT D                                                                 HYST1810
C                                                                               HYST1820
      IF( ESR .LE. 40 ) THEN                                                    HYST1830
        HD=HD1*31./ESR                                                          HYST1840
        VD=VD1*31./ESR                                                          HYST1850
      ELSE IF( ESR .GT. 40 ) THEN                                               HYST1860
C               /** FOR BOX AND ANGLE SECTION **/                              HYST1870
        IF( SHAPE .EQ. 1 .OR. SHAPE .EQ. 3 ) THEN                               HYST1880
          HD=HD1*60./ESR                                                        HYST1890
          VD=VD1*60./ESR                                                        HYST1900
C               /** FOR I SHAPE SECTION       **/                            HYST1910
        ELSE IF( SHAPE .EQ. 2 ) THEN                                            HYST1920
          HD=HD1*20./ESR                                                        HYST1930
          VD=VD1*20./ESR                                                        HYST1940
        ENDIF                                                                   HYST1950
      ENDIF                                                                     HYST1960
C                                                                               HYST1970
C DEFINE ENVELOPE SLOPE                                                         HYST1980
C                                                                               HYST1990
      SLOP1=1.                                                                  HYST2000
      SLOP2=(VB-VA)/(HB-HA)                                                     HYST2010
      SLOP3=(VC-VB)/(HC-HB)                                                     HYST2020
      SLOP4=(VD-VC)/(HD-HC)                                                     HYST2030
      SLOP5=(VE-VD)/(HE-HD)                                                     HYST2040
      SLOP6=0.0001                                                              HYST2050
C                                                                               HYST2060
C DEFINE ENVELOPE INTERMIDIATE POINTS                                           HYST2070
C                                                                               HYST2080
C  (1)  POINT H                                                                 HYST2090
C                                                                               HYST2100
      VH=(VB+SLOP2*(-VD+HD-HB))/(1.-SLOP2)                                      HYST2110
      HH=VH-VD+HD                                                               HYST2120
C                                                                               HYST2130
C  (2)  POINT C1                                                                HYST2140
C                                                                               HYST2150
      VC1=0.                                                                    HYST2160
      HC1=HC-(VC/SLOP4)                                                         HYST2170
C                                                                               HYST2180
C  (3)  POINT (PREF,DREF) - CALCULATE IN RULE NO  6                             HYST2190
C                                                                               HYST2200
C                                                                               HYST2210
C  (4)  POINT (HREP,VREF)   CALCULATE IN RULE NO. 5                             HYST2220
C                                                                               HYST2230
C--- DEFINE CURRENT REGIN ZONE NUMBER                                           HYST2240
      PREGIN=REGIN                                                              HYST2250
      IF( DEL .GE. HA) REGIN=1                                                  HYST2260
      IF( DEL .GE. HH .AND. DEL .LT. HA ) REGIN=2                               HYST2270
      IF( DEL .GE. HB .AND. DEL .LT. HH ) REGIN=3                               HYST2280
      IF( DEL .GE. HC .AND  DEL .LT. HB ) REGIN=4                               HYST2290
C                                                                               HYST2300
C     GOEL'S HYSTORESIS LOOP RULES                                              HYST2310
C                                                                               HYST2320
C  (1) RULE 1                                                                   HYST2330
C                                                                               HYST2340
      IF( RULE .EQ. 1 ) THEN                                                    HYST2350
          IF( P  GE. PMAX .AND. P  LT. PY ) THEN                                HYST2360
              STIF=SLOP1*ES                                                     HYST2370
              EKI=SLOP1*ES                                                      HYST2380
              IF( DEMAX .LT. 0 .AND  DE .LE. DEMAX ) THEN                       HYST2390
                REN=((0.55*ABS(DE)/ESR)+0.0002*ABS(DE)**2)*RFAC                 HYST2400
              ENDIF                                                             HYST2410
          ELSE IF( P .GE. PY ) THEN                                             HYST2420
              STIF=SLOP6                                                        HYST2430
              RULE=6                                                            HYST2440
              EKI=SLOP1*ES                                                      HYST2450
          ELSE IF( P  LT. PMAX) THEN                                            HYST2460
              DI=DE-HA*DELY                                                     HYST2470
              SLOP2=(VB-VA)/(HB-HA)                                             HYST2480
              STIF=SLOP2*ES                                                     HYST2490
              P=PMAX+(STIF*DI)                                                  HYST2500
              PL=P/PY                                                           HYST2510
C                                                                               HYST2520
              RULE=2                                                            HYST2530
              IF( DEMAX .LT. 0 .AND. DE .LE. DEMAX ) THEN                       HYST2540
                REN=((0.55*ABS(DE)/ESR)+0.0002*ABS(DE)**2)*RFAC                 HYST2550
              ENDIF                                                             HYST2560
              EKI=SLOP1*ES                                                      HYST2570
          ENDIF                                                                 HYST2580
          CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,EKI)                         HYST2590
          CALL DNE(1, DUC, EXCR, DELY, DA, ESE, PSE, PSEOLD, CSE )              HYST2600
      RETURN                                                                    HYST2610
      ENDIF                                                                     HYST2620
C                                                                               HYST2630
C  (2) RULE 2                                                                   HYST2640
C                                                                               HYST2650
      IF( RULE .EQ  2.) THEN                                                    HYST2660
          IF( DEL  GE. HH) THEN                                                 HYST2670
              IF( DDE .LE  0  ) THEN                                            HYST2680
```

```
                    IF(PREGIN .NE. REGIN ) REVRSC= .FALSE.
                    STIF=SLOP2*ES
                  IF( DEMAX .LT. 0 .AND. DE .LE. DEMAX ) THEN
                    RSN=((0.55*ABS(DE)/ESR)+0.0002*ABS(DE)**2)*RFAC
                  ELSE IF( DEMAX .GT. 0 ) THEN
                    RSN=((0.55*ABS(DE)/ESR)+0.0002*ABS(DE)**2)*RFAC
                  ENDIF
C                 ----------------------------------------
                    IF(.NOT. REVRSC )THEN
                      EKI=SLOP1*ES
                    ELSE IF( REVRSC )THEN
                      EKI=ES*((PTOP-PL)/(DTOP-DEL))
                    ENDIF
C                   ----------------------------------------
                ELSE IF( DDE .GT. 0. )THEN
                  IF(.NOT. REVRSC )THEN
                    RULE=7
                    STIF=SLOP1*ES
C
                    PLA=PL-SLOP2*DDEL
                    DLA=DEL-DDEL
                    DBOT=DLA
                    PBOT=PLA
                    PL=PLA+SLOP1*DDEL
                    P=PL*PY
C
                    PTOP=VE+(SLOP5*(PBOT-VE-SLOP1*DBOT
     *                   +SLOP1*HE)/(SLOP5-SLOP1))
                    DTOP=(PBOT-VE+SLOP5*HE-SLOP1*DBOT)
     *                   /(SLOP5-SLOP1)
                    EKI=ES*SLOP1
                  ELSE IF( REVRSC ) THEN
                    RULE=12
                    DBOT=DEL
                    PBOT=P/PY
                    SLOP7=(PTOP-PBOT)/(DTOP-DBOT)
                    STIF=SLOP7*ES
                    EKI=ES*SLOP7
                  ENDIF
                ENDIF
            ELSE IF(DEL .LT. HH .AND. DEL .GE. HB ) THEN
                IF( DDE .LE. 0. ) THEN
                    STIF=SLOP2*ES
                  IF( DEMAX .LT. 0 .AND. DE .LE. DEMAX ) THEN.
                    RSN=((0.55*ABS(DE)/ESR)+0.0002*ABS(DE)**2)*RFAC
                  ELSE IF( DEMAX .GT. 0 ) THEN
                    RSN=((0.55*ABS(DE)/ESR)+0.0002*ABS(DE)**2)*RFAC
                  ENDIF
C                 ----------------------------------------
                    IF(.NOT. REVRSC )THEN
                      EKI=SLOP1*ES
                    ELSE IF( REVRSC ) THEN
                      EKI=ES*((PTOP-PL)/(DTOP-DEL))
                    ENDIF
C                   ----------------------------------------
                ELSE IF( DDE .GT. 0. )THEN
                  IF( .NOT. REVRSC ) THEN
                    RULE=8
                    STIF=SLOP1*ES
C
                    PLA=PL SLOP2*DDEL
                    DLA=DEL-DDEL
                    DBOT=DLA
                    PBOT=PLA
                    PL=PLA+SLOP1*DDEL
                    P=PL*PY
C
                    PTOP=VD+(SLOP4*(PBOT-VD-SLOP1*DBOT
     *                   +SLOP1*HD)/(SLOP4-SLOP1))
                    DTOP=(PBOT-VD+SLOP4*HD-SLOP1*DBOT)
     *                   /(SLOP4-SLOP1)
                    EKI=ES*SLOP1
                  ELSE IF( REVRSC ) THEN
                    RULE=11
                    DBOT=DEL
                    PBOT=P/PY
                    PTOP=VD
                    DTOP=HD
                    SLOP7=(PTOP-PBOT)/(DTOP-DBOT)
                    STIF=SLOP7*ES
                    EKI=ES*SLOP7
                  ENDIF
                ENDIF
            ELSE IF( DEL .LT. HB) THEN
                IF(PREGIN .NE. REGIN ) REVRSC= .FALSE
                RULE=3
                EKI=ES*SLOP1
            ENDIF
        IF( REVRSC ) IRV=1.
        IF ( .NOT. REVRSC) IRV=0.
        CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,EKI)
            IF( ABS(DEMAX) .LT. ABS(DE) ) THEN
              CALL DNE(1, DUC, EXCR, DELY, DA, ESE, PSE, PSEOLD, CSE )
              DEMAX=DE
            ENDIF
        RETURN
        ENDIF
C
C   (3) RULE 3
C
        IF( RULE .EQ. 3.) THEN
            IF( DEL .GT. HC) THEN
                IF( DDE .LE. 0. ) THEN
                    IF(PREGIN .NE. REGIN ) REVRSC= .FALSE.
                    STIF=SLOP3*ES
                  IF( DEMAX .LT. 0 .AND. DE .LE. DEMAX ) THEN
                    RSN=((0.55*ABS(DE)/ESR)+0.0002*ABS(DE)**2)*RFAC
                  ELSE IF( DEMAX .GT. 0 ) THEN
                    RSN=((0.55*ABS(DE)/ESR)+0.0002*ABS(DE)**2)*RFAC
                  ENDIF
C                 ----------------------------------------
                    IF(.NOT. REVRSC )THEN
                      EKI=SLOP1*ES
                    ELSE IF( REVRSC )THEN
                      EKI=ES*((PTOP-PL)/(DTOP-DEL))
                    ENDIF
C                   ----------------------------------------
                ELSE IF( DDE .GT. 0. )THEN
                  IF( DEL .GT. -12. ) THEN
                    IF( .NOT. REVRSC )THEN
                      RULE=9
                      STIF=SLOP1*ES
C
                      PLA=PL-SLOP2*DDEL
                      DLA=DEL-DDEL
                      DBOT=DLA
                      PBOT=PLA
                      PL=PLA+SLOP1*DDEL
                      P=PL*PY
C
                      PTOP=VD+(SLOP4*(PBOT-VD-SLOP1*DBOT
     *                     +SLOP1*HD)/(SLOP4-SLOP1))
                      DTOP=(PBOT-VD+SLOP4*HD-SLOP1*DBOT)
     *                     /(SLOP4-SLOP1)
                      EKI=ES*SLOP1
                    ELSE IF( REVRSC ) THEN
                      RULE=10
                      DBOT=DEL
                      PBOT=P/PY
                      SLOP7=(PTOP-PBOT)/(DTOP-DBOT)
                      STIF=SLOP7*ES
                      EKI=ES*SLOP7
                    ENDIF
                  ELSE IF ( DEL .LE. -12 ) THEN
                    REVRSC= .FALSE.
                    RULE=4
                      STIF=SLOP4*ES
                      EKI=ES*SLOP4
                    ENDIF
                  ENDIF
                ELSE IF( DEL .LE. HC ) THEN
                  IF( DDE .LT. 0 ) THEN
                    RULE=3
                    HC=DEL
                    VC=PL
                    IDUC=1.
                    STIF=-SLOP5*ES
                    EKI=ES*((VD-VC)/(HD-HC))
                        IF( DEMAX .LT. 0 .AND. DE .LE. DEMAX ) THEN
                          RSN=((0.55*ABS(DE)/ESR)+0.0002*ABS(DE)**2)*RFAC
                        ELSE IF( DEMAX .GT. 0 ) THEN
                          RSN=((0.55*ABS(DE)/ESR)+0.0002*ABS(DE)**2)*RFAC
                        ENDIF
                  ELSE IF( DDE .GT. 0 ) THEN
                    RULE=4
                    STIF=SLOP4*ES
                    EKI=ES*SLOP4
                  ENDIF
                ENDIF
            IF( REVRSC ) IRV=1.
            IF ( .NOT. REVRSC) IRV=0.
            CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,EKI)
                IF( ABS(DEMAX) .LT. ABS(DE) ) THEN
                  CALL DNE(1, DUC, EXCR, DELY, DA, ESE, PSE, PSEOLD, CSE )
                  DEMAX=DE
                ENDIF
            RETURN
            ENDIF
C
C   (4) RULE 7
C
            IF( RULE .EQ. 7.) THEN
                IF( DEL .GT. DBOT .AND. DEL .LT. DTOP .AND. P .GT. PMAX) THEN
                    STIF=SLOP1*ES
                    EKI=ES*SLOP1
                ELSE IF(DEL .LT. DBOT .OR. P .LE. PMAX) THEN
                    IF( P .LE. PMAX ) THEN
                        REVRSC= .FALSE.
                        PTOP=0
                        DTOP=0
                        PBOT=PL
                        DBOT=DEL
                        S1=PLA/PY
                        S2=PMAX/PY
                        HA=(DELA/DELY) + ((S2-S1)/(SLOP1))
                        VA=S2
                        SLOP2=(VB-VA)/(HB-HA)
                    ENDIF
                    STIF=SLOP2*ES
                    RULE=2
C
                    PL=VA+ SLOP2*(DBOT-HA)
                    P=PL*PY
C
                    EKI=ES*((PTOP-PL)/(DTOP-DEL))
                ELSE IF(DEL .GT. DTOP .AND. PL .LT. VE) THEN
                    STIF=SLOP5*ES
                    RULE=5
                    EKI=ES*SLOP1
                ELSE IF(DEL .GT. DTOP .AND PL .GE. VE) THEN
                    STIF=SLOP6*ES
                    RULE=6
                    EKI=ES*SLOP1
                ENDIF
            IF( REVRSC ) IRV=1.
            IF ( .NOT. REVRSC) IRV=0
            CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,EKI)
            RETURN
            ENDIF
C
C   (5) RULE 8
C
            IF( RULE .EQ. 8.) THEN
                IF( DEL .GT. DBOT .AND. DEL .LT. DTOP ) THEN
                    STIF=SLOP1*ES
                    EKI=ES*SLOP1
                ELSE IF(DEL .LT. DBOT ) THEN
                    STIF=SLOP2*ES
                    RULE=2
C
                    PL=PBOT-(DBOT-DEL)*SLOP2
                    P=PL*PY
C
                    EKI=ES*((PTOP-PL)/(DTOP-DEL))
                ELSE IF(DEL .GT. DTOP ) THEN
                    STIF=SLOP4*ES
                    RULE=4
                    EKI=ES*SLOP1
                ENDIF
            IF( REVRSC ) IRV=1.
            IF ( .NOT. REVRSC) IRV=0
            CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,EKI)
            RETURN
            ENDIF
C
C   (6) RULE 9
C
            IF( RULE .EQ. 9.) THEN
                IF( DEL .GT. DBOT .AND. DEL .LT. DTOP ) THEN
                    STIF=SLOP1*ES
                    EKI=ES*SLOP1
                ELSE IF(DEL .LT. DBOT ) THEN
                    STIF=SLOP3*ES
                    RULE=3
C
                    PL=PBOT-(DBOT-DEL)*SLOP3
                    P=PL*PY
C
                    EKI=ES*((PTOP-PL)/(DTOP-DEL))
                ELSE IF(DEL .GT. DTOP ) THEN
                    STIF=SLOP4*ES
                    RULE=4
                    EKI=ES*SLOP1
                ENDIF
            IF( REVRSC ) IRV=1
            IF ( .NOT. REVRSC) IRV=0
            CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,EKI)
            RETURN
            ENDIF
C
C   (7) RULE 4
C
            IF( RULE .EQ. 4.) THEN
                IF (DEL .GE. HC1 .AND. DEL .LT. HD ) THEN
                    IF(DDE .GE. 0. ) THEN
                        STIF=SLOP4*ES
                        EKI=ES*SLOP1
                    ELSE IF( DDE .LT. 0 ) THEN
                        PREF=VD+(SLOP4*(VB-VD-SLOP1*HB
     *                       +SLOP1*HD)/(SLOP4-SLOP1))
                        DREF=(VB-VD+SLOP4*HD-SLOP1*HB)/(SLOP4-SLOP1)
                        IF( DEL .LT. DREF) THEN
                            PTOP=P/PY
                            DTOP=DEL
                            DBOT=(SLOP3*HB-VB+PTOP-SLOP1*DEL)/(SLOP3-SLOP1)
                            PBOT=VB-SLOP3*(DBOT-HB)
                            STIF=SLOP1*ES
                            RULE=9
                            REVRSC= .TRUE.
                            EKI=ES*SLOP1
                        ELSE IF( DEL .GE. DREF) THEN
                            PTOP=P/PY
```

```
                        DTOP=DEL                                                HYST5610
                        DBOT=(SLOP2*HA-VA+PTOP-SLOP1*DEL)/(SLOP2-SLOP1)HYST5620
                        PBOT=VA+SLOP2*(DBOT-HA)                                 HYST5630
                        STIF=SLOP1*ES                                           HYST5640
                        RULE=8                                                  HYST5650
                        REVRSC= .TRUE.                                          HYST5660
                        EKI=ES*SLOP1                                            HYST5670
                      ENDIF                                                     HYST5680
                    ENDIF                                                       HYST5690
               ELSE IF( DEL .GT. HD .AND  DDE .GT. 0. ) THEN                    HYST5700
                    STIF=SLOP5*ES                                               HYST5710
                    RULE=5                                                      HYST5720
                    EKI=ES*SLOP1                                                HYST5730
               ELSE IF( DEL .LT. HC1 .AND. DEL .GE. HC ) THEN                   HYST5740
                    REVRSC= .FALSE.                                             HYST5750
                    RULE=4                                                      HYST5760
                    STIF=SLOP4*ES                                               HYST5770
                    EKI=ES*SLOP4                                                HYST5780
               ELSE IF( DEL .LT  HC ) THEN                                      HYST5790
                    REVRSC= .FALSE.                                             HYST5800
                    RULE=3                                                      HYST5810
                    HC=DEL                                                      HYST5820
                    VC=PL                                                       HYST5830
                    IDUC=1.                                                     HYST5840
                    STIF=SLOP6*ES                                               HYST5850
                    EKI=ES*((VD-VC)/(HD-HC))                                    HYST5860
               ENDIF                                                            HYST5870
          IF( REVRSC ) IRV=1                                                    HYST5880
          IF ( .NOT. REVRSC) IRV=0.                                             HYST5890
          CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,EKI)                         HYST5900
               IF( ABS(DEMAX) .LT  ABS(DE) ) THEN                               HYST5910
                 CALL DNE(1, DUC, EXCR, DELY, DA, ESE, PSE, PSEOLD, CSE )       HYST5920
                 DEMAX=DE                                                       HYST5930
               ENDIF                                                            HYST5940
          RETURN                                                                HYST5950
          ENDIF                                                                 HYST5960
C                                                                               HYST5970
C  (8) RULE 5                                                                   HYST5980
C                                                                               HYST5990
          IF( RULE  EQ. 5.) THEN                                                HYST6000
               IF (DEL .LT. HE ) THEN                                           HYST6010
                    IF(DDE .GE. 0. ) THEN                                       HYST6020
                         STIF=SLOP5*ES                                          HYST6030
                         EKI=ES*SLOP1                                           HYST6040
                    ELSE IF( DDE .LT. 0  ) THEN                                 HYST6050
                         BREP=VA-SLOP1*HA                                       HYST6060
                         HREP=(VD-SLOP5*HD-BREP)/(SLOP1-SLOP5)                  HYST6070
                         IF( DEL .GE. HREP ) CYCLE=CYCLE+1                      HYST6080
                         PTOP=P/PY                                              HYST6090
                         DTOP=DEL                                               HYST6100
                         DBOT=(SLOP2*HA-VA+PTOP-SLOP1*DEL)/(SLOP2-SLOP1)HYST6110
                         PBOT=VA+SLOP2*(DBOT-HA)                                HYST6120
                         STIF=SLOP1*ES                                          HYST6130
                         RULE=7                                                 HYST6140
                         REVRSC= .TRUE.                                         HYST6150
                         EKI=ES*SLOP1                                           HYST6160
                    ENDIF                                                       HYST6170
               ELSE IF( DEL .GE. HE .AND. DDE .GT. 0. ) THEN                    HYST6180
                    STIF=SLOP6                                                  HYST6190
                    RULE=6                                                      HYST6200
                    EKI=ES*SLOP1                                                HYST6210
               ENDIF                                                            HYST6220
          IF( REVRSC ) IRV=1.                                                   HYST6230
          IF ( .NOT. REVRSC) IRV=0.                                             HYST6240
          CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,EKI)                         HYST6250
               IF( ABS(DEMAX) .LT  ABS(DE) ) THEN                               HYST6260
                 CALL DNE(1, DUC, EXCR, DELY, DA, ESE, PSE, PSEOLD, CSE )       HYST6270
                 DEMAX=DE                                                       HYST6280
               ENDIF                                                            HYST6290
          RETURN                                                                HYST6300
          ENDIF                                                                 HYST6310
C                                                                               HYST6320
C  (9) RULE 6                                                                   HYST6330
C                                                                               HYST6340
          IF( RULE .EQ  6.) THEN                                                HYST6350
               IF(DDE .GE  0.) THEN                                             HYST6360
                    STIF=SLOP6                                                  HYST6370
                    EKI=ES*SLOP1                                                HYST6380
               ELSE IF(DDE .LT. 0.) THEN                                        HYST6390
                    STIF=SLOP1*ES                                               HYST6400
                    RULE=1                                                      HYST6410
                    CYCLE=CYCLE+1.                                              HYST6420
                    REVRSC=  FALSE.                                             HYST6430
                    EKI=ES*SLOP1                                                HYST6440
               ENDIF                                                            HYST6450
          IF( REVRSC ) IRV=1.                                                   HYST6460
          IF ( .NOT. REVRSC) IRV=0.                                             HYST6470
          CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,EKI)                         HYST6480
               IF( ABS(DEMAX) .LT. ABS(DE) ) THEN                               HYST6490
                 CALL DNE(1, DUC, EXCR, DELY, DA, ESE, PSE, PSEOLD, CSE )       HYST6500
                 DEMAX=DE                                                       HYST6510
               ENDIF                                                            HYST6520
          RETURN                                                                HYST6530
          ENDIF                                                                 HYST6540
C                                                                               HYST6550
C  (10) RULE 10                                                                 HYST6560
C                                                                               HYST6570
          IF( RULE  EQ. 10 ) THEN                                               HYST6580
               IF( DEL .GT. DBOT .AND. DEL .LT. DTOP ) THEN                     HYST6590
                    STIF=SLOP7*ES                                               HYST6600
                    EKI=ES*SLOP7                                                HYST6610
               ELSE IF(DEL .LT. DBOT ) THEN                                     HYST6620
                    STIF=SLOP3*ES                                               HYST6630
                    RULE=3                                                      HYST6640
                    REVRSC= .FALSE.                                             HYST6650
                    PTOP=0                                                      HYST6660
                    DTOP=0                                                      HYST6670
                    PBOT=0                                                      HYST6680
                    DBOT=0                                                      HYST6690
                    EKI=ES*((PTOP-PL)/(DTOP-DEL))                               HYST6700
               ELSE IF(DEL .GT  DTOP ) THEN                                     HYST6710
                    STIF=SLOP4*ES                                               HYST6720
                    RULE=4                                                      HYST6730
                    EKI=ES*SLOP1                                                HYST6740
               ENDIF                                                            HYST6750
          IF( REVRSC ) IRV=1                                                    HYST6760
          IF ( .NOT  REVRSC) IRV=0                                              HYST6770
          CALL STRENG(ESE,PSE,PSZOLD,P,PLA,DE,DELA,EKI)                         HYST6780
          RETURN                                                                HYST6790
          ENDIF                                                                 HYST6800
C                                                                               HYST6810
C  (11) RULE 11                                                                 HYST6820
C                                                                               HYST6830
          IF( RULE  EQ. 11 ) THEN                                               HYST6840
               IF( DEL .GT. DBOT .AND. DEL  LT  DTOP ) THEN                     HYST6850
                    STIF=SLOP7*ES                                               HYST6860
                    EKI=ES*SLOP7                                                HYST6870
               ELSE IF(DEL .LT  DBOT ) THEN                                     HYST6880
                    STIF=SLOP2*ES                                               HYST6890
                    RULE=2                                                      HYST6900
                    REVRSC= .FALSE                                              HYST6910
                    PTOP=0                                                      HYST6920
                    DTOP=0                                                      HYST6930
                    PBOT=0                                                      HYST6940
                    DBOT=0                                                      HYST6950
                    EKI=ES*((VD-PL)/(HD-DEL))                                   HYST6960
               ELSE IF(DEL .GT. DTOP ) THEN                                     HYST6970
                    STIF=SLOP5*ES                                               HYST6980
                    RULE=5                                                      HYST6990
                    EKI=ES*SLOP1                                                HYST7000
               ENDIF                                                            HYST7010
          IF( REVRSC ) IRV=1                                                    HYST7020
          IF ( .NOT. REVRSC) IRV=0.                                             HYST7030
          CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,EKI)                         HYST7040
          RETURN                                                                HYST7050
          ENDIF                                                                 HYST7060
C                                                                               HYST7070
C  (12) RULE 12                                                                 HYST7080
C                                                                               HYST7090
          IF( RULE  EQ. 12 ) THEN                                               HYST7100
               IF( DEL .GT. DBOT .AND. DEL .LT. DTOP ) THEN                     HYST7110
                    STIF=SLOP7*ES                                               HYST7120
                    EKI=ES*SLOP7                                                HYST7130
               ELSE IF(DEL  LT. DBOT ) THEN                                     HYST7140
                    STIF=SLOP2*ES                                               HYST7150
                    RULE=2                                                      HYST7160
                    REVRSC= .FALSE.                                             HYST7170
                    PTOP=0                                                      HYST7180
                    DTOP=0                                                      HYST7190
                    PBOT=0                                                      HYST7200
                    DBOT=0                                                      HYST7210
                    EKI=ES*((PTOP-PL)/(DTOP DEL))                               HYST7220
               ELSE IF(DEL .GT. DTOP ) THEN                                     HYST7230
                    STIF=SLOP5*ES                                               HYST7240
                    RULE=5                                                      HYST7250
                    EKI=ES*SLOP1                                                HYST7260
               ENDIF                                                            HYST7270
          ENDIF                                                                 HYST7280
          IF( REVRSC ) IRV=1.                                                   HYST7290
          IF ( .NOT. REVRSC) IRV=0.                                             HYST7300
          CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,EKI)                         HYST7310
          RETURN                                                                HYST7320
          END                                                                   HYST7330
C                                                                               HYST0010
C******* HYST09 ************************************************               HYST0020
C                                                                               HYST0030
C     PINO-SUAREZ TOWER LONG DIRECTION GIRDER MOMENT-ROTATION                   HYST0040
C     HYSTERESIS MODEL BASED ON HYSTERESIS LOOP OF T2A GIRDER                   HYST0050
C                                                                               HYST0060
      SUBROUTINE HYST09(MOPT,STIF,P,X,PLA,XLA,DX,RV,RULE,PTOP,DTOP,            HYST0070
     &      PBOT,DBOT,SPTOP,SDTOP,SPBOT,SDBOT,SSPTOP,SSDTOP,SSPBOT,            HYST0080
     &      SSDBOT,HA,VA,HB,VB,HD,VD,HE,VE,HAP,VAP,HBP,VBP,SSHD,               HYST0090
     &      SSVD,SSHE,SSVE,SSSPTP,SSSDTP,SSSPBT,SSSDBT,SHD,SVD,SHE,            HYST0100
     &      SVE,DUC,DEMAX,FACH,FACV,FAC,DUCMAX)                                HYST0110
      DIMENSION DUC(3)                                                          HYST0120
      LOGICAL HIGH                                                              HYST0130
C                                                                               HYST0140
C ------------------------------------------------------------------            HYST0150
C  DUC     ---- ELEMENT DUCTILITY RATE                                         HYST0160
C  EXCR    ---- ELEMENT EXCURSION RATE                                         HYST0170
C  ESE     ---- ELEMENT ELASTIC STRAIN RATE                                    HYST0180
C  PSE     ---- ELEMENT PLASTIC STRAIN RATE                                    HYST0190
C  PSEOLD  ---- ELEMENT PLASTIC STRAIN RATE AT LAST ZERO CROSSING              HYST0200
C  CSE     ---- ELEMENT CONSTANT STRAIN ENERGY                                 HYST0210
C  STIF    ---- ELEMENT STIFFNESS                                              HYST0220
C  P       ---- CURRENT ELEMENT INTERNAL FORCE                                 HYST0230
C  X       ---- CURRENT ELEMENT INTERNAL DISPLACEMENT                          HYST0240
C  PLA     ---- PREVIOUS ELEMENT INTERNAL FORCE                               HYST0250
C  XLA     ---- PREVIOUS ELEMENT INTERNAL DISPLACEMENT                        HYST0260
C  DP      ---- ELEMENT INTERNAL FORCE INCREMENT                              HYST0270
C  DX      ---- ELEMENT INTERNAL DISPLACEMENT INCREMENT                       HYST0280
C  HIGH    ---- LOGICAL INDEX, IF P .GT. 0 THEN HIGH=.TRUE.                   HYST0290
C  RULE    ---- RULE NUMBER                                                   HYST0300
C  PTOP    ---- REVERSED LOADING POINT AT TOP ENVELOPE                        HYST0310
C  DTOP    ---- REVERSED DISP. POINT AT TOP ENVELOPE                          HYST0320
C  PBOT    ---- REVERSED LOADING POINT AT BOTTOM ENVELOPE                     HYST0330
C  DBOT    ---- REVERSED DISP. POINT AT BOTTOM ENVELOPE                       HYST0340
C  SPTOP   ---- REVERSED LOADING POINT AT TOP SMALL LOOP                      HYST0350
C  SDTOP   ---- REVERSED DISP. POINT AT TOP SMALL LOOP                        HYST0360
C  SPBOT   ---- REVERSED LOADING POINT AT BOTTOM SMALL LOOP                   HYST0370
C  SDBOT   ---- REVERSED DISP. POINT AT BOTTOM SMALL LOOP                     HYST0380
C  SSPTOP  ---- REVERSED LOADING POINT AT TOP SECOND SMALL LOOP               HYST0390
C  SSDTOP  ---- REVERSED DISP. POINT AT TOP SECOND SMALL LOOP                 HYST0400
C  SSPBOT  ---- REVERSED LOADING POINT AT BOTTOM SECOND SMALL LOOP            HYST0410
C  SSDBOT  ---- REVERSED DISP. POINT AT BOTTOM SCOND SMALL LOOP               HYST0420
C  SSSPTP  ---- REVERSED LOADING POINT AT TOP THIRD OR ABOVE SMALL LOOP       HYST0430
C  SSSDTP  ---- REVERSED DISP. POINT AT TOP THIRD OR ABOVE SMALL LOOP         HYST0440
C  SSSPBT  ---- REVERSED LOADING POINT AT BOT. THIRD OR ABOVE SMALL LOO       HYST0450
C  SSSDBT  ---- REVERSED DISP  POINT AT BOT. THIRD OR ABOVE  SMALL LOOP       HYST0460
C  HA,VA   ---- FIRST BUCKLING POINT AT TOP ENVELOPE                          HYST0470
C  HAP,VAP ---- FIRST BUCKLING POINT AT BOTTOM ENVELOPE                       HYST0480
C  HB,VB   ---- TOP ENVELOPE CONTROL POINT                                    HYST0490
C  HBP,VBP ---- BOTTOM ENVELOPE CONTROL POINT                                 HYST0500
C  HD,VD   ---- FIRST SMALL LOOP CONTROL POINT                                HYST0510
C  HE,VE   ---- FIRST SMALL LOOP CONTROL POINT                                HYST0520
C  SHD,SVD --- SECOND SMALL LOOP CONTROL POINT                                HYST0530
C  SHE,SVE --- SECOND SMALL LOOP CONTROL POINT                                HYST0540
C  SSHD,SSVD - THIRD OR ABOVE SMALL LOOP CONTROL POINT                        HYST0550
C  SSHE,SSVE - THIRD OR ABOVE SMALL LOOP CONTROL POINT                        HYST0560
C  DUCMAX    - FAILURE DUCTILITY                                              HYST0570
C              .GE  1 FAILURE DUCTILITY CONSIDERED                            HYST0580
C              .LT  1 FAILURE DUCTILITY NOT CONSIDERED                        HYST0590
C ------------------------------------------------------------------            HYST0600
C                                                                               HYST0610
C ----------------------------                                                  HYST0620
C - CHECK FAILURE DUCTILITY --                                                  HYST0630
C ----------------------------                                                  HYST0640
          IF( DUC(1) .GE. DUCMAX .AND  DUCMAX .GE. 1) THEN                      HYST0650
            P=0                                                                 HYST0660
            STIF=0.001                                                          HYST0670
            RETURN                                                              HYST0680
          ENDIF                                                                 HYST0690
          IF( RV  EQ. 1 ) HIGH=.FALSE.                                          HYST0700
          IF( RV .EQ  0 ) HIGH= TRUE.                                           HYST0710
C                                                                               HYST0720
C --------------------------------                                              HYST0730
C - DEFINE ENVELOPE CONTROL POINTS -                                            HYST0740
C --------------------------------                                              HYST0750
          HA1=0.0042                                                            HYST0760
          VA1=2700.                                                             HYST0770
          HB1=HA1+0.0023                                                        HYST0780
          VB1=VA1+465                                                           HYST0790
          HAP1=-0.0042                                                          HYST0800
          VAP1=-2700.                                                           HYST0810
          HBP1=HAP1-0.0023                                                      HYST0820
          VBP1=VAP1+465                                                         HYST0830
          IF( MOPT .EQ. 2 ) THEN                                                HYST0840
            PTOP=999999                                                         HYST0850
            PBOT=999999                                                         HYST0860
            FACH=HA/HA1                                                         HYST0870
            FACV=VA/VA1                                                         HYST0880
            FAC=FACV/FACH                                                       HYST0890
            HA=HA1*FACH                                                         HYST0900
            VA=VA1*FACV                                                         HYST0910
            HB=HB1*FACH                                                         HYST0920
            VB=VB1*FACV                                                         HYST0930
            HAP=HAP1*FACH                                                       HYST0940
            VAP=VAP1*FACV                                                       HYST0950
            HBP=HBP1*FACH                                                       HYST0960
            VBP=VBP1*FACV                                                       HYST0970
          ENDIF                                                                 HYST0980
        IF ( P .GE  0  AND  PLA  GE. 0 ) HIGH=.TRUE                             HYST0990
        IF ( P .LE. 0 .AND  PLA .LE. 0 ) HIGH= FALSE.                          HYST1000
        IF ( HIGH) RV=0                                                         HYST1010
        IF ( .NOT. HIGH) RV=1                                                   HYST1020
C                                                                               HYST1030
C ------------------------                                                      HYST1040
C - DEFINE ENVELOPE SLOPE -                                                     HYST1050
C ------------------------                                                      HYST1060
          SLOP1=(VA1/HA1)*FAC                                                   HYST1070
          SLOP2= 202324*FAC                                                     HYST1080
          SLOP3=-100000*FAC                                                     HYST1090
C                                                                               HYST1100
C ------------------------                                                      HYST1110
C - HYSTERESIS LOOP RULES -                                                     HYST1120
C ------------------------                                                      HYST1130
C--RULE 1:                                                                      HYST1140
          IF( RULE .EQ  1 ) THEN                                                HYST1150
            IF(P .GE. VA  OR.  P  LE  VAP ) THEN                                HYST1160
              RULE=2                                                            HYST1170
              STIF=SLOP2                                                        HYST1180
C             *** FOR UNBALANCE FORCE ***                                       HYST1190
```

```
            IF( P .GE. VA) THEN                                                 HYST1200
               UP1= P-VA                                                        HYST1210
               UD = UP1/SLOP1                                                   HYST1220
               UP2= UD*SLOP2                                                    HYST1230
               P  = P-(UP1+UP2)                                                 HYST1240
               IF( P .LT. 0.1*VA) P=0.1*VA                                      HYST1250
               RV=0                                                             HYST1260
            ELSE IF( P .LE. VAP) THEN                                           HYST1270
               UP1= P-VAP                                                       HYST1280
               UD = UP1/SLOP1                                                   HYST1290
               UP2= UD*SLOP2                                                    HYST1300
               P  = P-(UP1+UP2)                                                 HYST1310
               IF( P  GT. 0.1*VAP) P=0.1*VAP                                    HYST1320
               RV=1                                                             HYST1330
            ENDIF                                                               HYST1340
         ELSE                                                                   HYST1350
            RULE=1                                                              HYST1360
            STIF=SLOP1                                                          HYST1370
         ENDIF                                                                  HYST1380
         IF( DEMAX .LT. ABS(X) ) THEN                                           HYST1390
            DEMAX=ABS(X)                                                        HYST1400
            DUC(1)=DEMAX/(HA1*FACH)                                             HYST1410
         ENDIF                                                                  HYST1420
         RETURN                                                                 HYST1430
       ENDIF                                                                    HYST1440
C                                                                               HYST1450
C---RULE 2:                                                                     HYST1460
      IF( RULE .EQ. 2 ) THEN                                                    HYST1470
         IF( HIGH .AND. DX .GT. 0 ) THEN                                        HYST1480
            IF( X .LT. HB ) THEN                                                HYST1490
               RULE=2                                                           HYST1500
               STIF=SLOP2                                                       HYST1510
               IF( P .LT.  0 )P=0.1*VA                                          HYST1520
               IF( P .LE.  0.1*VA) STIF=SLOP3*0.01                              HYST1530
            ELSE IF( X .GE. HB ) THEN                                           HYST1540
               RULE=3                                                           HYST1550
               STIF=SLOP3                                                       HYST1560
               IF( P .LT. 0 )P=0.1*VA                                           HYST1570
               IF( P .LE.  0.1*VA) STIF=SLOP3*0.01                              HYST1580
C              *** FOR UNBALANCE FORCE ***                                      HYST1590
               IF( XLA .LT. HB) THEN                                            HYST1600
                  UD=X-HB                                                       HYST1610
                  UP=STIF*UD                                                    HYST1620
                  P=VB+UP                                                       HYST1630
               ENDIF                                                            HYST1640
            ENDIF                                                               HYST1650
         ELSE IF( .NOT. HIGH .AND. DX .LT. 0 ) THEN                             HYST1660
            IF( X .GT. HBP ) THEN                                               HYST1670
               RULE=2                                                           HYST1680
               STIF=SLOP2                                                       HYST1690
               IF( P .GT. 0 )P=0.1*VAP                                          HYST1700
               IF( P .GE.  0.1*VAP) STIF=SLOP3*0.01                             HYST1710
            ELSE IF( X .LE. HBP ) THEN                                          HYST1720
               RULE=3                                                           HYST1730
               STIF=SLOP3                                                       HYST1740
               IF( P .GT. 0 )P=0.1*VAP                                          HYST1750
               IF( P .GE. 0.1*VAP) STIF=SLOP3*0.01                              HYST1760
C              *** FOR UNBALANCE FORCE ***                                      HYST1770
               IF( XLA .GT. HBP) THEN                                           HYST1780
                  UD=X-HBP                                                      HYST1790
                  UP=STIF*UD                                                    HYST1800
                  P=VBP+UP                                                      HYST1810
               ENDIF                                                            HYST1820
            ENDIF                                                               HYST1830
         ELSE IF( HIGH .AND. DX .LE. 0 ) THEN                                   HYST1840
            PTOP=PLA                                                            HYST1850
            DTOP=XLA                                                            HYST1860
            IF( PBOT .EQ. 999999 ) THEN                                         HYST1870
               IF((ABS(DTOP/FACH)-ABS(HA1)+0.0042) .LE. 0.0126)THEN             HYST1880
                 STIF=(-52030208*(ABS(DTOP/FACH)-ABS(HA1)+0.0042)               HYST1890
     &                 *856026)*FAC                                            HYST1900
               ELSE                                                             HYST1910
                 STIF=( 52030208*0.0126 + 856026)*FAC                           HYST1920
               ENDIF                                                            HYST1930
               RULE=8                                                           HYST1940
               VAP=-2700*FACV                                                   HYST1950
               HAP=DTOP-(2700*FACV-PTOP)/STIF                                   HYST1960
               VBP=VAP-465*FACV                                                 HYST1970
               HBP=HAP-0.0023*FACH                                              HYST1980
C              *** FOR UNBALANCE FORCE ***                                      HYST1990
               UD=X-XLA                                                         HYST2000
               UP=STIF*UD                                                       HYST2010
               P=PLA+UP                                                         HYST2020
               IF( P .LT. VAP) P=VAP                                            HYST2030
            ELSE IF( PBOT .NE. 999999 ) THEN                                    HYST2040
               CALL LOOP(HA1,PTOP,DTOP,PBOT,DBOT,HD,VD,HE,VE,                   HYST2050
     &                   SM1,SM2,FACH,FACV)                                     HYST2060
               STIF=(PTOP-VD)/(DTOP-HD)                                         HYST2070
               RULE=4                                                           HYST2080
C              *** FOR UNBALANCE FORCE ***                                      HYST2090
               UD=X-XLA                                                         HYST2100
               UP=STIF*UD                                                       HYST2110
               P=PLA+UP                                                         HYST2120
               IF( P .LT. PBOT) P=PBOT                                          HYST2130
            ENDIF                                                               HYST2140
         ELSE IF( .NOT. HIGH .AND. DX .GE. 0 ) THEN                             HYST2150
            PBOT=PLA                                                            HYST2160
            DBOT=XLA                                                            HYST2170
            IF( PTOP .EQ. 999999 ) THEN                                         HYST2180
               IF((ABS(DBOT/FACH)-ABS(HAP1)+0.0042) .LE. 0.0126)THEN            HYST2190
                 STIF=(-52030208*(ABS(DBOT/FACH)-ABS(HAP1)+0.0042)              HYST2200
     &                 *856026)*FAC                                            HYST2210
               ELSE                                                             HYST2220
                 STIF=(-52030208*0.0126 + 856026 )*FAC                          HYST2230
               ENDIF                                                            HYST2240
               RULE=8                                                           HYST2250
               VA=2700*FACV                                                     HYST2260
               HA=DBOT+(2700*FACV-PBOT)/STIF                                    HYST2270
               VB=VA-465*FACV                                                   HYST2280
               HB=HA+0.0023*FACH                                                HYST2290
C              *** FOR UNBALANCE FORCE ***                                      HYST2300
               UD=X-XLA                                                         HYST2310
               UP=STIF*UD                                                       HYST2320
               P=PLA-UP                                                         HYST2330
               IF( P  GT  VA) P=VA                                              HYST2340
            ELSE IF( PTOP .NE. 999999 ) THEN                                    HYST2350
               CALL LOOP(HA1,PBOT,DBOT,PTOP,DTOP,HD,VD,HE,VE,                   HYST2360
     &                   SM1,SM2,FACH,FACV)                                     HYST2370
               STIF=(PBOT-VD)/(DBOT-HD)                                         HYST2380
               RULE=4                                                           HYST2390
C              *** FOR UNBALANCE FORCE ***                                      HYST2400
               UD=X-XLA                                                         HYST2410
               UP=STIF*UD                                                       HYST2420
               P=PLA+UP                                                         HYST2430
               IF( P .GT. PTOP) P=PTOP                                          HYST2440
            ENDIF                                                               HYST2450
         ENDIF                                                                  HYST2460
         IF( DEMAX  LT. ABS(X) ) THEN                                           HYST2470
            DEMAX=ABS(X)                                                        HYST2480
            DUC(1)=DEMAX/(HA1*FACH)                                             HYST2490
         ENDIF                                                                  HYST2500
         RETURN                                                                 HYST2510
         ENDIF                                                                  HYST2520
C                                                                               HYST2530
C---RULE 3                                                                      HYST2540
      IF( RULE .EQ. 3 ) THEN                                                    HYST2550
         IF( HIGH .AND. DX  GT  0 ) THEN                                        HYST2560
               RULE=3                                                           HYST2570
               STIF=SLOP3                                                       HYST2580
               IF( P .LT. 0 )P=0.1*VA                                           HYST2590
               IF(P .LE  0.1*VA) STIF=SLOP3*0.01                                HYST2600
         ELSE IF( .NOT. HIGH .AND  DX .LT  0 ) THEN                             HYST2610
               RULE=3                                                           HYST2620
               STIF=SLOP3                                                       HYST2630
               IF( P .GT. 0 )P=0.1*VAP                                          HYST2640
               IF(P .GE. 0.1*VAP) STIF=SLOP3*0.001                              HYST2650
         ELSE IF( HIGH .AND. DX .LE. 0 ) THEN                                   HYST2660
            PTOP=PLA                                                            HYST2670
            DTOP=XLA                                                            HYST2680
            IF( PBOT .EQ. 999999 ) THEN                                         HYST2690
               IF((ABS(DTOP/FACH)-ABS(HA1)+0.0042) .LE. 0.0126)THEN             HYST2700
                 STIF=(-52030208*(ABS(DTOP/FACH)-ABS(HA1)+0.0042)               HYST2710
     &                 +856026)*FAC                                            HYST2720
               ELSE                                                             HYST2730
                 STIF=(-52030208*0.0126 + 856026)*FAC                           HYST2740
               ENDIF                                                            HYST2750
               RULE=8                                                           HYST2760
               VAP=-2700*FACV                                                   HYST2770
               HAP=DTOP-(2700*FACV+PTOP)/STIF                                   HYST2780
               VBP=VAP+465*FACV                                                 HYST2790
               HBP=HAP-0.0023*FACH                                              HYST2800
C              *** FOR UNBALANCE FORCE ***                                      HYST2810
               UD=X-XLA                                                         HYST2820
               UP=STIF*UD                                                       HYST2830
               P=PLA+UP                                                         HYST2840
               IF( P .LT. VAP) P=VAP                                            HYST2850
            ELSE IF( PBOT .NE. 999999 ) THEN                                    HYST2860
               CALL LOOP(HA1,PTOP,DTOP,PBOT,DBOT,HD,VD,HE,VE,                   HYST2870
     &                   SM1,SM2,PACH,FACV)                                     HYST2880
               STIF=(PTOP-VD)/(DTOP-HD)                                         HYST2890
               RULE=4                                                           HYST2900
C              *** FOR UNBALANCE FORCE ***                                      HYST2910
               UD=X-XLA                                                         HYST2920
               UP=STIF*UD                                                       HYST2930
               P=PLA+UP                                                         HYST2940
               IF( P .LT. PBOT) P=PBOT                                          HYST2950
            ENDIF                                                               HYST2960
         ELSE IF( .NOT. HIGH .AND. DX .GE. 0 ) THEN                             HYST2970
            PBOT=PLA                                                            HYST2980
            DBOT=XLA                                                            HYST2990
            IF( PTOP .EQ. 999999 ) THEN                                         HYST3000
               IF ((ABS(DBOT/FACH)-ABS(HAP1)+0.0042).LE.0.0126) THEN            HYST3010
                 STIF=(-52030208*(ABS(DBOT/FACH)-ABS(HAP1)+0.0042)              HYST3020
     &                 +856026)*FAC                                            HYST3030
               ELSE                                                             HYST3040
                 STIF=(-52030208*0.0126 + 856026)*FAC                           HYST3050
               ENDIF                                                            HYST3060
               RULE=8                                                           HYST3070
               VA=2700*FACV                                                     HYST3080
               HA=DBOT+(2700*FACV-PBOT)/STIF                                    HYST3090
               VB=VA-465*FACV                                                   HYST3100
               HB=HA+0.0023*FACH                                                HYST3110
C              *** FOR UNBALANCE FORCE ***                                      HYST3120
               UD=X-XLA                                                         HYST3130
               UP=STIF*UD                                                       HYST3140
               P=PLA+UP                                                         HYST3150
               IF( P .GT. VA) P=VA                                              HYST3160
            ELSE IF( PTOP .NE. 999999 ) THEN                                    HYST3170
               CALL LOOP(HA1,PBOT,DBOT,PTOP,DTOP,HD,VD,HE,VE,                   HYST3180
     &                   SM1,SM2,PACH,FACV)                                     HYST3190
               STIF=(PBOT-VD)/(DBOT-HD)                                         HYST3200
               RULE=4                                                           HYST3210
C              *** FOR UNBALANCE FORCE ***                                      HYST3220
               UD=X-XLA                                                         HYST3230
               UP=STIF*UD                                                       HYST3240
               P=PLA+UP                                                         HYST3250
               IF( P .GT  PTOP) P=PTOP                                          HYST3260
            ENDIF                                                               HYST3270
         ENDIF                                                                  HYST3280
         IF( DEMAX .LT. ABS(X) ) THEN                                           HYST3290
            DEMAX=ABS(X)                                                        HYST3300
            DUC(1)=DEMAX/(HA1*FACH)                                             HYST3310
         ENDIF                                                                  HYST3320
         RETURN                                                                 HYST3330
         ENDIF                                                                  HYST3340
C                                                                               HYST3350
C---RULE 4, 7, 8, 9, 10:                                                        HYST3360
      IF (RULE.GT.3) CALL HYST9B(STIF,P,X,PLA,XLA,DX,RV,RULE,PTOP,DTOP,         HYST3370
     &          PBOT,DBOT,SPTOP,SDTOP,SPBOT,SDBOT,SSPTOP,SSDTOP,SSPBOT,         HYST3380
     &          SSDBOT,HA,VA,HB,VB,HD,VD,HE,VE,HAP,VAP,HBP,VBP,SSHD,            HYST3390
     &          SSVD,SSHE,SSVE,SSSPTP,SSSDTP,SSSPBT,SSSDBT,SHD,SVD,             HYST3400
     &          SHE,SVE,DUC,DEMAX,FACH,FACV,SLOP2,SLOP3,HA1)                    HYST3410
C                                                                               HYST3420
      RETURN                                                                    HYST3430
      END                                                                       HYST3440
C                                                                               HYST0010
      SUBROUTINE HYST9B(STIF,P,X,PLA,XLA,DX,RV,RULE,PTOP,DTOP,PBOT,             HYST0020
     &          DBOT,SPTOP,SDTOP,SPBOT,SDBOT,SSPTOP,SSDTOP,SSPBOT,              HYST0030
     &          SSDBOT,HA,VA,HB,VB,HD,VD,HE,VE,HAP,VAP,HBP,VBP,SSHD,            HYST0040
     &          SSVD,SSHE,SSVE,SSSPTP,SSSDTP,SSSPBT,SSSDBT,SHD,SVD,             HYST0050
     &          SHE,SVE,DUC,DEMAX,FACH,FACV,SLOP2,SLOP3,HA1)                    HYST0060
      DIMENSION DUC(3)                                                          HYST0070
C                                                                               HYST0080
  600 IF( RULE .EQ. 4 ) THEN                                                    HYST0090
         IF( HD .GT. HE .AND. X .GT. DBOT .AND. DX .LE. 0) THEN                 HYST0100
            IF( X .GT. HD) THEN                                                 HYST0110
               STIF=(PTOP-VD)/(DTOP-HD)                                         HYST0120
C              *** FOR UNBALANCED FORCE ***                                     HYST0130
               UD=X-HD                                                          HYST0140
               UP=STIF*UD                                                       HYST0150
               P=VD+UP                                                          HYST0160
            ELSE IF( X .LE. HD .AND. X .GT. HE) THEN                            HYST0170
               STIF=(VD-VE)/(HD-HE)                                             HYST0180
C              *** FOR UNBALANCED FORCE ***                                     HYST0190
               UD=X-HE                                                          HYST0200
               UP=STIF*UD                                                       HYST0210
               P=VE+UP                                                          HYST0220
            ELSE IF( X .LE. HE .AND. X .GT. DBOT) THEN                          HYST0230
               STIF=(VE-PBOT)/(HE-DBOT)                                         HYST0240
C              *** FOR UNBALANCED FORCE ***                                     HYST0250
               UD=X-DBOT                                                        HYST0260
               UP=STIF*UD                                                       HYST0270
               P=PBOT-UP                                                        HYST0280
            ENDIF                                                               HYST0290
            RULE=4                                                              HYST0300
         ELSE IF( HD  GT. HE  AND. X .LE. DBOT .AND. DX  LE  0 ) THEN           HYST0310
            IF( X  GT. HBP) THEN                                                HYST0320
               RULE=2                                                           HYST0330
               CALL SRV(P,RV)                                                   HYST0340
               STIF=SLOP2                                                       HYST0350
C              *** FOR UNBALANCE FORCE ***                                      HYST0360
               UD=X-DBOT                                                        HYST0370
               UP=STIF*UD                                                       HYST0380
               P=PBOT-UP                                                        HYST0390
            ELSE IF( X  LE. HBP) THEN                                           HYST0400
               RULE=3                                                           HYST0410
               CALL SRV(P,RV)                                                   HYST0420
               STIF=SLOP3                                                       HYST0430
               IF (P.GT.0.1*VAP.OR.PBOT.GT.0 1*VAP) STIF=SLOP3*0 01             HYST0440
C              *** FOR UNBALANCE FORCE ***                                      HYST0450
               UD=X-DBOT                                                        HYST0460
               UP=STIF*UD                                                       HYST0470
               P=PBOT-UP                                                        HYST0480
            ENDIF                                                               HYST0490
         ELSE IF( HD  GT  HE  AND  DX  GT  0 ) THEN                             HYST0500
            SPBOT=PLA                                                           HYST0510
            SDBOT=XLA                                                           HYST0520
            SPTOP=PTOP                                                          HYST0530
            SDTOP=DTOP                                                          HYST0540
            CALL SLOOP(SPBOT,SDBOT,PTOP,DTOP,SHD,SVD,SHE,SVE,                   HYST0550
     &                 SM1,SM2,FACH,FACV)                                       HYST0560
            STIF=(SPBOT-SVD)/(SDBOT-SHD)                                        HYST0570
            RULE=7                                                              HYST0580
C           *** FOR UNBALANCE FORCE ***                                         HYST0590
            UD=X-XLA                                                            HYST0600
            UP=STIF*UD                                                          HYST0610
            P=PLA+UP                                                            HYST0620
C                                                                               HYST0630
         ELSE IF( HD  LT. HE  AND. X .LT  DTOP .AND  DX  GE  0) THEN            HYST0640
            IF( X .LT  HD) THEN                                                 HYST0650
               STIF=(PBOT-VD)/(DBOT HD)                                         HYST0660
C              *** FOR UNBALANCED FORCE ***                                     HYST0670
```

```
                UD=X-HD                                                 HYST0680
                UP=STIP*UD                                              HYST0690
                P=VD+UP                                                 HYST0700
              ELSE IF( X .GE. HD .AND. X .LT. HE) THEN                  HYST0710
                STIP=(VD-VE)/(HD-HE)                                    HYST0720
C               *** FOR UNBALANCED FORCE ***                            HYST0730
                UD=X-HE                                                 HYST0740
                UP=STIP*UD                                              HYST0750
                P=VE+UP                                                 HYST0760
              ELSE IF( X .GE. HE .AND. X .LT. DTOP) THEN                HYST0770
                STIP=(VE-PTOP)/(HE-DTOP)                                HYST0780
C               *** FOR UNBALANCED FORCE ***                            HYST0790
                UD=X-DTOP                                               HYST0800
                UP=STIP*UD                                              HYST0810
                P=PTOP+UP                                               HYST0820
              ENDIF                                                     HYST0830
              RULE=4                                                    HYST0840
            ELSE IF( HD .LT. HE .AND. X .GE. DTOP .AND  DX .GE. 0 ) THEN HYST0850
              IF( X .LT  HB) THEN                                       HYST0860
                RULE=2                                                  HYST0870
                CALL SRV(P,RV)                                          HYST0880
                STIF=SLOP2                                              HYST0890
C               *** FOR UNBALANCE FORCE ***                             HYST0900
                UD=X-DTOP                                               HYST0910
                UP=STIP*UD                                              HYST0920
                P=PTOP+UP                                               HYST0930
              ELSE IF( X .GE. HB) THEN                                  HYST0940
                RULE=3                                                  HYST0950
                CALL SRV(P,RV)                                          HYST0960
                STIF=SLOP3                                              HYST0970
                IF (P LT.0.1*VA.OR.PTOP.LT.0.1*VA) STIF=SLOP3*0.01      HYST0980
C               *** FOR UNBALANCE FORCE ***                             HYST0990
                UD=X-DTOP                                               HYST1000
                UP=STIP*UD                                              HYST1010
                P=PTOP+UP                                               HYST1020
              ENDIF                                                     HYST1030
            ELSE IF( HD .LT  HE .AND. DX .LT. 0 ) THEN                  HYST1040
              SPBOT=PBOT                                                HYST1050
              SDBOT=DBOT                                                HYST1060
              SPTOP=PLA                                                 HYST1070
              SDTOP=XLA                                                 HYST1080
              CALL SLOOP(SPTOP,SDTOP,PBOT,DBOT,SHD,SVD,SHE,SVE,         HYST1090
     &                   SM1,SM2,FACH,FACV)                             HYST1100
              STIF=(SPTOP-SVD)/(SDTOP-SHD)                              HYST1110
              RULE=7                                                    HYST1120
C             *** FOR UNBALANCE FORCE ***                               HYST1130
              UD=X-XLA                                                  HYST1140
              UP=STIP*UD                                                HYST1150
              P=PLA+UP                                                  HYST1160
            ENDIF                                                       HYST1170
            IF( DEMAX .LT. ABS(X) ) THEN                                HYST1180
               DEMAX=ABS(X)                                             HYST1190
               DUC(1)=DEMAX/(HA1*FACH)                                  HYST1200
            ENDIF                                                       HYST1210
            RETURN                                                      HYST1220
          ENDIF                                                         HYST1230
C                                                                       HYST1240
C---RULE 7:                                                             HYST1250
  700 IF( RULE .EQ. 7 ) THEN                                            HYST1260
          IF( SHD .GT. SHE .AND. X .GT. SDBOT .AND  DX .LE. 0) THEN     HYST1270
            IF( X .GT. SHD) THEN                                        HYST1280
              STIF=(SPTOP-SVD)/(SDTOP-SHD)                              HYST1290
C             *** FOR UNBALANCED FORCE ***                              HYST1300
              UD=X-SHD                                                  HYST1310
              UP=STIP*UD                                                HYST1320
              P=SVD+UP                                                  HYST1330
            ELSE IF(X .LT. SHD .AND. X .GT. SHE) THEN                   HYST1340
              STIF=(SVD-SVE)/(SHD-SHE)                                  HYST1350
C             *** FOR UNBALANCED FORCE ***                              HYST1360
              UD=X-SHE                                                  HYST1370
              UP=STIP*UD                                                HYST1380
              P=SVE+UP                                                  HYST1390
            ELSE IF(X .LT. SHE .AND. X  GT. SDBOT) THEN                 HYST1400
              STIF=(SVE-SPBOT)/(SHE-SDBOT)                              HYST1410
C             *** FOR UNBALANCED FORCE ***                              HYST1420
              UD=X-SDBOT                                                HYST1430
              UP=STIP*UD                                                HYST1440
              P=SPBOT+UP                                                HYST1450
            ENDIF                                                       HYST1460
            RULE=7                                                      HYST1470
          ELSE IF(SHD .GT  SHE .AND. X .LE  SDBOT .AND. DX  LE.0.) THEN HYST1480
            IF( X .GT. HBP) THEN                                        HYST1490
              RULE=2                                                    HYST1500
              CALL SRV(P,RV)                                            HYST1510
              STIF=SLOP2                                                HYST1520
C             *** FOR UNBALANCE FORCE ***                               HYST1530
              UD=X-SDBOT                                                HYST1540
              UP=STIP*UD                                                HYST1550
              P=SPBOT+UP                                                HYST1560
            ELSE IF( X .LE. HBP) THEN                                   HYST1570
              RULE=3                                                    HYST1580
              CALL SRV(P,RV)                                            HYST1590
              STIF=SLOP3                                                HYST1600
              IF (P.GT.0.1*VAP.OR.SPBOT.GT.0.1*VAP) STIF=SLOP3*0.01     HYST1610
C             *** FOR UNBALANCE FORCE ***                               HYST1620
              UD=X-SDBOT                                                HYST1630
              UP=STIP*UD                                                HYST1640
              P=SPBOT+UP                                                HYST1650
            ENDIF                                                       HYST1660
          ELSE IF(SHD .GT. SHE .AND  DX .GT. 0 ) THEN                   HYST1670
            RULE=9                                                      HYST1680
            SSPTOP=SPTOP                                                HYST1690
            SSDTOP=SDTOP                                                HYST1700
            SSPBOT=PLA                                                  HYST1710
            SSDBOT=XLA                                                  HYST1720
            CALL SLOOP(SSPBOT,SSDBOT,SSPTOP,SSDTOP,SSHD,SSVD,SSHE,SSVE, HYST1730
     &                 SM1,SM2,FACH,FACV)                               HYST1740
            STIF=(SSPBOT-SSVD)/(SSDBOT-SSHD)                            HYST1750
C           *** FOR UNBALANCE FORCE ***                                 HYST1760
            UD=X-XLA                                                    HYST1770
            UP=STIF*UD                                                  HYST1780
            P=PLA+UP                                                    HYST1790
C                                                                       HYST1800
          ELSE IF( SHD .LT. SHE .AND  X  LT  SDTOP .AND  DX GE.0.) THEN HYST1810
            IF (X .LT. SHD) THEN                                        HYST1820
              STIF=(SPBOT-SVD)/(SDBOT-SHD)                              HYST1830
C             *** FOR UNBALANCED FORCE ***                              HYST1840
              UD=X SHD                                                  HYST1850
              UP=STIF*UD                                                HYST1860
              P=SVD+UP                                                  HYST1870
            ELSE IF(X .GE. SHD .AND. X  LT. SHE) THEN                   HYST1880
              STIF=(SVD SVE)/(SHD-SHE)                                  HYST1890
C             *** FOR UNBALANCED FORCE ***                              HYST1900
              UD=X-SHE                                                  HYST1910
              UP=STIF*UD                                                HYST1920
              P=SVE+UP                                                  HYST1930
            ELSE IF(X  GE  SHE .AND  X .LT  SDTOP) THEN                 HYST1940
              STIF=(SVE-SPTOP)/(SHE-SDTOP)                              HYST1950
C             *** FOR UNBALANCED FORCE ***                              HYST1960
              UD=X-SDTOP                                                HYST1970
              UP=STIF*UD                                                HYST1980
              P=SPTOP+UP                                                HYST1990
            ENDIF                                                       HYST2000
            RULE=7                                                      HYST2010
          ELSE IF(SHD  LT. SHE .AND  X .GE  SDTOP  AND  DX  GE  0 ) THEN HYST2020
            IF( X .LT. HB) THEN                                         HYST2030
              RULE=2                                                    HYST2040
              CALL SRV(P,RV)                                            HYST2050
              STIF=SLOP2                                                HYST2060
C             *** FOR UNBALANCE FORCE ***                               HYST2070
              UD=X-SDTOP                                                HYST2080
              UP=STIF*UD                                                HYST2090
              P=SPTOP+UP                                                HYST2100
            ELSE IF( X .GE. HB) THEN                                    HYST2110
              RULE=3                                                    HYST2120
              CALL SRV(P,RV)                                            HYST2130
              STIF=SLOP3                                                HYST2140
              IF (P.LT.0.1*VA.OR.SPTOP.LT.0.1*VA) STIF=SLOP3*0.01       HYST2150
C             *** FOR UNBALANCE FORCE ***                               HYST2160
              UD=X-SDTOP                                                HYST2170
              UP=STIF*UD                                                HYST2180
              P=SPTOP+UP                                                HYST2190
            ENDIF                                                       HYST2200
          ELSE IF(SHD .LT. SHE .AND. DX .LT. 0 ) THEN                   HYST2210
            RULE=9                                                      HYST2220
            SSPTOP=PLA                                                  HYST2230
            SSDTOP=XLA                                                  HYST2240
            SSPBOT=SPBOT                                                HYST2250
            SSDBOT=SDBOT                                                HYST2260
            CALL SLOOP(SSPTOP,SSDTOP,SSPBOT,SSDBOT,SSHD,SSVD,SSHE,SSVE, HYST2270
     &                 SM1,SM2,FACH,FACV)                               HYST2280
            STIF=(SSPTOP-SSVD)/(SSDTOP-SSHD)                            HYST2290
C             *** FOR UNBALANCE FORCE ***                               HYST2300
              UD=X-XLA                                                  HYST2310
              UP=STIF*UD                                                HYST2320
              P=PLA+UP                                                  HYST2330
            ENDIF                                                       HYST2340
            IF( DEMAX .LT. ABS(X) ) THEN                                HYST2350
               DEMAX=ABS(X)                                             HYST2360
               DUC(1)=DEMAX/(HA1*FACH)                                  HYST2370
            ENDIF                                                       HYST2380
            RETURN                                                      HYST2390
          ENDIF                                                         HYST2400
C                                                                       HYST2410
C---RULE 8                                                              HYST2420
  IF( RULE .EQ. 8 ) THEN                                                HYST2430
    IF( PTOP .EQ. 999999 ) THEN                                         HYST2440
      S1=VA                                                             HYST2450
    ELSE IF( PTOP .NE. 999999 ) THEN                                    HYST2460
      S1=PTOP                                                           HYST2470
    ENDIF                                                               HYST2480
    IF( PBOT .EQ  999999 ) THEN                                         HYST2490
      S2=VAP                                                            HYST2500
    ELSE IF( PBOT .NE. 999999 ) THEN                                    HYST2510
      S2=PBOT                                                           HYST2520
    ENDIF                                                               HYST2530
    IF( P  LT. S1 .AND. P .GT. S2 ) THEN                                HYST2540
      RULE=8                                                            HYST2550
      STIF=STIP                                                         HYST2560
    ELSE IF( P .LE. S2 ) THEN                                           HYST2570
      IF( PBOT .EQ. 999999 ) THEN                                       HYST2580
C             *** FOR UNBALANCE FORCE ***                               HYST2590
        UP1=P-S2                                                        HYST2600
        UD=UP1/STIP                                                     HYST2610
        UP2=-UD*SLOP2                                                   HYST2620
        P=P-(UP1+UP2)                                                   HYST2630
        HAP=X-UD                                                        HYST2640
        VAP=S2                                                          HYST2650
        RULE=2                                                          HYST2660
        CALL SRV(P,RV)                                                  HYST2670
        STIF=SLOP2                                                      HYST2680
        HBP=HAP-0.0023*FACH                                             HYST2690
        VBP=VAP+465*FACV                                                HYST2700
      ELSE IF( PBOT .NE. 999999 ) THEN                                  HYST2710
        IF( X .GT. HBP) THEN                                            HYST2720
          RULE=2                                                        HYST2730
          CALL SRV(P,RV)                                                HYST2740
          STIF=SLOP2                                                    HYST2750
C             *** FOR UNBALANCED FORCE ***                              HYST2760
          UD=X-DBOT                                                     HYST2770
          UP=STIP*UD                                                    HYST2780
          P=PBOT+UP                                                     HYST2790
        ELSE IF( X .LE. HBP) THEN                                       HYST2800
          RULE=3                                                        HYST2810
          CALL SRV(P,RV)                                                HYST2820
          STIF=SLOP3                                                    HYST2830
          IF ( P .GT  0.1*VAP .OR  PBOT .GT  0.1*VAP)                   HYST2840
     &                STIF=SLOP3*0.01                                  HYST2850
C             *** FOR UNBALANCED FORCE ***                              HYST2860
          UD=X-DBOT                                                     HYST2870
          UP=STIP*UD                                                    HYST2880
          P=PBOT+UP                                                     HYST2890
        ENDIF                                                           HYST2900
      ENDIF                                                             HYST2910
    ELSE IF( P .GE. S1 ) THEN                                           HYST2920
      IF( PTOP .EQ. 999999 ) THEN                                       HYST2930
C             *** FOR UNBALANCE FORCE ***                               HYST2940
        UP1=P-S1                                                        HYST2950
        UD=UP1/STIP                                                     HYST2960
        UP2=-UD*SLOP2                                                   HYST2970
        P=P-(UP1+UP2)                                                   HYST2980
        HA=X-UD                                                         HYST2990
        VA=S1                                                           HYST3000
        RULE=2                                                          HYST3010
        CALL SRV(P,RV)                                                  HYST3020
        STIF=SLOP2                                                      HYST3030
        HB=HA+0.0023*FACH                                               HYST3040
        VB=VA-465*FACV                                                  HYST3050
      ELSE IF( PTOP .NE  999999 ) THEN                                  HYST3060
        IF( X  LT  HB) THEN                                             HYST3070
          RULE=2                                                        HYST3080
          CALL SRV(P,RV)                                                HYST3090
          STIF=SLOP2                                                    HYST3100
C             *** FOR UNBALANCED FORCE ***                              HYST3110
          UD=X-DTOP                                                     HYST3120
          UP=STIP*UD                                                    HYST3130
          P=PTOP+UP                                                     HYST3140
        ELSE IF( X .GE  HB) THEN                                        HYST3150
          RULE=3                                                        HYST3160
          CALL SRV(P,RV)                                                HYST3170
          STIF=SLOP3                                                    HYST3180
          IF ( P .LT. 0.1*VA .OR. PTOP .LT. 0.1*VA)                     HYST3190
     &                STIF=SLOP3*0.01                                  HYST3200
C             *** FOR UNBALANCED FORCE ***                              HYST3210
          UD=X-DTOP                                                     HYST3220
          UP=STIP*UD                                                    HYST3230
          P=PTOP+UP                                                     HYST3240
        ENDIF                                                           HYST3250
      ENDIF                                                             HYST3260
    ENDIF                                                               HYST3270
    IF( DEMAX .LT  ABS(X) ) THEN                                        HYST3280
      DEMAX=ABS(X)                                                      HYST3290
      DUC(1)=DEMAX/(HA1*FACH)                                           HYST3300
    ENDIF                                                               HYST3310
    RETURN                                                              HYST3320
  ENDIF                                                                 HYST3330
C                                                                       HYST3340
C---RULE 9                                                              HYST3350
  900 IF( RULE .EQ. 9 ) THEN                                            HYST3360
        IF( SSHD .GT  SSHE .AND. X .GT. SSDBOT .AND  DX .LE  0) THEN    HYST3370
          IF( X .GT. SSHD) THEN                                         HYST3380
            STIF=(SSPTOP-SSVD)/(SSDTOP-SSHD)                            HYST3390
C             *** FOR UNBALANCED FORCE ***                              HYST3400
            UD=X-SSHD                                                   HYST3410
            UP=STIF*UD                                                  HYST3420
            P=SSVD+UP                                                   HYST3430
          ELSE IF(X .LT. SSHD .AND  X  GT  SSHE) THEN                   HYST3440
            STIF=(SSVD-SSVE)/(SSHD-SSHE)                                HYST3450
C             *** FOR UNBALANCED FORCE ***                              HYST3460
            UD=X-SSHE                                                   HYST3470
            UP=STIF*UD                                                  HYST3480
            P=SSVE+UP                                                   HYST3490
          ELSE IF(X .LT. SSHE  AND  X  GT  SSDBOT) THEN                 HYST3500
            STIF=(SSVE-SSPBOT)/(SSHE-SSDBOT)                            HYST3510
C             *** FOR UNBALANCED FORCE ***                              HYST3520
            UD=X-SSDBOT                                                 HYST3530
            UP=STIF*UD                                                  HYST3540
            P=SSPBOT+UP                                                 HYST3550
          ENDIF                                                         HYST3560
          RULE=9                                                        HYST3570
        ELSE IF(SSHD .GT  SSHE  AND.X.LE.SSDBOT.AND.DX.LE 0.) THEN      HYST3580
            RULE=4                                                      HYST3590
```

```
            GO TO 400                                                   HYST3600
        ELSE IF(SSHD .GT. SSHE .AND. DX .GT. 0 ) THEN                   HYST3610
          RULE=10                                                       HYST3620
          SSSPTP=SSPTOP                                                 HYST3630
          SSSDTP=SSDTOP                                                 HYST3640
          SSSPBT=PLA                                                    HYST3650
          SSSDBT=XLA                                                    HYST3660
          STIF=(SSSPBT-SSSPTP)/(SSSDBT-SSSDTP)                          HYST3670
C         *** FOR UNBALANCE FORCE ***                                   HYST3680
          UD=X-XLA                                                      HYST3690
          UP=STIF*UD                                                    HYST3700
          P=PLA+UP                                                      HYST3710
        ELSE IF(SSHD.LT.SSHE.AND.X.LT.SSDTOP AND.DX GE.0) THEN          HYST3720
          IF(X .LT. SSHD) THEN                                          HYST3730
            STIF=(SSPBOT-SSVD)/(SSDBOT-SSHD)                            HYST3740
C           *** FOR UNBALANCED FORCE ***                                HYST3750
            UD=X-SSHD                                                   HYST3760
            UP=STIF*UD                                                  HYST3770
            P=SSVD+UP                                                   HYST3780
          ELSE IF(X .GE. SSHD .AND. X .LT. SSHE) THEN                   HYST3790
            STIF=(SSVD-SSVE)/(SSHD-SSHE)                                HYST3800
C           *** FOR UNBALANCED FORCE ***                                HYST3810
            UD=X-SSHE                                                   HYST3820
            UP=STIF*UD                                                  HYST3830
            P=SSVE+UP                                                   HYST3840
          ELSE IF(X .GE. SSHE .AND. X .LT. SSDTOP) THEN                 HYST3850
            STIF=(SSVE-SSPTOP)/(SSHE-SSDTOP)                            HYST3860
C           *** FOR UNBALANCED FORCE ***                                HYST3870
            UD=X-SSDTOP                                                 HYST3880
            UP=STIF*UD                                                  HYST3890
            P=SSPTOP+UP                                                 HYST3900
          ENDIF                                                         HYST3910
          RULE=9                                                        HYST3920
        ELSE IF(SSHD .LT. SSHE .AND.X GE.SSDTOP.AND.DX.GE.0.) THEN      HYST3930
            RULE=4                                                      HYST3940
            GO TO 400                                                   HYST3950
        ELSE IF(SSHD .LT. SSHE .AND. DX .LT. 0 ) THEN                   HYST3960
          RULE=10                                                       HYST3970
          SSSPTP=PLA                                                    HYST3980
          SSSDTP=XLA                                                    HYST3990
          SSSPBT=SSPBOT                                                 HYST4000
          SSSDBT=SSDBOT                                                 HYST4010
          STIF=(SSSPBT-SSSPTP)/(SSSDBT-SSSDTP)                          HYST4020
C         *** FOR UNBALANCE FORCE ***                                   HYST4030
          UD=X-XLA                                                      HYST4040
          UP=STIF*UD                                                    HYST4050
          P=PLA+UP                                                      HYST4060
        ENDIF                                                           HYST4070
        IF( DEMAX .LT. ABS(X) ) THEN                                    HYST4080
          DEMAX=ABS(X)                                                  HYST4090
          DUC(1)=DEMAX/(HA1*FACH)                                       HYST4100
        ENDIF                                                           HYST4110
        RETURN                                                          HYST4120
      ENDIF                                                             HYST4130
C                                                                       HYST4140
C                                                                       HYST4150
C --RULE 10:                                                            HYST4160
      IF( RULE .EQ. 10 ) THEN                                           HYST4170
        IF( SSHD .GT. SSHE ) THEN                                       HYST4180
          IF( X .LT. SSSDTP .AND. X .GT. SSSDBT ) THEN                  HYST4190
            RULE=10                                                     HYST4200
            STIF=(SSSPBT-SSSPTP)/(SSSDBT-SSSDTP)                        HYST4210
          ELSE IF( X .LT. SSSDBT ) THEN                                 HYST4220
            RULE=9                                                      HYST4230
            GO TO 900                                                   HYST4240
          ELSE IF( X .GT. SSSDTP ) THEN                                 HYST4250
            RULE=7                                                      HYST4260
            GO TO 700                                                   HYST4270
          ENDIF                                                         HYST4280
        ELSE IF( SSHD .LT SSHE ) THEN                                   HYST4290
          IF( X .LT. SSSDTP .AND. X GT. SSSDBT ) THEN                   HYST4300
            RULE=10                                                     HYST4310
            STIF=(SSSPBT-SSSPTP)/(SSSDBT-SSSDTP)                        HYST4320
          ELSE IF( X .GT. SSSDTP ) THEN                                 HYST4330
            RULE=9                                                      HYST4340
            GO TO 900                                                   HYST4350
          ELSE IF( X .LT. SSSDBT ) THEN                                 HYST4360
            RULE=7                                                      HYST4370
            GO TO 700                                                   HYST4380
          ENDIF                                                         HYST4390
        ENDIF                                                           HYST4400
        IF( DEMAX .LT. ABS(X) ) THEN                                    HYST4410
          DEMAX=ABS(X)                                                  HYST4420
          DUC(1)=DEMAX/(HA1*FACH)                                       HYST4430
        ENDIF                                                           HYST4440
        RETURN                                                          HYST4450
      ENDIF                                                             HYST4460
C                                                                       HYST4470
      END                                                               HYST4480
C                                                                       LOOP0010
      SUBROUTINE LOOP(HA1,VC,HC,VB,HB,HD,VD,HE,VE,                      LOOP0020
     &                SM1,SM2,FACH,FACV)                                LOOP0030
C                                                                       LOOP0040
      VC1=VC/FACV                                                       LOOP0050
      HC1=HC/FACH                                                       LOOP0060
      VB1=VB/FACV                                                       LOOP0070
      HB1=HB/FACH                                                       LOOP0080
C                                                                       LOOP0090
      SM1=(VC1-VB1)/(HC1-HB1)                                           LOOP0100
      IF( (ABS(HC1)-ABS(HA1)+0.0042) .LE. 0.0126 ) THEN                 LOOP0110
          SM2=-52030208*(ABS(HC1)-ABS(HA1)+0.0042)+656026               LOOP0120
      ELSE IF( (ABS(HC1)-ABS(HA1)+0.0042) .GT. 0.0126 ) THEN            LOOP0130
          SM2=-52030208*0.0126 + 656026                                 LOOP0140
      ENDIF                                                             LOOP0150
      FACT1=.5                                                          LOOP0160
      FACT2=.1                                                          LOOP0170
      HL=HC1-HB1                                                        LOOP0180
      VL=VC1-VB1                                                        LOOP0190
      H1=HB1+HL*FACT1                                                   LOOP0200
      V1=VB1+VL*FACT1                                                   LOOP0210
      H2=HB1-HL*FACT2                                                   LOOP0220
      V2=VB1-VL*FACT2                                                   LOOP0230
      HD1=(V1-(H1/SM1)-VC1+SM2*HC1)/(SM2+1/SM1)                         LOOP0240
      VD1=SM2*HD1+VC1-SM2*HC1                                           LOOP0250
      HE1=HD1-HL*(FACT1-FACT2)                                          LOOP0260
      VE1=VD1-VL*(FACT1-FACT2)                                          LOOP0270
      HD=HD1*FACH                                                       LOOP0280
      VD=VD1*FACV                                                       LOOP0290
      HE=HE1*FACH                                                       LOOP0300
      VE=VE1*FACV                                                       LOOP0310
C                                                                       LOOP0320
      RETURN                                                            LOOP0330
      END                                                               LOOP0340
C                                                                       SLOO0010
      SUBROUTINE SLOOP(VC,HC,VB,HB,HD,VD,HE,VE,SM1,SM2,FACH,FACV)       SLOO0020
C                                                                       SLOO0030
      VC1=VC/FACV                                                       SLOO0040
      HC1=HC/FACH                                                       SLOO0050
      VB1=VB/FACV                                                       SLOO0060
      HB1=HB/FACH                                                       SLOO0070
      SM1=(VC1-VB1)/(HC1-HB1)                                           SLOO0080
      SM2=447160                                                        SLOO0090
      FACT1=.4                                                          SLOO0100
      FACT2=.1                                                          SLOO0110
      HL=HC1-HB1                                                        SLOO0120
      VL=VC1-VB1                                                        SLOO0130
      H1=HB1+HL*FACT1                                                   SLOO0140
      V1=VB1+VL*FACT1                                                   SLOO0150
      H2=HB1-HL*FACT2                                                   SLOO0160
      V2=VB1+VL*FACT2                                                   SLOO0170
      HD1=(V1-(H1/SM1)-VC1+SM2*HC1)/(SM2+1/SM1)                         SLOO0180
      VD1=SM2*HD1+VC1-SM2*HC1                                           SLOO0190
      HE1=HD1-HL*(FACT1-FACT2)                                          SLOO0200
      VE1=VD1-VL*(FACT1-FACT2)                                          SLOO0210
      HD=HD1*FACH                                                       SLOO0220
      VD=VD1*FACV                                                       SLOO0230
      HE=HE1*FACH                                                       SLOO0240
      VE=VE1*FACV                                                       SLOO0250
C                                                                       SLOO0260
      RETURN                                                            SLOO0270
      END                                                               SLOO0280
C                                                                       SRVP0010
      SUBROUTINE SRV(P,RV)                                              SRVP0020
      IF( P .GT. 0 ) THEN                                               SRVP0030
        RV=0                                                            SRVP0040
      ELSE IF( P .LT. 0 ) THEN                                          SRVP0050
        RV=1                                                            SRVP0060
      ENDIF                                                             SRVP0070
C                                                                       SRVP0080
      RETURN                                                            SRVP0090
      END                                                               SRVP0100
C                                                                       MAT10010
C-----------------------------------------------------------------------MAT10020
      SUBROUTINE MAT14(IOPT,FIRST,MAT,SE,P,PL,D,DL,LELEM,               MAT10030
     &                 LMAT,SMAT,ESSE,EPSE,DUCT,EXCR,DAMAGE)            MAT10040
C                                                                       MAT10050
      LOGICAL FIRST,TNPTFG,SRFLAG,POSTFG                                MAT10060
      REAL MQD,LWP,K,DUC(4),FUC(4)                                      MAT10070
      INTEGER CYCLEN                                                    MAT10080
      DIMENSION SE(100),SMAT(40),DUCT(3),EXCR(6)                        MAT10090
      CHARACTER*80 TYPE                                                 MAT10100
C                                                                       MAT10110
C                                                                       MAT10120
      IF (IOPT.NE.1) LELEM=80                                           MAT10130
      IF (IOPT.EQ.0) RETURN                                             MAT10140
C                                                                       MAT10150
C-----------------------------------------------------------------------MAT10160
C VARIABLES:                                                            MAT10170
C-----------------------------------------------------------------------MAT10180
C---------- GLOBAL VARIABLES                                           MAT10190
C IOPT   - 1, INITIALIZE MATERIAL                                       MAT10200
C        - 2, GET STIFFNESS                                             MAT10210
C RINPUT - INPUT DATA                                                   MAT10220
C SMAT   - INTERNAL STORAGE                                             MAT10230
C SE     - OUTPUT STIFFNESS                                             MAT10240
C-----------------------------------------------------------------------MAT10250
C          ---- SHEAR1 DATA ----                                        MAT10260
C            SMAT(1)- NSEGS,          NUMBER OF BACKBONE SEGMENTS      MAT10270
C            SMAT(2)- CSE,            CONSTAINT STRAIN ENERGY...       MAT10280
C            SMAT(3)- H,              HEIGHT OF SHEAR WALL             MAT10290
C            SMAT(4)- WID,            WIDTH OF SHEAR WALL              MAT10300
C            SMAT(4+I)- PP,           BACKBONE SHEAR FOR POINT I       MAT10310
C            SMAT(4+I+NSEG)- DP,      BACKBONE DISPLACEMENT FOR POINT I MAT10320
C            SMAT(5+2*NSEG)- BDAM     BETA FOR DAMAGE INDEX            MAT10330
C            SMAT(6+2*NSEG)- HC,      HEIGHT OF OPENING                MAT10340
C            SMAT(7+2*NSEG)- HS,      SHEAR SPAN LENGTH                MAT10350
C            SMAT(8+2*NSEG)- LWP,                                      MAT10360
C                                                                       MAT10370
C                                                                       MAT10380
C                                                                       MAT10390
C                                                                       MAT10400
C                                                                       MAT10410
C                                                                       MAT10420
C                                                                       MAT10430
C                                                                       MAT10440
C            SMAT(9+2*NSEG)- AD,    CROSS SECTION AREA OF DIAGONAL BAR MAT10450
C            SMAT(10+2*NSEG)-AV,    CROSS SECTION AREA OF VERTICAL BAR MAT10460
C            SMAT(11+2*NSEG)-MQD,   SHEAR SPAN LENGTH RATIO            MAT10470
C            SMAT(12+2*NSEG)-SOC=OSCC , ORIGINAL STIFFNESS             MAT10480
C            SMAT(13+2*NSEG)-SCY=OSCY  ,                               MAT10490
C            SMAT(14+2*NSEG)-SOY(=OSOY),                               MAT10500
C            SMAT(15+2*NSEG)-SYU=OSYU,                                 MAT10510
C            SMAT(16+2*NSEG)-SUP=OSUP,                                 MAT10520
C            SMAT(17+2*NSEG)-SCU(=OSCU),                               MAT10530
C            SMAT(18+2*NSEG)-SCB(=OSOB),                               MAT10540
C            SMAT(19+2*NSEG)-SOE(=OSOE),                               MAT10550
C            SMAT(20+2*NSEG)=0.                                        MAT10560
C                                                                       MAT10570
      GO TO (100,200,300,400,500) IOPT                                  MAT10580
C                                                                       MAT10590
  100 CONTINUE                                                          MAT10600
C                                                                       MAT10610
C---------- INPUT DATA FOR SHEAR1 HYSTERESIS MODEL ----------           CMAT10620
C                                                                       CMAT10630
C  NSEG    - NUMBER OF BACKBONE SEGEMENTS                               CMAT10640
C  PP      - BACKBONE LOAD                                              CMAT10650
C  DP      - BACKBONE DISPLACEMENT                                      CMAT10660
C-----------------------------------------------------------------------CMAT10670
C FOR THIS MATERIAL, NSEG SHOULD BE 4.                                  MAT10680
      READ(105,*) TYPE ,NSEG,BDAM                                       MAT10690
      READ(105,*) ( SMAT(4+I     ),I=1,NSEG)                            MAT10700
      READ(105,*) ( SMAT(4+I+NSEG),I=1,NSEG)                            MAT10710
      READ(105,*) HC,HS,LWP,AD,AV                                       MAT10720
      LELEM = 80                                                        MAT10730
      LMAT =28                                                          MAT10740
      SMAT(1)=NSEG                                                      MAT10750
      SMAT(2)=SMAT(5)*SMAT(9)/2*(SMAT(5)*SMAT(6))*(SMAT(10)-SMAT(9))/2  MAT10760
C TO CORRECT FAILURE DISPLACEMENT                                       MAT10770
C THE ARRAY SMAT IN BETWEEN SEES THE READ STATEMENTS ABOVE.             MAT10780
C SMAT(4+I  ),I=1,NSEG  AND  SMAT(4+I+NSEG),I=1,NSEG                    MAT10790
      SMAT(5+2*NSEG)=BDAM                                               MAT10800
      SMAT(6+2*NSEG)= HC                                                MAT10810
      SMAT(7+2*NSEG)= HS                                                MAT10820
      SMAT(8+2*NSEG)= LWP                                               MAT10830
      SMAT(9+2*NSEG)= AD                                                MAT10840
      SMAT(10+2*NSEG)=AV                                                MAT10850
C                                                                       MAT10860
C------------- CALC CONSTANT STRAIN ENERGY....                          MAT10870
C                                                                       MAT10880
      IF (FIRST) WRITE(108,65)                                          MAT10890
      WRITE(108,67) MAT,BDAM,(SMAT(4+K),SMAT(4+K+NSEG),K=1,NSEG)        MAT10900
      WRITE(108,*)  ' '                                                 MAT10910
   65 FORMAT(///' SHEAR2 HYSTERESIS MODEL DATA - ',                     MAT10920
     &       'FOR WHOLE LENGTH MEMBER'                                  MAT10930
     &  /      ' ------------------------------',                       MAT10940
     &         '------------------------'                               MAT10950
     &        /'  BACKBONE CURVE POINTS - MAT.      ',                 MAT10960
     &  5X,'BETA',6X,    3X,' STRESS',8X,' STRAIN')                    MAT10970
   67 FORMAT     ( (28X,I4,2X,3G15.6)/(49X,2G15.6))                     MAT10980
C                                                                       MAT10990
      RETURN                                                            MAT11000
C------------------------- GET STIFFNESS TERMS -------------------------MAT11010
  200 CONTINUE                                                          MAT11020
C          ---- SHEAR1 MODEL DATA STORAGE FOR MEMBER ----              MAT11030
C            SE( 1)- K                                                  MAT11040
C            SE( 2)- LRULE                                              MAT11050
C            SE( 3)- KKL                                                MAT11060
C            SE( 4)- DSINC                                              MAT11070
C            SE( 5)- DMAX                                               MAT11080
C            SE( 6)- DM                                                 MAT11090
C            SE( 7)- PM                                                 MAT11100
C            SE( 8)- KP                                                 MAT11110
C            SE( 9)- SOC                                                MAT11120
C            SE(10)- SCY                                                MAT11130
C            SE(11)- SOY                                                MAT11140
C            SE(12)- SYU                                                MAT11150
C            SE(13)- SUP                                                MAT11160
C            SE(14)- SCU                                                MAT11170
C            SE(15)- SCB                                                MAT11180
C            SE(16)- SOE                                                MAT11190
C            SE(17)- FU0(I,I=1,4)                                       MAT11200
C            SE(21)- DU0(I,I=1,4)                                       MAT11210
C            SE(25)- KEYDS2                                             MAT11220
C            SE(26)- SECNO                                              MAT11230
C            SE(27)- S2Y                                                MAT11240
C            SE(28)- DU17                                               MAT11250
C            SE(29)- FU17                                               MAT11260
C            SE(30)- AREA1                                              MAT11270
C            SE(31)- CYCLEN                                             MAT11280
C            SE(32)- CSE                                                MAT11290
C            SE(33)- FU00                                               MAT11300
C            SE(34)- DU00                                               MAT11310
```

```
C       SE(35)= DKP                                                     MAT11320
C       SE(36)= TNPTFG                                                  MAT11330
C       SE(37)= POSTFG                                                  MAT11340
C       SE(38)= CO1                                                     MAT11350
C       SE(39)= RAT                                                     MAT11360
C       SE(40)= RAT1                                                    MAT11370
C       SE(41)= SQ6                                                     MAT11380
C       SE(42)= SR                                                      MAT11390
C       SE(43)= DP10                                                    MAT11400
C       SE(44)= SRFLAG                                                  MAT11410
C --USE     = EQUIVALENT UNLOADING STIFFNESS                            MAT11420
C       SE(45)=USE                                                      MAT11430
C       SE(46)= IOPT2                                                   MAT11440
C --IOPT2   = USED IN HYST14 SUBRTN (DNCKUL,DDISSI SUBRTN IN HYST14)     MAT11450
C       SE(47)= IOPT                                                    MAT11460
C       SE(48)= PSQ6                                                    MAT11470
C       SE(49)= KLAST                                                   MAT11480
C       SE(50)= DPTT                                                    MAT11490
C       SE(51)= DPT                                                     MAT11500
C       SE(52)= ESE                                                     MAT11510
C       SE(53)=ESE                                                      MAT11520
C       SE(54)= ESE                                                     MAT11530
C       SE(55)=PSE                                                      MAT11540
C       SE(56)= PSE_OLD                                                 MAT11550
C       SE(57)= EXCR(1)                                                 MAT11560
C       SE(58)= EXCR(2)                                                 MAT11570
C       SE(59)= EXCR(3)                                                 MAT11580
C--- -  SE(60),SE(61),SE(62) SEE DNE SUBRTN                            MAT11590
C       SE(60)= EXCR(4)                                                 MAT11600
C       SE(61)= EXCR(5)                                                 MAT11610
C       SE(62)= EXCR(6)                                                 MAT11620
C       SE(63)=FC                                                       MAT11630
C       SE(64)=FY                                                       MAT11640
C       SE(65)=FU                                                       MAT11650
C       SE(66)=FP                                                       MAT11660
C       SE(67)=DC                                                       MAT11670
C       SE(68)=DY                                                       MAT11680
C       SE(69)=DU                                                       MAT11690
C       SE(70)=DP                                                       MAT11700
C       SE(71)=DSINCL                                                   MAT11710
C       SE(72)=UPOS(1)                                                  MAT11720
C       SE(73)=UPOS(2)                                                  MAT11730
C       SE(74)=UPOS(3)                                                  MAT11740
C       SE(75)=UNEG(1)                                                  MAT11750
C       SE(76)=UNEG(2)                                                  MAT11760
C       SE(77)=UNEG(3)                                                  MAT11770
C       SE(78)=LENGTH                                                   MAT11780
C       SE(79)=WIDTH                                                    MAT11790
C       SE(80)= SHEAR SPAN LENGTH RATIO: SMAT(11+2*NSEG) OR MQD         MAT11800
C                                                                       MAT11810
C-----  INITIALIZE ---------                                            MAT11820
      SRFLAG=.FALSE.                                                    MAT11830
      POSTFG= TRUE                                                      MAT11840
      TNPTFG= FALSE.                                                    MAT11850
      KKL=1                                                             MAT11860
      CYCLEN=1                                                          MAT11870
      DO 31 L=1,4                                                       MAT11880
   31 FU0(L)=0                                                          MAT11890
      DO 32 L=1,4                                                       MAT11900
   32 DU0(L)=0.                                                         MAT11910
      SE( 3)=KKL                                                        MAT11920
      SE(31)=CYCLEN                                                     MAT11930
      NSEG=SMAT(1)                                                      MAT11940
      SMAT(3)=SE(78)                                                    MAT11950
      SMAT(4)=SE(79)                                                    MAT11960
      SMAT(11+2*NSEG)=SMAT(7+2*NSEG)/SMAT(4)                            MAT11970
      SE(32)=.5*SMAT(5)*SMAT(9)+.5*(SMAT(5)+SMAT(6))*(SMAT(10)-SMAT(9)) MAT11980
      SE(80)=SMAT(11+2*NSEG)                                            MAT11990
      SE(32)=SMAT(2)                                                    MAT12000
      MQD=SE(80)                                                        MAT12010
C FIND CO1, THIS COEFF. IS OBSERVED FROM EXPERIMENTAL DATA              MAT12020
      CO1=0.3                                                           MAT12030
      SE(38)=CO1                                                        MAT12040
      IF (TNPTFG) SE(36)=1                                              MAT12050
      IF(.NOT.TNPTFG) SE(36)=0                                          MAT12060
      IF(POSTFG) SE(37)=1                                               MAT12070
      IF( NOT.POSTFG) SE(37)=0                                          MAT12080
      IF(SRFLAG) SE(44)=1                                               MAT12090
      IF(.NOT.SRFLAG) SE(44)=0                                          MAT12100
C                                                                       MAT12110
      CALL OPNSTF(SMAT(5)  ,SMAT(6)  ,SMAT(7)  ,SMAT(8) ,              MAT12120
     &            SMAT(9)  ,SMAT(10) ,SMAT(11) ,SMAT(12),              MAT12130
     &                       SOC     ,SCY      ,                       MAT12140
     &            SOY      ,SYU      ,SUP      ,SCU      ,             MAT12150
     &            SCB      ,SOE                  )                     MAT12160
      SMAT(12+2*NSEG)=SOC                                               MAT12170
      SMAT(13+2*NSEG)=SCY                                               MAT12180
      SMAT(14+2*NSEG)=SOY                                               MAT12190
      SMAT(15+2*NSEG)=SYU                                               MAT12200
      SMAT(16+2*NSEG)=SUP                                               MAT12210
      SMAT(17+2*NSEG)=SCU                                               MAT12220
      SMAT(18+2*NSEG)=SCB                                               MAT12230
      SMAT(19+2*NSEG)=SOE                                               MAT12240
      SE(63)=SMAT(5)                                                    MAT12250
      SE(64)=SMAT(6)                                                    MAT12260
      SE(65)=SMAT(7)                                                    MAT12270
      SE(66)=SMAT(8)                                                    MAT12280
      SE(67)=SMAT(9)                                                    MAT12290
      SE(68)=SMAT(10)                                                   MAT12300
      SE(69)=SMAT(11)                                                   MAT12310
      SE(70)=SMAT(12)                                                   MAT12320
      SE( 1)=SOC                                                        MAT12330
      SE(10)=SCY                                                        MAT12340
      SE(11)=SOY                                                        MAT12350
      SE(12)=SYU                                                        MAT12360
      SE(13)=SUP                                                        MAT12370
      SE(14)=SCU                                                        MAT12380
      SE(15)=SCB                                                        MAT12390
      SE(16)=SOE                                                        MAT12400
      SE( 9) = SE( 1)                                                   MAT12410
      RETURN                                                            MAT12420
C                                                                       MAT12430
C------------------------- GET STIFFNESS TERMS ------------------------- MAT12440
  100 CONTINUE                                                          MAT12450
      IF(SE(36) EQ.1) TNPTFG= TRUE.                                     MAT12460
      IF(SE(36).EQ.0) TNPTFG=.FALSE.                                    MAT12470
      IF(SE(37).EQ 1) POSTFG=.TRUE.                                     MAT12480
      IF(SE(37).EQ.0) POSTFG=.FALSE.                                    MAT12490
C SRFLAG TO CONTROL KKL -1 OR 3 DEPENDING ON INITIAL DISP. INCREMENT    MAT12500
      IF(SE(44).EQ.1) SRFLAG=.TRUE.                                     MAT12510
      IF(SE(44).EQ.0) SRFLAG=.FALSE.                                    MAT12520
C.  TO SET INITIAL KKL BASED ON INITIAL DISPLACEMENT INCREMENT(DSINC)    MAT12530
C  IF INITIAL DISPL. INCREMENT IS POSITIVE, THEN POSTFG=.TRUE           MAT12540
      J11=NSEG+5                                                        MAT12550
      J22=NSEG+6                                                        MAT12560
      J33=NSEG+7                                                        MAT12570
      J52=6+2*NSEG                                                      MAT12580
      J58=12+2*NSEG                                                     MAT12590
      J59=13+2*NSEG                                                     MAT12600
      J65=18+2*NSEG                                                     MAT12610
      KKL=SE(3)                                                         MAT12620
      CYCLEN=SE(31)                                                     MAT12630
      LRULE=SE( 2)                                                      MAT12640
      IOPT2=SE(46)                                                      MAT12650
       CALL HYST14( P      ,D      ,PL      ,DL      ,SMAT(3)  ,       MAT12660
     &   SMAT( 5) ,SMAT(J11),SMAT( 6) ,SMAT(J22),SMAT( 7) ,SMAT(J33),   MAT12670
     &   SMAT(J52),SMAT(J58),SMAT(J59),SMAT(J65),SE( 1)   ,LRULE    ,   MAT12680
     &   KKL     ,SE( 4)   ,SE(-5)   ,SE( 6)   ,SE( 8)   ,SE( 9)   ,   MAT12690
     &   SE(10)  ,SE(12)   ,SE(13)   ,SE(15)   ,SE(16)   ,SE(17)   ,   MAT12700
     &   SE(21)  ,SE(25)   ,SE(29)   ,SE(30)   ,CYCLEN   ,SE(32)   ,   MAT12710
     &   SE(33)  ,TNPTFG   ,POSTFG   ,SE(38)   ,SE(41)   ,SE(42)   ,   MAT12720
     &   SE(45)  ,IOPT2    ,SE(48)   ,SE(49)   ,SE(50)   ,SE(51)   ,   MAT12730
     &   SE(52)  ,SE(55)   ,SE(56)   ,SE(57)   ,SE(63)   ,SE(64)   ,   MAT12740
     &   SE(65)  ,SE(66)   ,SE(67)   ,SE(68)   ,SE(69)   ,SE(70)   ,   MAT12750
     &   SE(71)  ,SE(72)   ,SE(75)   )                                  MAT12760
      IF (TNPTFG) SE(36)=1                                              MAT12770
      IF (.NOT.TNPTFG) SE(36)=0                                         MAT12780
      IF (POSTFG) SE(37)=1                                              MAT12790
      IF ( NOT POSTFG) SE(37)=0                                         MAT12800
      IF (SRFLAG) SE(44)=1                                              MAT12810
      IF ( NOT SRFLAG) SE(44)=0                                         MAT12820
      SE( 3)=KKL                                                        MAT12830
      SE(31)=CYCLEN                                                     MAT12840
      SE( 2)=LRULE                                                      MAT12850
      SE(46)=IOPT2                                                      MAT12860
      RETURN                                                            MAT12870
C                                                                       MAT12880
C------ TRANSFER ENERGIES FOR BOTH IOPT=3 AND 4.                        MAT12890
C .    (NEGLECT DUCTILITIES CALCULATION AT PRESENT STAGE)               MAT12900
  400 EESE=SE(52)                                                       MAT12910
      EPSE=SE(55)                                                       MAT12920
C------ TRANSFER DUCTILITIES AND EXCRUSION RATIOS FOR BOTH IOPT=3 $ 4.  MAT12930
      DO 410 I=1,3                                                      MAT12940
      DUCT(I)=MAX(SE(71+I),SE(74+I))                                    MAT12950
      EXCR(I)=SE(56+I)                                                  MAT12960
  410 EXCR(I+3)=SE(59+I)                                                MAT12970
      RETURN                                                            MAT12980
C                                                                       MAT12990
C---------------------------- DAMAGE INDEX ------------------------------MAT13000
  500 EESE=SE(52)                                                       MAT13010
      EPSE=SE(55)                                                       MAT13020
C------ DDMAX IS THE LARGER OF DMAX OR DM IN HYST14 SUBRTN              MAT13030
      DDMAX=MAX(SE(5),SE(6))                                            MAT13040
      PY=SMAT(6)                                                        MAT13050
C--- -  DDU ,FAILURE DISPLACEMENT OF BACKBONE CURVE                    MAT13060
      DDU=SMAT(12)                                                      MAT13070
      BDAM =SMAT(5+2*NSEG)                                              MAT13080
      DAMAGE=DDMAX/DDU *BDAM/(PY*DDU)*EPSE                              MAT13090
      RETURN                                                            MAT13100
      END                                                               MAT13110
C                                                                       OPNS0010
C ===> TO CALCULATE AVAILABLE STIFFNESS USED IN THE HYSTERESIS LOOP     OPNS0020
C                                                                       OPNS0030
      SUBROUTINE OPNSTF(FC,FY,FU,FP,DC,DY,DU,DP,SOC,SCY,SOY,           OPNS0040
     &                  SYU,SUP,SCU,SCB,SOE)                            OPNS0050
C                                                                       OPNS0060
C COMPUTE STIFFNESS S(OC)                                               OPNS0070
      SOC=FC/DC                                                         OPNS0080
C COMPUTE STIFFNESS S(CY)                                               OPNS0090
      SCY=(FY-FC)/(DY-DC)                                               OPNS0100
C COMPUTE STIFFNESS S(CY)                                               OPNS0110
      SOY=FY/DY                                                         OPNS0120
C COMPUTE STIFFNESS S(YU)                                               OPNS0130
      SYU=(FU-FY)/(DU-DY)                                               OPNS0140
C COMPUTE STIFFNESS S(UP)                                               OPNS0150
      SUP=(FP-FU)/(DP-DU)                                               OPNS0160
C COMPUTE STIFFNESS S(CU)                                               OPNS0170
      SCU=(FU-FC)/(DU-DC)                                               OPNS0180
C COMPUTE STIFFNESS S(OB) ; B IS MIDPT. OF PT. A AND PT. Y              OPNS0190
C PT. A -INTERSECTION OF EXTENSION OF OC WITH HORIZONTAL LINE THROU Y   OPNS0200
      XA=FY/SOC                                                         OPNS0210
      XB=(XA+DY)/2                                                      OPNS0220
      SOB=FY/XB                                                         OPNS0230
C COMPUTE STIFFNESS S(OE) ; E IS MIDPT. OF PT. D AND PT. U              OPNS0240
C PT. D -INTERSECTION OF EXTENSION OF OY WITH HORIZONTAL LINE THROU U   OPNS0250
      XD=FU/SOY                                                         OPNS0260
      XE=(XD+DU)/2                                                      OPNS0270
      SOE=FU/XE                                                         OPNS0280
      RETURN                                                            OPNS0290
      END                                                               OPNS0300
C-----------------------------------------------------------------------HYST0010
C-- THIS HYSTERESIS SUBROUTINE FOR PERFORATED SHEAR WALL. . HYST14..... HYST0020
C-----------------------------------------------------------------------HYST0030
C                                                                       HYST0040
      SUBROUTINE HYST14(FP      ,DP      ,PPP     ,DPP     ,W     ,     HYST0050
     &            OPC     ,ODC     ,OPY     ,ODY     ,OPU     ,ODU  ,  HYST0060
     &            HO      ,OSOC    ,OSCY    ,OSCB    ,K       ,LRULE,  HYST0070
     &            KKL     ,DSINC   ,DMAX    ,DH      ,KP      ,SOC  ,  HYST0080
     &            SCY     ,SYU     ,SUP     ,SCB     ,SOE     ,FU0  ,  HYST0090
     &            DU0     ,KEYDS2  ,FU17    ,AREA1   ,CYCLEN  ,CSE  ,  HYST0100
     &            FU00    ,TNPTFG  ,POSTFG  ,CO1     ,SQ6     ,SR   ,  HYST0110
     &            USE     ,IOPT2   ,PSQ6    ,KLAST   ,DPTT    ,DPT  ,  HYST0120
     &            ESE     ,PSE     ,PSEOLD  ,EXCR    ,PC      ,PY   ,  HYST0130
     &            FU      ,PP      ,DC      ,DY      ,DU      ,DP   ,  HYST0140
     &            DSINCL  ,UPOS    ,UNEG    )                         HYST0150
C                                                                       HYST0160
      INTEGER CYCLEN                                                    HYST0170
      REAL K,KP,DU0(4),FU0(4),KLAST,KEYDS1,KEYDS2                       HYST0180
      REAL UPOS(3),UNEG(3),EXCR(6),ESE(3)                               HYST0190
      LOGICAL TNPTFG,POSTFG                                             HYST0200
C                                                                       HYST0210
C                                                                       HYST0220
C---------- ---------- ---------- ---------- ---------- ---------- ----HYST0230
C        DEFINITION FOR VARIABLE :                                      HYST0240
C-----------------------------------------------------------------------HYST0250
C HS ------ WALL SHEAR SPAN,FROM LOADING POINT TO THE BOTTOM OF WALL    HYST0260
C WID------ WALL WIDTH                                                 HYST0270
C HO ------ THE HEIGHT OF THE OPENING                                  HYST0280
C H  ------ THE HEIGHT OF THE WALL                                     HYST0290
C AD  ----- THE NET AREA OF DIAGONAL STEEL BARS                        HYST0300
C AV ------ THE NET AREA OF VERTICAL STEEL BARS                        HYST0310
C HO,H ABOVE JUST USED FOR CALCULATION OF FAILURE DISPL.(STIFF SUBRTN)  HYST0320
C ODC ------INITIAL (OR ORIGINAL) CRACKING DISPL.                      HYST0330
C ODY ------INITIAL YIELDING DISPL.                                    HYST0340
C ODU ------INITIAL ULTIMATE DISPL.                                    HYST0350
C ODP ------INITIAL FAILURE DISPL.                                     HYST0360
C OPC ------INITIAL CRACKING LOAD                                      HYST0370
C OPY ------INITIAL YIELDING LOAD                                      HYST0380
C OPU ------INITIAL ULTIMATE LOAD                                      HYST0390
C OPP ------INITIAL FAILURE LOAD                                       HYST0400
C DC ------CRACKING DISPL.                                            HYST0410
C DY ------YIELDING DISPL.                                            HYST0420
C DU ------ULTIMATE DISPL.                                            HYST0430
C DP ------FAILURE DISPL.                                             HYST0440
C PC ------CRACKING LOAD                                              HYST0450
C PY ------YIELDING LOAD                                              HYST0460
C PU ------ULTIMATE LOAD                                              HYST0470
C PP ------FAILURE LOAD                                               HYST0480
C DMAX ----MAX. DISPL. FOR THE LOADING SENSE IN THE CURRENT CYCLE      HYST0490
C DCP ------CRACKING DISPL. IN THE NEGATIVE LOADING SENSE              HYST0500
C DYP ------YIELDING DISPL. IN THE NEGATIVE LOADING SENSE              HYST0510
C DKP---SEE DFORCE SUBRTN, WHICH EQUAL TO CURRENT DP AND ASSIGNED TO   HYST0520
C       DPP JUST BEFORE CHANGING DP TO A NEW VALUE                     HYST0530
C PCP ------CRACKING LOAD IN THE NEGATIVE LOADING SENSE                HYST0540
C PYP ------YIELDING LOAD IN THE NEGATIVE LOADING SENSE                HYST0550
C DP ,PP------CURRENT DISPL. ,CURRENT LOAD                             HYST0560
C PP ------FAILURE LOAD=0.5*(PC+PY)                                    HYST0570
C PM ------MAX. LOAD BEFORE THE CURRENT CYCLE                          HYST0580
C FU0 ,DU0------LOAD AND DISPL. FOR UNLOADING POINT U0                 HYST0590
C PY08------0.8*PY                                                     HYST0600
C KKL=1 --- FOR POSITIVE LOADING                                       HYST0610
C KKL=2 --- FOR POSITIVE UNLOADING                                     HYST0620
C KKL=3 --- FOR NEGATIVE LOADING                                       HYST0630
C KKL=4 --- FOR NEGATIVE UNLOADING                                     HYST0640
C KKL=5 --- FOR REVERSAL LOADING FROM POSITIVE TO NEGATIVE             HYST0650
C KKL=6 --- FOR REVERSAL LOADING FROM NEGATIVE TO POSITIVE             HYST0660
C KKL=7  -- FOR RELOADING AFTER UNLOADING IN POSITIVE DIRECTION        HYST0670
C KKL=8 --- FOR RELOADING AFTER UNLOADING IN NEGATIVE DIRECTION        HYST0680
C SRFLAG--- SEE NEWST1 SUBRTN;TO CONTROL REVERS. LOAD IN YIELD STATE   HYST0690
C MQD --- M/(Q*D) ,SHEAR-SPAN LENGTH RATIO                            HYST0700
C LWP------L'/W' FOR OPENING SHEAR WALL CASE                           HYST0710
C KLAST----LAST STIFFNESS                                              HYST0720
C KEYDS1=KEYDS(SECNO-2)  ;KEYDS2=KEYDS(SECNO-1)                        HYST0730
C FU0(1)=FU0(2*CYCLEN), FU0(3)=FU0(3*CYCLEN-1)                         HYST0740
C FU0(3)=FU0(2*CYCLEN-2); FU0(4)=FU0(2*CYCLEN-3)                       HYST0750
C DU0(1)=DU0(2*CYCLEN); DU0(2)=DU0(2*CYCLEN-1)                         HYST0760
C USE- EQUIVALENT UNLOADING STIFFNESS (SAME AS SE IN THE HYST03)        HYST0770
C                                                                       HYST0780
C                                                                       HYST0790
      LRULST=LRULE                                                      HYST0800
      KLAST=K                                                           HYST0810
      KKLLST=KKL                                                        HYST0820
```

```
      DSINC=DP-DPP                                                      HYST0830
C ---- IF WALL FAIL(K=0) AT FORCE=0 THEN STILL K=0 AND RETURN          HYST0840
      IF(K.EQ.0) THEN                                                   HYST0850
        WRITE(108,900)                                                  HYST0860
900     FORMAT(1X,'*** WALL FAIL, AND STOP THE PROGRAM')                HYST0870
        STOP                                                            HYST0880
      END IF                                                            HYST0890
C                                                                       HYST0900
C--- WHEN DISPLACEMENT INCREMENT CHANGE SIGN, REMEMBER TWO PREVIOUS     HYST0910
C    MAX. DISP. OF TWO HALF CYCLES                                      HYST0920
C                                                                       HYST0930
      IF((DSINCL*DSINC).LT.0) THEN                                      HYST0940
      KEYDS1=KEYDS2                                                     HYST0950
      KEYDS2=DPP                                                        HYST0960
C                                                                       HYST0970
C TO MEMORIZE THE POINTS AT WHICH DSINC*DSINCL < 0.                     HYST0980
C                                                                       HYST0990
        IF(POSTPG) THEN                                                 HYST1000
          IF(KKLLST.EQ.2 .AND. KKL EQ 5) THEN                           HYST1010
            FU0(2)=0                                                    HYST1020
            DU0(2)=0                                                    HYST1030
          END IF                                                        HYST1040
          IF(KKLLST.EQ.4 .AND. KKL.EQ.6) THEN                           HYST1050
            FU0(1)=0.                                                   HYST1060
            DU0(1)=0.                                                   HYST1070
          END IF                                                        HYST1080
          IF(DSINC.LT.0) THEN                                           HYST1090
            IF(KKLLST.EQ.8 .OR. KKLLST.EQ.10) THEN                      HYST1100
              FU0(1)=MAX(FU0(1),FPP)                                    HYST1110
              DU0(1)=MAX(DU0(1),KEYDS2)                                 HYST1120
            ELSE                                                        HYST1130
              FU0(1)=FPP                                                HYST1140
              DU0(1)=KEYDS2                                             HYST1150
            END IF                                                      HYST1160
          END IF                                                        HYST1170
          IF(DSINC.GT.0) THEN                                           HYST1180
            IF(KKLLST.EQ.7 .OR. KKLLST.EQ.9) THEN                       HYST1190
              FU0(2)=MIN(FU0(2),FPP)                                    HYST1200
              DU0(2)=MIN(DU0(2),KEYDS2)                                 HYST1210
            ELSE                                                        HYST1220
              FU0(2)=FPP                                                HYST1230
              DU0(2)=KEYDS2                                             HYST1240
            END IF                                                      HYST1250
          END IF                                                        HYST1260
        ELSE                                                            HYST1270
          IF(KKLLST.EQ.2  AND  KKL.EQ.5) THEN                           HYST1280
            FU0(1)=0.                                                   HYST1290
            DU0(1)=0.                                                   HYST1300
          END IF                                                        HYST1310
          IF(KKLLST EQ.4  AND. KKL.EQ.6) THEN                           HYST1320
            FU0(2)=0.                                                   HYST1330
            DU0(2)=0.                                                   HYST1340
          END IF                                                        HYST1350
          IF(DSINC.GT.0) THEN                                           HYST1360
            IF(KKLLST.EQ.7 .OR. KKLLST.EQ.9) THEN                       HYST1370
              FU0(1)=MIN(FU0(1),FPP)                                    HYST1380
              DU0(1)=MIN(DU0(1),KEYDS2)                                 HYST1390
            ELSE                                                        HYST1400
              FU0(1)=FPP                                                HYST1410
              DU0(1)=KEYDS2                                             HYST1420
            END IF                                                      HYST1430
          END IF                                                        HYST1440
          IF(DSINC.LT.0) THEN                                           HYST1450
            IF(KKLLST.EQ.8 .OR. KKLLST.EQ.10) THEN                      HYST1460
              FU0(2)=MAX(FU0(2),FPP)                                    HYST1470
              DU0(2)=MAX(DU0(2),KEYDS2)                                 HYST1480
            ELSE                                                        HYST1490
              FU0(2)=FPP                                                HYST1500
              DU0(2)=KEYDS2                                             HYST1510
            END IF                                                      HYST1520
          END IF                                                        HYST1530
        END IF                                                          HYST1540
C                                                                       HYST1550
        IF((ABS(KEYDS2) LT.DC) AND.(KKL.EQ 6).AND.(KEYDS2.GT 0.)) THEN  HYST1560
          KKL=2                                                         HYST1570
          GO TO 23                                                      HYST1580
        END IF                                                          HYST1590
        IF((ABS(KEYDS2) LT.DC) AND.(KKL.EQ.5).AND.(KEYDS2.LT.0.)) THEN  HYST1600
          KKL=4                                                         HYST1610
          GO TO 23                                                      HYST1620
        END IF                                                          HYST1630
        IF((KKL.EQ.2).AND.((DM.GE.DC).OR.(DMAX.GE.DC))) THEN            HYST1640
          KKL=8                                                         HYST1650
          GO TO 23                                                      HYST1660
        END IF                                                          HYST1670
        IF((KKL.EQ.2).AND.(DM.LT.DC).AND.(DMAX.LT.DC)) THEN             HYST1680
          KKL=1                                                         HYST1690
          GO TO 23                                                      HYST1700
        END IF                                                          HYST1710
        IF((KKL.EQ.4).AND.((DM.GE.DC).OR.(DMAX GE DC))) THEN            HYST1720
          KKL=7                                                         HYST1730
          GO TO 23                                                      HYST1740
        END IF                                                          HYST1750
        IF((KKL.EQ.4).AND.(DM.LT.DC).AND.(DMAX.LT.DC)) THEN             HYST1760
          KKL=3                                                         HYST1770
          GO TO 23                                                      HYST1780
        END IF                                                          HYST1790
C                                                                       HYST1800
        IF(KKL.EQ.1) THEN                                               HYST1810
          KKL=2                                                         HYST1820
          GO TO 23                                                      HYST1830
        ELSE IF(KKL.EQ 3) THEN                                          HYST1840
          KKL=4                                                         HYST1850
          GO TO 23                                                      HYST1860
        ELSE IF((KKL EQ 2) AND.(FP.LE.0).AND.(DPP.NE.0)                 HYST1870
     &          .AND (DSINC.LT.0)) THEN                                 HYST1880
          KKL=5                                                         HYST1890
          GO TO 23                                                      HYST1900
        ELSE IF((KKL.EQ.4) AND.(FP.LE.0) AND.(DPP NE.0)                 HYST1910
     &          .AND.(DSINC.GT.0)) THEN                                 HYST1920
          KKL=6                                                         HYST1930
          GO TO 23                                                      HYST1940
        END IF                                                          HYST1950
        IF(KKL.EQ.8) THEN                                               HYST1960
          KKL=2                                                         HYST1970
          GO TO 23                                                      HYST1980
        END IF                                                          HYST1990
        IF(KKL.EQ.7) THEN                                               HYST2000
          KKL=4                                                         HYST2010
          GO TO 23                                                      HYST2020
        END IF                                                          HYST2030
        IF(KKL.EQ 10) THEN                                              HYST2040
          KKL=2                                                         HYST2050
          GO TO 23                                                      HYST2060
        END IF                                                          HYST2070
        IF(KKL.EQ.9) THEN                                               HYST2080
          KKL=4                                                         HYST2090
          GO TO 23                                                      HYST2100
        END IF                                                          HYST2110
        IF(KKL.EQ.5) THEN                                               HYST2120
          KKL=10                                                        HYST2130
          PSEOLD=AREA1                                                  HYST2140
          GO TO 13                                                      HYST2150
        END IF                                                          HYST2160
        IF(KKL.EQ.6) THEN                                               HYST2170
          KKL=9                                                         HYST2180
          PSEOLD=AREA1                                                  HYST2190
          GO TO 13                                                      HYST2200
        END IF                                                          HYST2210
C                                                                       HYST2220
        IF(KEYDS2.LT  0.0.AND.(ABS(KEYDS2).LT.DC) AND.(KKL.EQ.5)) KKL=4 HYST2230
        IF(KEYDS2.GT  0.0 AND.(KEYDS2 LT DC).AND.(KKL.EQ.6)) KKL=2      HYST2240
      END IF                                                            HYST2250
C                                                                       HYST2260
   13 IF((KKL EQ.10).AND.((DSINC*DSINCL).LT 0)) THEN                    HYST2270
        IF(POSTPG) THEN                                                 HYST2280
          FPFP=FU0(1)                                                   HYST2290
          DDDD=DU0(1)                                                   HYST2300
        ELSE                                                            HYST2310
          FPFP=FU0(2)                                                   HYST2320
          DDDD=DU0(2)                                                   HYST2330
        END IF                                                          HYST2340
        FP1=FPFP*0.9                                                    HYST2350
        DP1=DDDD                                                        HYST2360
        KP=(FP1-0 )/(DP1-DPP)                                           HYST2370
        X=KP                                                            HYST2380
        USE=MAX(K,SCB)                                                  HYST2390
        IF(POSTPG) CYCLEN=CYCLEN+1                                      HYST2400
        IOPT2=3                                                         HYST2410
        CALL DDISSI(IOPT2,ESE,PSE,PSEOLD,DPP,FPP,DPP,0.,AREA1,          HYST2420
     &              K,USE,KKL,PC,DC,FY,DY,DM,DMAX,GDC,SCB,FU00)         HYST2430
        FP=0.+(DP-DPP)*KP                                               HYST2440
        FPP=0.                                                          HYST2450
        LRULE=51                                                        HYST2460
        CALL NEWSTI(GDC,GDY,GDU,GFC,GFY,GFU,AREA1,DM,FC,FY,             HYST2470
     &              FU,FP,DC,DY,DU,DF,HO,H,GSOC,GSCY,SOC,               HYST2480
     &              SCY,SYU,SUF,SR,SCB,SOE,SQ6,PSQ6,DPT,SDP)            HYST2490
        CALL DINTER(PC,FY,FU,FP,FPP,DC,DY,DU,DF,DPP,SOC,SCY,SYU,SUF,K,  HYST2500
     &              FP,DP,DSINC,KKL,LRULE,POSTPG,DU0,FU0)               HYST2510
        DMAX=ABS(DP)                                                    HYST2520
        DSINCL=DSINC                                                    HYST2530
        RETURN                                                          HYST2540
      END IF                                                            HYST2550
      IF(KKL.EQ.10) THEN                                                HYST2560
        IF(POSTPG) THEN                                                 HYST2570
          DDKK=DU0(1)                                                   HYST2580
        ELSE                                                            HYST2590
          DDKK=DU0(2)                                                   HYST2600
        END IF                                                          HYST2610
        IF(ABS(DP) LT.ABS(DDKK)) THEN                                   HYST2620
          LRULE=52                                                      HYST2630
          CALL DINTER(PC,FY,FU,FP,FPP,DC,DY,DU,DF,DPP,SOC,SCY,SYU,SUF,  HYST2640
     &                K,PP,DP,DSINC,KKL,LRULE,POSTPG,DU0,FU0)           HYST2650
        END IF                                                          HYST2660
        IF(ABS(DP).GT.ABS(DDKK)) THEN                                   HYST2670
          K=KP*0.5                                                      HYST2680
          IF(K.NE.KLAST) THEN                                           HYST2690
            IF(POSTPG) FP=FPP+(DU0(1)-DPP)*KLAST+(DP-DU0(1))*K          HYST2700
            IF(.NOT.POSTPG) FP=FPP+(DU0(2)-DPP)*KLAST+(DP-DU0(2))*K     HYST2710
          END IF                                                        HYST2720
          LRULE=53                                                      HYST2730
          CALL DINTER(PC,FY,FU,FP,FPP,DC,DY,DU,DF,DPP,SOC,SCY,SYU,SUF,  HYST2740
     &                K,PP,DP,DSINC,KKL,LRULE,POSTPG,DU0,FU0)           HYST2750
        END IF                                                          HYST2760
        DSINCL=DSINC                                                    HYST2770
        DMAX=MAX(ABS(DP),DMAX)                                          HYST2780
        IOPT2=1                                                         HYST2790
        CALL DDISSI(IOPT2,ESE,PSE,PSEOLD,DPP,FPP,DP,FP,AREA1,           HYST2800
     &              K,USE,KKL,PC,DC,FY,DY,DM,DMAX,GDC,SCB,FU00)         HYST2810
        RETURN                                                          HYST2820
      END IF                                                            HYST2830
C                                                                       HYST2840
      IF((KKL.EQ.9).AND.((DSINC*DSINCL).LT.0)) THEN                     HYST2850
        IF(POSTPG) THEN                                                 HYST2860
          FPFP=FU0(2)                                                   HYST2870
          DDDD=DU0(2)                                                   HYST2880
        ELSE                                                            HYST2890
          FPFP=FU0(1)                                                   HYST2900
          DDDD=DU0(1)                                                   HYST2910
        END IF                                                          HYST2920
        FP1=FPFP*0.9                                                    HYST2930
        DP1=DDDD                                                        HYST2940
        KP=(FP1-0.)/(DP1-DPP)                                           HYST2950
        X=KP                                                            HYST2960
        USE=MAX(K,SCB)                                                  HYST2970
        IF(.NOT.POSTPG) CYCLEN=CYCLEN+1                                 HYST2980
        IOPT2=3                                                         HYST2990
        CALL DDISSI(IOPT2,ESE,PSE,PSEOLD,DPP,FPP,DPP,0.,AREA1,          HYST3000
     &              K,USE,KKL,PC,DC,FY,DY,DM,DMAX,GDC,SCB,FU00)         HYST3010
        FP=0.+(DP-DPP)*KP                                               HYST3020
        FPP=0.                                                          HYST3030
        LRULE=51                                                        HYST3040
        CALL NEWSTI(GDC,GDY,GDU,GFC,GFY,GFU,AREA1,DM,FC,FY,             HYST3050
     &              FU,FP,DC,DY,DU,DF,HO,H,GSOC,GSCY,SOC,               HYST3060
     &              SCY,SYU,SUF,SR,SCB,SOE,SQ6,PSQ6,DPT,SDP)            HYST3070
        CALL DINTER(PC,FY,FU,FP,FPP,DC,DY,DU,DF,DPP,SOC,SCY,SYU,SUF,K,  HYST3080
     &              FP,DP,DSINC,KKL,LRULE,POSTPG,DU0,FU0)               HYST3090
        DMAX=ABS(DP)                                                    HYST3100
        DSINCL=DSINC                                                    HYST3110
        RETURN                                                          HYST3120
      END IF                                                            HYST3130
      IF(KKL.EQ.9) THEN                                                 HYST3140
        IF(POSTPG) THEN                                                 HYST3150
          DDKK=DU0(2)                                                   HYST3160
        ELSE                                                            HYST3170
          DDKK=DU0(1)                                                   HYST3180
        END IF                                                          HYST3190
        IF(ABS(DP).LT.ABS(DDKK)) THEN                                   HYST3200
          LRULE=52                                                      HYST3210
          CALL DINTER(PC,FY,FU,FP,FPP,DC,DY,DU,DF,DPP,SOC,SCY,SYU,SUF,  HYST3220
     &                K,FP,DP,DSINC,KKL,LRULE,POSTPG,DU0,FU0)           HYST3230
        END IF                                                          HYST3240
        IF(ABS(DP).GT.ABS(DDKK)) THEN                                   HYST3250
          X=KP*0.5                                                      HYST3260
          IF(K.NE.KLAST) THEN                                           HYST3270
            IF(POSTPG)  FP=FPP+(DU0(2)-DPP)*KLAST+(DP-DU0(2))*K         HYST3280
            IF(.NOT.POSTPG)  FP=FPP+(DU0(1)-DPP)*KLAST+(DP-DU0(1))*K    HYST3290
          END IF                                                        HYST3300
          LRULE=53                                                      HYST3310
          CALL DINTER(PC,FY,FU,FP,FPP,DC,DY,DU,DF,DPP,SOC,SCY,SYU,SUF,  HYST3320
     &                K,FP,DP,DSINC,KKL,LRULE,POSTPG,DU0,FU0)           HYST3330
        END IF                                                          HYST3340
        DMAX=MAX(ABS(DP),DMAX)                                          HYST3350
        DSINCL=DSINC                                                    HYST3360
        IOPT2=1                                                         HYST3370
        CALL DDISSI(IOPT2,ESE,PSE,PSEOLD,DPP,FPP,DP,FP,AREA1,           HYST3380
     &              K,USE,KKL,PC,DC,FY,DY,DM,DMAX,GDC,SCB,FU00)         HYST3390
        RETURN                                                          HYST3400
      END IF                                                            HYST3410
C------ CHECK IF PASS TURNING POINT                                     HYST3420
   23 CALL DCURRD(KKL,DSINC,FP,FPP,DM,FC,FY,FU,FF,X,DP,DMAX,           HYST3430
     &       DPP,DC,DY,DU,CYCLEN,AREA1,PMAX,TNPTPG,GDC,GDY,GDU,GPU,     HYST3440
     &       DF,HO,H,GSOC,POSTPG,OSCY,SOC,SCY,SYU,SUF,SR,SOB,SOE,       HYST3450
     &       SQ6,GFC,GFY,PSQ6,DPT,DPTT,KKLLST,USE,PSEOLD,UPOS,ESE,PSE,  HYST3460
     &       CSE,EXCR,IOPT2)                                            HYST3470
C                                                                       HYST3480
      IF((X.EQ.0).AND.(FP.EQ.0).AND.(FP.EQ FP)) GO TO 77                HYST3490
C                                                                       HYST3500
C -- TO CALCULATE THE STIFFNESS IN EACHLOAD STEP                        HYST3510
      CALL DNCKUL(FC,FY,FU,FF,DC,DY,DU,DP,FP,DP,DMAX,DM,KP,SOC,         HYST3520
     &     SCY,SYU,SUF,SOB,SOE,KKL,LRULE,FPP,DPP,FU0,DU0,DSINC,K,FU17,  HYST3530
     &     OSOB,AREA1,PU00,POSTPG,PSEOLD,PSE,ESE,HO,H,GPC,GPY,GPU,GDC,  HYST3540
     &     GDY,GDU,SQ6,GSOC,OSCY,SR,KEYDS2,CO1,PSQ6,KLAST,DPTT,DPT,     HYST3550
     &     FPT,TNPTPG,KKLLST,LRULST,IOPT2,USE)                          HYST3560
C                                                                       HYST3570
      IF(X.EQ.0.) RETURN                                                HYST3580
   77 IF(TNPTPG) THEN                                                   HYST3590
        IF(KKL.EQ.1 .OR. KKL.EQ 8  OR. KKL EQ.10  OR  KKL.EQ 3         HYST3600
     &     .OR  KKL.EQ.7 .OR. KKL.EQ.9  OR  KKL.EQ.5 OR. KKL EQ.6) THEN HYST3610
          IF(ABS(FP).LE.(0.7*FY)  AND  KKL.NE 1 .AND. KKL NE 3) THEN    HYST3620
            USE=K                                                       HYST3630
            IOPT2=3                                                     HYST3640
          END IF                                                        HYST3650
          IF(ABS(FP) GT.(0 7*FY)) IOPT2=1                               HYST3660
          CALL DDISSI(IOPT2,ESE,PSE,PSEOLD,DPT,FPT,DP,FP,AREA1,         HYST3670
     &                K,USE,KKL,PC,DC,FY,DY,DM,DMAX,GDC,SOB,FU00)       HYST3680
        ELSE IF(KKL.EQ 2  OR. KKL EQ 4) THEN                            HYST3690
          IOPT2=3                                                       HYST3700
          CALL DDISSI(IOPT2,ESE,PSE,PSEOLD,DPT,FPT,DP,FP,AREA1,         HYST3710
     &                X,USE,KKL,PC,DC,FY,DY,DM,DMAX,GDC,SOB,FU00)       HYST3720
        END IF                                                          HYST3730
        TNPTPG= FALSE.                                                  HYST3740
```

```
      ELSE IF((KKL.NE.5  AND. KKL.NE.6) OR.                             HYST3750
     &        ((KKL EQ 5 .OR. KKL.EQ.6) .AND. ((DP*FP).GT.0.))) THEN    HYST3760
        IF(ABS(FP) LE.(0.7*FY) .AND. KKL.NE.1  AND. KKL NE.3) THEN      HYST3770
          USE=K                                                         HYST3780
          IOPT2=3                                                       HYST3790
        END IF                                                          HYST3800
        IF(ABS(FP).GT.(0.7*FY)) IOPT2=1                                 HYST3810
        CALL DDISSI(IOPT2,ESE,PSE,PSEOLD,DPP,FPP,DP,FP,AREA1,           HYST3820
     &              K,USE,KKL,PC,DC,FY,DY,DM,DMAX,CDC,SOB,FU00)         HYST3830
      ELSE IF((KKL EQ.5 .OR. KKL EQ.6) .AND.((DP*FP).LT.0.)) THEN       HYST3840
        USE=1E25                                                        HYST3850
        CALL DDISSI(IOPT2,ESE,PSE,PSEOLD,DPP,FPP,DP,FP,AREA1,           HYST3860
     &              K,USE,KKL,PC,DC,FY,DY,DM,DMAX,CDC,SOB,FU00)         HYST3870
      END IF                                                            HYST3880
      DSINCL=DSINC                                                      HYST3890
C CALC DUCTILITY                                                        HYST3900
      IF(KXL.EQ.1  .OR. KXL.EQ 8 .OR. KXL.EQ.10) THEN                   HYST3910
        DLI=MAX(DP,DPP)                                                 HYST3920
        CALL DNE(1,UPOS,EXCR,ODY,DLI,ESE(2),PSE,PSEOLD,CSE)             HYST3930
      END IF                                                            HYST3940
      IF(KKL EQ.3 .OR  KXL.EQ.7  OR. KXL EQ.9 ) THEN                    HYST3950
        DLI=MIN(DP,DPP)                                                 HYST3960
        CALL DNE(1,UNEG,EXCR,-ODY,DLI,ESE(3),PSE,PSEOLD,CSE)            HYST3970
      END IF                                                            HYST3980
   66 RETURN                                                            HYST3990
      END                                                               HYST4000
C                                                                       DNCX0010
C*-----------------------------------------------------------------------DNCX0020
C-- SUBROUTINE DNCKUL  SUBRTN (FIND STIFFNESS AND FORCE IN EACH STEP) --DNCX0030
C*-----------------------------------------------------------------------DNCX0040
      SUBROUTINE DNCKUL(FC,FY,FU,FP,DC,DY,DU,DP,FP,DP,DMAX,DM,KP,SOC,   DNCK0050
     &    SCY,SYU,SUP,SOB,SOE,KXL,LRULE,FPP,DPP,FU0,DU0,DSINC,K,FU17,   DNCK0060
     &    OSOB,AREA1,FU00,POSTFG,PSEOLD,PSE,ESE,HO,H,OPC,OPY,OFU,ODC,   DNCK0070
     &    ODY,ODU,SQ6,OSOC,OSCY,SR,KEYDS2,CO1,PSQ6,KLAST,DPTT,DPT,      DNCK0080
     &    FPT,TNPTFG,KXLLST,LRULST,IOPT2,USE)                           DNCK0090
C                                                                       DNCK0100
      REAL K,DU0(2),FU0(2),KP,DSINC,KLAST,KEYDS2                        DNCK0110
      DIMENSION ESE(3)                                                  DNCK0120
      LOGICAL TNPTFG,POSTFG                                             DNCK0130
C 1. ....LOAD ....                                  ... .. ....RULE 1.0 DNCK0140
      IF((KXL.EQ.1).OR.(KXL.EQ.3)) THEN                                 DNCK0150
        LRULE=10                                                        DNCK0160
        IF(ABS(DP).LE.DC) THEN                                          DNCK0170
          K=SOC                                                         DNCK0180
          USE=K                                                         DNCK0190
          IOPT2=3                                                       DNCK0200
          RETURN                                                        DNCK0210
        ELSE IF((ABS(DP).GT.DC) .AND.(ABS(DP).LE.DY)) THEN              DNCK0220
          DDT=SIGN(DC,DP)                                               DNCK0230
          K=SCY                                                         DNCK0240
          IF(ABS(FP).LE.(0.7*FY)) IOPT2=2                               DNCK0250
          IF(ABS(FP) GT.(0.7*FY)) IOPT2=1                               DNCK0260
        ELSE IF((ABS(DP) GT.DY).AND.(ABS(DP) LE.DU)) THEN               DNCK0270
          DDT=SIGN(DY,DP)                                               DNCK0280
          K=SYU                                                         DNCK0290
          IOPT2=1                                                       DNCK0300
        ELSE IF(ABS(DP) GT.DU .AND. ABS(DP).LE.DP) THEN                 DNCK0310
          DDT=SIGN(DU,DP)                                               DNCK0320
          K=SUP                                                         DNCK0330
          IOPT2=1                                                       DNCK0340
        ELSE IF(ABS(DP).GT DP .AND. ABS(DP).LE.(20*DY)) THEN            DNCK0350
          DDT=SIGN(DP,DP)                                               DNCK0360
          K=((0.1*FU)-FP)/((20*DY)-DP)                                  DNCK0370
          USE=OSOC                                                      DNCK0380
        ELSE IF(ABS(DP).GT.(20*DY)) THEN                                DNCK0390
          DDT=SIGN(20*DY,DP)                                            DNCK0400
          K=0.0001*SCY                                                  DNCK0410
          USE=OSOC                                                      DNCK0420
        END IF                                                          DNCK0430
        IF(KLAST.EQ.K) RETURN                                           DNCK0440
        IF(TNPTFG) THEN                                                 DNCK0450
          FPPP=FPP                                                      DNCK0460
          FP=FPPP-(DDT-DPP)*KLAST-(DP-DDT)*K                            DNCK0470
          FPT=FPPP+(DDT-DPP)*KLAST                                      DNCK0480
          DPT=DDT                                                       DNCK0490
        END IF                                                          DNCK0500
        RETURN                                                          DNCK0510
      END IF                                                            DNCK0520
C 2. ....UNLOAD ....                                                    DNCK0530
      IF((KXL.EQ.2).OR.(KXL.EQ 4)) THEN                                 DNCK0540
        IOPT2=3                                                         DNCK0550
        IF(DM.LT.CDC  AND. DMAX.LT.CDC) THEN                            DNCK0560
C .....                                  .....RULE 2.1                  DNCK0570
          LRULE=21                                                      DNCK0580
          K=SOC                                                         DNCK0590
          IF(KXLLST.EQ.5 .OR. KXLLST.EQ.6 .OR. KXLLST.EQ.7              DNCK0600
     &       .OR. KXLLST.EQ 8) THEN                                     DNCK0610
            DP1=DP                                                      DNCK0620
            FP1=FP                                                      DNCK0630
            CALL DDISSI(3,ESE,PSE,PSEOLD,DPP,FPP,DPT,0. ,AREA1,         DNCK0640
     &                  K,USE,KXL,PC,DC,FY,DY,DM,DMAX,CDC,SOB,FU00)     DNCK0650
            FP=0.+K*DPTT                                                DNCK0660
            DPTT=0                                                      DNCK0670
            DPP=DPT                                                     DNCK0680
            FPP=0.                                                      DNCK0690
            DP=DP1                                                      DNCK0700
            FP=FP1                                                      DNCK0710
          END IF                                                        DNCK0720
          IOPT2=3                                                       DNCK0730
          RETURN                                                        DNCK0740
        END IF                                                          DNCK0750
C                                                                       DNCK0760
        IF(LRULE.EQ.54 .OR. LRULE.EQ.23)   THEN                         DNCK0770
          K=KLAST                                                       DNCK0780
          RETURN                                                        DNCK0790
        END IF                                                          DNCK0800
C                                                                       DNCK0810
        IF(KXLLST.EQ.5  OR. KXLLST.EQ.6 .OR. KXLLST.EQ.7 .OR.          DNCK0820
     &     KXLLST.EQ 8 .OR. KXLLST.EQ.9  OR. KXLLST.EQ.10) THEN        DNCK0830
          SK1=(0.-FPP)/(DPP/3 DPP)                                      DNCK0840
          K=SK1                                                         DNCK0850
          IF((DM.GE.CDC) .AND.(ABS(KEYDS2).LE.DC)) THEN                 DNCK0860
C   .  .                                                                DNCK0870
            IF(KXLLST.EQ.7 .OR  KXLLST EQ.8 .OR.                        DNCK0880
     &         KXLLST.EQ.9 .OR. KXLLST.EQ.10) THEN                      DNCK0890
              K=MAX(KP,SK1)                                             DNCK0900
            END IF                                                      DNCK0910
            FP=FPP+(DP-DPP)*K                                           DNCK0920
C .....                                  .....RULE 5.4                  DNCK0930
            LRULE=54                                                    DNCK0940
            IF((FP*FPP) LT.0) THEN                                      DNCK0950
              IF(FPP.GT.0) K=FC/(DC-DPT)                                DNCK0960
              IF(FPP LT 0) K=-FC/(-DC-DPP)                              DNCK0970
              FP=0 +K*DPTT                                              DNCK0980
              IF(KKL.EQ 2) KXL=5                                        DNCK0990
              IF(KKL.EQ.4) KXL=6                                        DNCK1000
            END IF                                                      DNCK1010
            RETURN                                                      DNCK1020
          END IF                                                        DNCK1030
          IF((ABS(FPP) LE.(0.7*FY)) AND.(KXLLST.EQ.7  OR.               DNCK1040
     &      KXLLST.EQ.8  OR. KXLLST.EQ.9 .OR. KXLLST EQ.10)) THEN       DNCK1050
            K=MAX(KLAST,OSOB)                                           DNCK1060
            K=MAX(K,SK1)                                                DNCK1070
C .....                                  ...RULE 2.3                    DNCK1080
            LRULE=23                                                    DNCK1090
            USE=K                                                       DNCK1100
            FP=FPP+(DP-DPP)*K                                           DNCK1110
            IF((FP*FPP).LT.0.) THEN                                     DNCK1120
              DPT=DPP+(0.-FPP)/K                                        DNCK1130
              CALL NEWSTI(ODC,ODY,ODU,OPC,OPY,OFU,AREA1,DM,FC,FY,       DNCK1140
     &                    FU,FP,DC,DY,DU,DP,HO,H,OSOC,OSCY,SOC,         DNCK1150
     &                    SCY,SYU,SUP,SR,SOB,SOE,SQ6,PSQ6,DPT,SDP)      DNCK1160
              IF(SQ6.LE.0.3  AND. SQ6.GT.0 ) THEN                       DNCK1170
                DJP=SIGN(DC,-DPT)                                       DNCK1180
                FJP=SIGN(FC,DJP)                                        DNCK1190
                K=FJP/(DJP-DPT)                                         DNCK1200
              ELSE                                                      DNCK1210
                K=SR                                                    DNCK1220
              END IF                                                    DNCK1230
              IOPT2=3                                                   DNCK1240
              IF(KXL.EQ.2) KXL=5                                        DNCK1250
              IF(KXL.EQ.4) KXL=6                                        DNCK1260
              DPP=DPT                                                   DNCK1270
              FPP=0.                                                    DNCK1280
              FP=0.+(DP-DPT)*K                                          DNCK1290
              RETURN                                                    DNCK1300
            END IF                                                      DNCK1310
            IOPT2=3                                                     DNCK1320
            RETURN                                                      DNCK1330
          END IF                                                        DNCK1340
        END IF                                                          DNCK1350
        IF(KXL.NE.KXLLST) THEN                                          DNCK1360
          IF(POSTFG) THEN                                               DNCK1370
            FPPP=FU0(1)                                                 DNCK1380
            GGGG=FU0(2)                                                 DNCK1390
          ELSE                                                          DNCK1400
            FPPP=FU0(2)                                                 DNCK1410
            GGGG=FU0(1)                                                 DNCK1420
          END IF                                                        DNCK1430
          IF(DSINC.LT.0.) THEN                                          DNCK1440
            FU00=FPPP                                                   DNCK1450
            FU09=0.85*FPPP                                              DNCK1460
            FU17=FPPP/6                                                 DNCK1470
          END IF                                                        DNCK1480
          IF(DSINC.GT.0.) THEN                                          DNCK1490
            FU00=GGGG                                                   DNCK1500
            FU09=0.85*GGGG                                              DNCK1510
            FU17=GGGG/6                                                 DNCK1520
          END IF                                                        DNCK1530
        END IF                                                          DNCK1540
        IF((FP*FPP).LT.0) THEN                                          DNCK1550
          IF(FPP.GT.0) K=FC/(DC-DPT)                                    DNCK1560
          IF(FPP.LT 0) K=-FC/(-DC-DPP)                                  DNCK1570
          FP=0.+K*DPTT                                                  DNCK1580
          IF(KXL.EQ.2) KXL=5                                            DNCK1590
          IF(KXL.EQ.4) KXL=6                                            DNCK1600
        END IF                                                          DNCK1610
        IF((KXLLST.EQ.7 .OR. KXLLST.EQ 8 .OR. KXLLST.EQ.9               DNCK1620
     &    OR. KXLLST.EQ.10).AND.(ABS(FPP).LT.ABS(FU00))) THEN           DNCK1630
          FU09=FPP*0.85                                                 DNCK1640
          FU17=FPP/6                                                    DNCK1650
        END IF                                                          DNCK1660
        IF(ABS(FU17*6).LE.(0.70*FY)) THEN                               DNCK1670
          IF((ABS(FP).GT.FC/3).AND.(ABS(FP).LE.(0.70*FY))) THEN         DNCK1680
C .....                                                  ....RULE 2.2   DNCK1690
            LRULE=22                                                    DNCK1700
            K=OSOC                                                      DNCK1710
            IF(K.NE.KLAST) THEN                                         DNCK1720
              FP=FPP+(DP-DPP)*K                                         DNCK1730
            END IF                                                      DNCK1740
          END IF                                                        DNCK1750
          IF(ABS(FP).LE.FC/3) THEN                                      DNCK1760
C .....  .....  .....  .....  .....  .....  .....  .....RULE 2 3         DNCK1770
            LRULE=23                                                    DNCK1780
            IF(LRULE.NE.LRULST) TNPTFG=.TRUE.                           DNCK1790
            IF(TNPTFG) THEN                                             DNCK1800
              IF(ABS(DP).GT.(20*DY)) THEN                               DNCK1810
                K=SOC                                                   DNCK1820
              ELSE                                                      DNCK1830
                FPT=SIGN(FC/3,FPP)                                      DNCK1840
                DPT=DPP+(FPT-FPP)/KLAST                                 DNCK1850
                DCP=SIGN(DC,DPP)                                        DNCK1860
                FCP=SIGN(FC,DCP)                                        DNCK1870
                K=(FCP-FPT)/(DCP DPT)                                   DNCK1880
                IF(K.GT.OSOC) THEN                                      DNCK1890
                  FCP=0                                                 DNCK1900
                  DCP=DPP/3                                             DNCK1910
                  K=(0.-FPT)/(DCP-DPT)                                  DNCK1920
                  USE=K                                                 DNCK1930
                END IF                                                  DNCK1940
                FPT=FPP+(DPT-DPP)*KLAST                                 DNCK1950
                FP=FPP+(DPT-DPP)*KLAST+(DP-DPT)*K                       DNCK1960
              END IF                                                    DNCK1970
              RETURN                                                    DNCK1980
            END IF                                                      DNCK1990
          END IF                                                        DNCK2000
          RETURN                                                        DNCK2010
        ELSE                                                            DNCK2020
          IF((ABS(FP).GT.ABS(FU17)).AND.(ABS(FP) LE.ABS(FU17*6)         DNCK2030
     &      .OR. KLAST.LT.0.)) THEN                                     DNCK2040
            FPP=FU09                                                    DNCK2050
C .....  .....  .....  .....  .....  .....  .....  .....RULE 2.5         DNCK2060
            LRULE=25                                                    DNCK2070
            IF(LRULE.NE.LRULST) THEN                                    DNCK2080
              DDPP=SIGN(DC,-DPP)                                        DNCK2090
              FPPP=SIGN(FC,DDPP)                                        DNCK2100
              SKK1=(FPPP-FPP)/(DDPP-DPP)                                DNCK2110
              K=MAX(SKK1,OSOB)                                          DNCK2120
              FP=FPP+(DP-DPP)*K                                         DNCK2130
            END IF                                                      DNCK2140
          END IF                                                        DNCK2150
          IF((ABS(ABS(FP)-ABS(FU17)).LT.ABS(K*DSINC)).AND.              DNCK2160
     &      (ABS(FP).LT.ABS(FU17))) THEN                                DNCK2170
            FPT=FU17                                                    DNCK2180
            DPT=DPP+(FPT-FPP)/K                                         DNCK2190
            TNPTFG=.TRUE.                                               DNCK2200
          END IF                                                        DNCK2210
          IF((TNPTFG).OR.(ABS(FP).LE.ABS(FU17))) THEN                   DNCK2220
C .....  .....  .....  .....  .....  .....  .....  .....RULE 2 6         DNCK2230
            LRULE=26                                                    DNCK2240
            IF(TNPTFG) THEN                                             DNCK2250
              FVP=SIGN(FY,-DP)                                          DNCK2260
              DYP=SIGN(DY,-DP)                                          DNCK2270
              K=(FVP-FPT)/(DYP-DPT)                                     DNCK2280
              FP=FPP+(DPT-DPP)*KLAST+(DP-DPT)*K                         DNCK2290
              FPT=FPP-(DPT-DPP)*KLAST                                   DNCK2300
              IF((FP*FPP).LT.0.) THEN                                   DNCK2310
                DPT=DPP+(0.-FPP)/K                                      DNCK2320
                CALL NEWSTI(ODC,ODY,ODU,OPC,OPY,OFU,AREA1,DM,FC,FY,     DNCK2330
     &                      FU,FP,DC,DY,DU,DP,HO,H,OSOC,OSCY,SOC,SCY,   DNCK2340
     &                      SYU,SUP,SR,SOB,SOE,SQ6,PSQ6,DPT,SDP)        DNCK2350
                IF(SQ6.LE 0.3 .AND. SQ6.GT 0.) THEN                     DNCK2360
                  DJP=SIGN(DC,-DPT)                                     DNCK2370
                  FJP=SIGN(FC,DJP)                                      DNCK2380
                  K=FJP/(DJP DPT)                                       DNCK2390
                ELSE                                                    DNCK2400
                  K=SR                                                  DNCK2410
                END IF                                                  DNCK2420
                IOPT2=3                                                 DNCK2430
                IF(KXL.EQ.2) KXL=5                                      DNCK2440
                IF(KXL.EQ.4) KXL=6                                      DNCK2450
                DPP=DPT                                                 DNCK2460
                FPP=0.                                                  DNCK2470
                FP=0.+(DP-DPT)*K                                        DNCK2480
                RETURN                                                  DNCK2490
              END IF                                                    DNCK2500
            END IF                                                      DNCK2510
            RETURN                                                      DNCK2520
          END IF                                                        DNCK2530
          RETURN                                                        DNCK2540
        END IF                                                          DNCK2550
      END IF                                                            DNCK2560
C 3. .....REVERSAL LOADING . ...                                        DNCK2570
      IF((KXL.EQ.5).OR (KXL EQ.6)) THEN                                 DNCK2580
        IF((SQ6.LE.CO1).OR (LRULE.EQ.31)) THEN                          DNCK2590
          LRULE=31                                                      DNCK2600
          IF(DM.LT.DC  AND. DMAX LT.DC) THEN                            DNCK2610
            K=KLAST                                                     DNCK2620
            PSE=0.                                                      DNCK2630
            DPP=0.                                                      DNCK2640
            FPP=0.                                                      DNCK2650
            USE=K                                                       DNCK2660
```

```
          IOPT2=3
          IF(KKL.EQ.5) KKL=3
          IF(KKL.EQ.6) KKL=1
          RETURN
        END IF
        IF(KKL.NE.KKLLST) THEN
          DJP=SIGN(DC, DPT)
          FJP=SIGN(FC,DJP)
          IF(DPTT.NE.0) THEN
            K=FJP/(DJP-DPT)
          ELSE
            K=FJP/(DJP-DPP)
          END IF
          IF(DPTT.NE.0) THEN
            IOPT2=3
            CALL DDISSI(IOPT2,ESE,PSE,PSEOLD,DPP,FPP,DPT,0. ,AREA1,
     &                  K,USE,KKL,FC,DC,FY,DY,DM,DMAX,ODC,SOB,FU00)
            FP=0.+K*DPTT
            DPTT=0.
            DPP=DPT
            FPP=0
          END IF
        END IF
        IF((DP*FP).GT.0.) THEN
          USE=(0.-FP)/(DP/3-DP)
        END IF
        IOPT2=3
        RETURN
      END IF
C . ....... ....... ..... ................... ... .....RULE 3.2
      LRULE=32
      IF((FP*FPP).LT.0.) THEN
        K=SR
      END IF
      IF(DPTT.NE.0) THEN
        IOPT2=3
        CALL DDISSI(IOPT2,ESE,PSE,PSEOLD,DPP,FPP,DPT,0. ,AREA1,
     &              K,USE,KKL,FC,DC,FY,DY,DM,DMAX,ODC,SOB,FU00)
        FP=0.+K*DPTT
        DPTT=0.
        DPP=DPT
        FPP=0.
      END IF
      IF((DP*FP).GT.0.) THEN
        USE=(0.-FP)/(DP/3-DP)
      END IF
      IOPT2=3
      RETURN
     END IF
C4. . .RELOADING AFTER UNLOADING......
     IF((KKL.EQ.7) OR.(KKL.EQ.8)) THEN
      IF(DM.LT.ODC .AND. DMAX.LT.ODC) THEN
        K=KLAST
        USE=K
        IOPT2=3
        IF(KKL.EQ.7) KKL=3
        IF(KKL.EQ.8) KKL=1
        RETURN
      END IF
      IF(KKL.NE.KKLLST) THEN
        IF(POSTFG) THEN
          FFFF=FU0(1)
          GGGG=FU0(2)
          DDDD=DU0(1)
          EEEE=DU0(2)
        ELSE
          FFFF=FU0(2)
          DDDD=DU0(2)
          GGGG=FU0(1)
          EEEE=DU0(1)
        END IF
        IF(DSINC.GT.0) THEN
          FU09=0.9*FFFF
          IF(ABS(FFFF).LT.0.7*FY) DPT=DDDD-(FFFF-FU09)/OSOC
          IF(ABS(FFFF).GE.0.7*FY) DPT=DDDD
        END IF
        IF(DSINC.LT.0) THEN
          FU09=0.9*GGGG
          IF(ABS(GGGG).LT.0.7*FY) DPT=EEEE-(GGGG-FU09)/OSOC
          IF(ABS(GGGG).GE.0.7*FY) DPT=EEEE
        END IF
        IF(LRULE.EQ.54) THEN
          KP=KLAST
        ELSE
          SKP=(FU09-FPP)/(DPT-DPP)
          KP=MAX(SKP,KLAST)
        END IF
        TNPTFG=.TRUE.
      END IF
      IF(ABS(DP).LE.ABS(DPT)) THEN
        LRULE=41
        K=KP
        USE=K
        IOPT2=3
        IF(TNPTFG) THEN
          FP=FPP+(DP-DPP)*K
        END IF
      ELSE
        IF(LRULST.EQ.42) GO TO 688
        IF(LRULST.EQ.54) THEN
          KP=KLAST
        END IF
        IF(LRULST.NE.41) THEN
          K=MAX(0.5*KP,KLAST)
          FP=FPP+(DP-DPP)*K
          IOPT2=1
          DPP=DPT
          FPP=FPT
          LRULE=42
          TNPTFG=.FALSE.
          GO TO 688
        END IF
C ........ ....... ... . . ... .. ..... ......... .. ... RULE 4.2
        LRULE=42
        K=0.5*KP
        IOPT2=1
        IF(K.NE.KLAST) TNPTFG=.TRUE.
        IF(TNPTFG) THEN
          FP=FPP+(DPT-DPP)*KLAST+(DP-DPT)*K
          FPT=FPP+(DPT-DPP)*KLAST
        END IF
      END IF
  688 CALL DINTER(FC,FY,FU,FP,FPP,DC,DY,DU,DF,DPP,SOC,SCY,SYU,SUF,
     &            K,FP,DP,DSINC,KKL,LRULE,POSTFG,DU0,FU0)
      IF(DM.GE.ODC .AND. ABS(KEYDS2).LE.DC) THEN
        SK1=(0.-FP)/(DP/3-DP)
        USE=MAX(KP,SK1)
        IOPT2=3
      ELSE IF(ABS(FP).LE.(0.7*FY)) THEN
        SK1=(0.-FP)/(DP/3-DP)
        USE=MAX(KLAST,OSOB)
        USE=MAX(USE,SK1)
        IOPT2=3
      ELSE IF(ABS(FP).GT.(0.7*FY)) THEN
        IOPT2=1
      END IF
C
      RETURN
     END IF
  99 FORMAT(2F11.5,I6,F9.4,I3)
 199 FORMAT(2F11.5,I4,I5)
 299 FORMAT(5F12.4)
     RETURN
     END
C---------------------------------------------------------------------
C --  SUBROUTINE DCURRD SUBRTN (TO COMPUTE CURRENT DISPL,DM,DMAX)
C ---------------------------------------------------------------------
      SUBROUTINE DCURRD(KKL,DSINC,FP,FPP,DM,FC,FY,FU,PP,K,DP,DMAX,
     &       DPP,DC,DY,DU,CYCLEN,AREA1,FMAX,TNPTFG,ODC,ODY,ODU,OFU,
     &       DF,HO,H,OSOC,POSTFG,OSCY,SOC,SCY,SYU,SUF,SR,SOB,SOE,
     &       SQ6,OFC,OFY,PSQ6,DPT,DPTT,KKLLST,USE,PSEOLD,UPOS,ESE,PSE,
     &       CSE,EXCR,IOPT2)
C
C DPP ----DISPL BEFORE DISPL INCREMENT ADDED
      LOGICAL TNPTFG,POSTFG
      INTEGER CYCLEN
      REAL K
      DIMENSION ESE(3),EXCR(6),UPOS(3)
C
      IF ((KKLLST.EQ.1 .AND. DSINC.GT.0.) .OR
     &    (KKLLST.EQ.3 .AND. DSINC.LT.0.)) THEN
        IF((ABS(DP)-DC).LT.ABS(DSINC).AND.ABS(DP).GT.DC) TNPTFG=.TRUE.
        IF((ABS(DP)-DY).LT.ABS(DSINC).AND.ABS(DP).GT.DY) TNPTFG=.TRUE.
        IF((ABS(DP)-DU).LT.ABS(DSINC).AND.ABS(DP).GT.DU) TNPTFG=.TRUE.
        IF((ABS(DP)-DF).LT.ABS(DSINC).AND.ABS(DP).GT.DF) TNPTFG=.TRUE.
        IF((ABS(DP)-(20*DY)).LT.ABS(DSINC).AND.ABS(DP).GT.(20*DY))
     &     TNPTFG=.TRUE.
        GO TO 131
      END IF
      IF((KKL.EQ.5 .OR. KKL.EQ.6) .AND. ((DP*FP).GT.0 )) THEN
        IF((ABS(DP)-DC).LT.ABS(DSINC)) THEN
          IF(ABS(DP).GT.DC) THEN
            TNPTFG=.TRUE.
            IF(KKL.EQ.5) KKL=3
            IF(KKL.EQ.6) KKL=1
          END IF
        END IF
        GO TO 131
      END IF
  131 DMAX=MAX(ABS(DP),DMAX)
C
      IF((FP.EQ.0) OR.((FP*FPP).LT.0)) THEN
        DM=MAX(DM,DMAX)
C DPTT,DPT
C            (0,DPT)     |(0,0)
C-------------*---------|------------
C        *    ->.DPTT :<- |
C     ->:   DSINC    :<- |
C
        IF((FP*FPP).LT. 0 ) THEN
          DPT=DPP-FPP/K
          DPTT=DP-DPT
          CALL DNE(0,UPOS,EXCR,ODY,DP,ESE,PSE,PSEOLD,CSE)
C ----- IF K=SUF AND REACH FORCE=0 THEN SET FORCE=0 AND DP=DPTT
C         ,AFTER THAT K=0
          IF(K.EQ.(-0.0001*SCY)) THEN
            DP=DP
            FP=0.
            K=0.
            STOP
          END IF
          IF(DM.LT.ODC .AND. DMAX.LT.ODC) THEN
            IOPT2=3
          ELSE
            IOPT2=2
          END IF
C
          CALL DDISSI(IOPT2,ESE,PSE,PSEOLD,DPP,FPP,DPT,0. ,AREA1,
     &                K,USE,KKL,FC,DC,FY,DY,DM,DMAX,ODC,SOB,FU00)
          IF((POSTFG) AND.(FPP.LT.0)) THEN
            CYCLEN=CYCLEN+1
            FMAX=0.
            DMAX=ABS(DP)
          END IF
          IF(.NOT. POSTFG .AND. FPP.GT.0 ) THEN
            CYCLEN=CYCLEN+1
            FMAX=0.
            DMAX=ABS(DP)
          END IF
        END IF
        IF(DM.GT.ODC) THEN
          CALL NEWSTI(ODC,ODY,ODU,OFC,OFY,OFU,AREA1,DM,FC,FY,
     &                FU,FP,DC,DY,DU,DP,HO,H,OSOC,OSCY,SOC,
     &                SCY,SYU,SUF,SR,SOB,SOE,SQ6,PSQ6,DPT,SDP)
        END IF
        IF(KKL.EQ.2) KKL=5
        IF(KKL.EQ.4) KKL=6
       END IF
       IF(((KKL.EQ.7).AND.(KKLLST.EQ.4)).OR.
     &  ((KKL.EQ.8).AND.(KKLLST.EQ.2))) THEN
        CYCLEN=CYCLEN+1
        DM=MAX(DM,DMAX)
       END IF
       RETURN
       END
C---------------------------------------------------------------------
C --  NEWSTIF SUBRTN (FIND STIFFNESS WHEN PASSING HORIZ. AXIS (FORCE=0)
C ---------------------------------------------------------------------
      SUBROUTINE NEWSTI(ODC,ODY,ODU,OFC,OFY,OFU,AREA1,DM,FC,FY,
     &            FU,FP,DC,DY,DU,DP,HO,H,OSOC,OSCY,SOC,
     &            SCY,SYU,SUF,SR,SOB,SOE,SQ6,PSQ6,DPT,SDP)
C
      SQ6=AREA1/(OFU*ODU)
      TOL=1E-10
      IF(ABS(DM).LT.ODC .AND. SQ6.LT.TOL) SQ6=0.
      PSDP=SDP
      SDP=DPT
      ALPHA=HO/H
      IF((SQ6.GT.0.).AND.(SQ6.LE.0.3)) THEN
        DC=DC
        FC=FC
      ELSE
        B=-0.25473*SQ6
        SR=(0.27798*10**B)*OSOC
        DC=ABS(SDP)*0.25
        DC=MAX(DC,ODC)
        IF(DC.GT.(1.5*ODC)) THEN
          DC=1.5*ODC
        END IF
        IF(ABS(SDP).GT.DU) THEN
          FC=0.98*FC
          SR=FC/(ABS(SDP)-DC)
        END IF
C
        FC=SR*(ABS(SDP)-DC)
      END IF
      IF(SQ6.NE.0.) THEN
        B=-0.20102*SQ6
        BB=1.1578*10**B
        IF(BB.GT.1.) BB=1
        SCY=BB*OSCY
      END IF
      COEF1=(OFY-OFC)/(ODY-ODC)
      COEF2=(OFU-OFY)/(ODU-ODY)
      SYU=(COEF2/COEF1)*SCY
      IF(SQ6.LE.0 ) THEN
        DY=DY
      ELSE IF((SQ6.GT.0.) AND.(SQ6.LE.0.312)) THEN
        DY=ODY
      ELSE
        DY=(1.5299+0.4546*LOG(SQ6))*ODY
      END IF
      IF(SQ6.EQ.0.) THEN
        DU=DU
      ELSE
        DU=DY*(ODU/ODY)
        DU=MAX(DU,1.5*DY)
      END IF
      FY=FC+(DY-DC)*SCY
      FU=FY+(DU-DY)*SYU
      SUF=-0.5*SCY
```

```
      FP=(PC+FY)/2                                                      NEWS0570
      DP=DU*(FP-FU)/SUF                                                 NEWS0580
      SOC=PC/DC                                                         NEWS0590
      SOY=FY/DY                                                         NEWS0600
      SCU=(FU-PC)/(DU-DC)                                               NEWS0610
      XA=FY/SOC                                                         NEWS0620
      XB=(XA+DY)/2                                                      NEWS0630
      SOB=FY/XB                                                         NEWS0640
      XD=FU/SOY                                                         NEWS0650
      XE=(XD+DU)/2                                                      NEWS0660
      SOE=FU/XE                                                         NEWS0670
      PSQ6=SQ6                                                          NEWS0680
      RETURN                                                            NEWS0690
      END                                                               NEWS0700
C-----------------------------------------------------------------------DDIS0010
C ---  SUBROUTINE DDISSI  SUBRTN (AREA1=ENERGY DISSIPATION)             DDIS0020
C ---  CALCULATE THE ELASTIC STRAIN ENERGY(ESE(1)) AND                  DDIS0030
C ---  PLASTIC STRAIN ENERGY (PSE)                                      DDIS0040
C -------------------------------------------------------------------- DDIS0050
      SUBROUTINE DDISSI(IOPT2,ESE,PSE,PSEOLD,DPP,FPP,DP,FP,AREA1,       DDIS0060
     &                  K,USE,KKL,PC,DC,FY,DY,DM,DMAX,ODC,SCB,FU00)     DDIS0070
      REAL K                                                            DDIS0080
      DIMENSION ESE(3)                                                  DDIS0090
C                                                                       DDIS0100
C AREA1 REPRESENTING AREA ENCLOSED BY THE HYSTERESIS RESPONSE           DDIS0110
C TSE,ESE(1),ESE(2),ESE(3),PSE,PSEOLD SEE STRENG SUBRTN                 DDIS0120
C USE     = EQUIVALENT UNLOADING STIFFNESS                              DDIS0130
C.....  ..... ..... ..... ..... ..... .....                            DDIS0140
      IF(IOPT2.EQ 1  AND. K.NE.0.) THEN                                 DDIS0150
C  .. CALCULATE THE EQUIVALENT UNLOADING STIFFNESS                      DDIS0160
        IF((ABS(FU00).GT.0.7*FY) .OR. (KKL.EQ.7 .OR. KKL.EQ 8 .OR.      DDIS0170
     &      KKL.EQ.9 .OR. KKL.EQ.10)) THEN                              DDIS0180
          DDPP=SIGN(DC,-DP)                                             DDIS0190
          FPPP=SIGN(FC,DDPP)                                            DDIS0200
          SKK1=(FPPP-FU00*0.85)/(DDPP-DP)                               DDIS0210
          SK1=MAX(SKK1,SCB)                                             DDIS0220
          DP1=DP+(FP/6-FP*0.85)/SK1                                     DDIS0230
          FYP=SIGN(FY,-DP1)                                             DDIS0240
          DYP=SIGN(DY,FYP)                                              DDIS0250
          SK2=(FYP-FP/6)/(DYP-DP1)                                      DDIS0260
          DP2=DP1+(0.-FP/6)/SK2                                         DDIS0270
          AA=0.5*(0.85*FP+FP/6)*(DP-DP1)+0.5*FP/6*(DP1-DP2)             DDIS0280
          USE=(FP**2)/(2*ABS(AA))                                       DDIS0290
        ELSE IF(ABS(FP).LE.0.7*FY) THEN                                 DDIS0300
          SK1 = PC/DC                                                   DDIS0310
          FP1=SIGN(FC/1,FP)                                             DDIS0320
          DP1=DP+(FP1-FP)/SK1                                           DDIS0330
          DCP = SIGN(DC,-DP)                                            DDIS0340
          FCP = SIGN(FC,DCP)                                            DDIS0350
          SK2 = (FCP-FP1)/(DCP-DP1)                                     DDIS0360
          DP2 = DP1 + (0.-FP1)/SK2                                      DDIS0370
          AA=0.5*(FP1+FP)*(DP-DP1)+0.5*FP1*(DP1-DP2)                    DDIS0380
          USE = (FP**2)/(2*ABS(AA))                                     DDIS0390
        END IF                                                          DDIS0400
      END IF                                                            DDIS0410
      IF(IOPT2.EQ.2  AND. K.NE.0.) THEN                                 DDIS0420
        IF(DM.LE.ODC .AND. DMAX.LE.ODC .AND.(KKL EQ.1  .OR.             DDIS0430
     &     KKL.EQ.3 .OR  KKL EQ.5 .OR. KKL.EQ.6)) THEN                  DDIS0440
          USE=K                                                         DDIS0450
        ELSE IF((DP*FP).GT. 0.) THEN                                    DDIS0460
          USE=(0.-FP)/(DP/3-DP)                                         DDIS0470
        END IF                                                          DDIS0480
      END IF                                                            DDIS0490
      CALL STRENG(ESE,PSE,PSEOLD,FP,FPP,DP,DPP,USE)                     DDIS0500
      AREA1=ESE(1)+PSE                                                  DDIS0510
      RETURN                                                            DDIS0520
      END                                                               DDIS0530
C-----------------------------------------------------------------------DINT0010
C --- SUBROUTINE DINTER SUBRTN (CHECK IF NEW CURRENT TRACK INTERSECT     DINT0020
C --- ;IF YES,FIND CONNECT PT.(UPDATED CURRENT PT )                     DINT0030
C ---------------------------------------------------------------------- DINT0040
      SUBROUTINE DINTER(PC,FY,FU,FP,FPP,DC,DY,DU,DP,DPP,SOC,SCY,SYU,    DINT0050
     &            SUP,K,FP,DP,DSINC,KKL,LRULE,POSTPG,DUC,FU0)           DINT0060
C                                                                       DINT0070
      REAL K,KJ,DUC(4),FU0(4)                                           DINT0080
      LOGICAL POSTPG                                                    DINT0090
C                                                                       DINT0100
      IF(ABS(DP).LT.DC) THEN                                            DINT0110
        KJ=FP/DP                                                        DINT0120
        IF(KJ.LT SOC) RETURN                                            DINT0130
        IF(KJ.EQ.SOC) THEN                                              DINT0140
          IF(DSINC.GT.0) KKL=1                                          DINT0150
          IF(DSINC.LT.0) KKL=3                                          DINT0160
          RETURN                                                        DINT0170
        END IF                                                          DINT0180
        IF(KJ.GT.SOC .AND. K.GT.SOC) THEN                               DINT0190
          IF(LRULE.EQ.51 .OR. LRULE.EQ.52) THEN                         DINT0200
            X=(K*DPP-FPP)/(K-SCC)                                       DINT0210
            Y=X*SOC                                                     DINT0220
            ADP=SIGN(X,DPP)                                             DINT0230
            APP=SIGN(Y,DPP)                                             DINT0240
            FP=APP+SOC*(DP-ADP)                                         DINT0250
            IF(DSINC.GT 0) KKL=1                                        DINT0260
            IF(DSINC.LT.0) KKL=3                                        DINT0270
            K=SOC                                                       DINT0280
          END IF                                                        DINT0290
C ..... ...FOR RELOAD DURING REVERSAL STAGE                             DINT0300
          IF(LRULE.EQ.41 .OR. LRULE.EQ.42) THEN                         DINT0310
            IF(DP.GT.0) THEN                                            DINT0320
              IF(POSTPG) THEN                                           DINT0330
                FPPP=FU0(1)                                             DINT0340
                GGGG=DU0(1)                                             DINT0350
              ELSE                                                      DINT0360
                FPPP=FU0(2)                                             DINT0370
                GGGG=DU0(2)                                             DINT0380
              END IF                                                    DINT0390
            ELSE                                                        DINT0400
              IF(POSTPG) THEN                                           DINT0410
                FPPP=FU0(2)                                             DINT0420
                GGGG=DU0(2)                                             DINT0430
              ELSE                                                      DINT0440
                FPPP=FU0(1)                                             DINT0450
                GGGG=DU0(1)                                             DINT0460
              END IF                                                    DINT0470
            END IF                                                      DINT0480
            IF(ABS(DP).GT.ABS(GGGG) .AND. (DP*GGGG).GT.0) THEN          DINT0490
              Y2=FPP+(GGGG-DPP)*K                                       DINT0500
              SK=(ABS(FC) ABS(Y2))/(ABS(DC)-ABS(GGGG))                  DINT0510
              FP=Y2+(DP-GGGG)*SK                                        DINT0520
              IF(DP.GT 0) KKL=6                                         DINT0530
              IF(DP LT.0) KKL=5                                         DINT0540
              K=SK                                                      DINT0550
              RETURN                                                    DINT0560
            END IF                                                      DINT0570
          END IF                                                        DINT0580
        END IF                                                          DINT0590
      END IF                                                            DINT0600
      IF((ABS(DP) GE.DC) AND.(ABS(DP).LT.DY)) THEN                      DINT0610
        IF(DP.GT.0) THEN                                                DINT0620
          IF(POSTPG) THEN                                               DINT0630
            FPPP=FU0(1)                                                 DINT0640
            GGGG=DU0(1)                                                 DINT0650
          ELSE                                                          DINT0660
            FPPP=FU0(2)                                                 DINT0670
            GGGG=DU0(2)                                                 DINT0680
          END IF                                                        DINT0690
        ELSE                                                            DINT0700
          IF(POSTPG) THEN                                               DINT0710
            FPPP=FU0(2)                                                 DINT0720
            GGGG=DU0(2)                                                 DINT0730
          ELSE                                                          DINT0740
            FPPP=FU0(1)                                                 DINT0750
            GGGG=DU0(1)                                                 DINT0760
          END IF                                                        DINT0770
        END IF                                                          DINT0780
        IF(ABS(DPP) LT.0 .AND. (LRULE.EQ.41 .OR. LRULE.EQ.42)           DINT0790
     &     AND. (DP*GGGG) GT.0) THEN                                    DINT0800
          Y2=FPP+(GGGG-DPP)*K                                           DINT0810
          SK=(ABS(FC)-ABS(Y2))/(ABS(DC)-ABS(GGGG))                      DINT0820
          IF(DP GT 0) FP=Y2+(DC-GGGG)*SK+(DP-DC)*SCY                    DINT0830
          IF(DP.LT.0) FP=Y2+(-DC-GGGG)*SK+(DP-DC)*SCY                   DINT0840
          IF(DSINC.GT.0) KKL=1                                          DINT0850
          IF(DSINC.LT.0) KKL=3                                          DINT0860
          K=SCY                                                         DINT0870
          RETURN                                                        DINT0880
        END IF                                                          DINT0890
        KJ=(ABS(FP)-FC)/(ABS(DP)-DC)                                    DINT0900
        IF(KJ.LT.SCY) RETURN                                            DINT0910
        IF(KJ.EQ.SCY) THEN                                              DINT0920
          IF(DSINC.GT.0) KKL=1                                          DINT0930
          IF(DSINC.LT.0) KKL=3                                          DINT0940
          RETURN                                                        DINT0950
        END IF                                                          DINT0960
        IF(KJ.GT.SCY) THEN                                              DINT0970
          X=(K*ABS(DPP)-DC*SCY-ABS(FPP)+FC)/(K-SCY)                     DINT0980
          Y=ABS(FPP)+K*ABS(X)-K*ABS(DPP)                                DINT0990
          IF(ABS(X).LT.DC) THEN                                         DINT1000
            X=(K*DPP-FPP)/(K-SOC)                                       DINT1010
            Y=X*SOC                                                     DINT1020
            ADP=SIGN(X,DPP)                                             DINT1030
            APP=SIGN(Y,DPP)                                             DINT1040
            IF(DP GT.0) FP=APP+SOC*(DC-ADP)+SCY*(DP-DC)                 DINT1050
            IF(DP.LT 0) FP=APP+SOC*(-DC-ADP)+SCY*(DP+DC)                DINT1060
            IF(DSINC GT.0) KKL=1                                        DINT1070
            IF(DSINC.LT.0) KKL=3                                        DINT1080
            K=SCY                                                       DINT1090
            RETURN                                                      DINT1100
          END IF                                                        DINT1110
          ADP=SIGN(X,DPP)                                               DINT1120
          APP=SIGN(Y,DPP)                                               DINT1130
          FP=APP+SCY*(DP-ADP)                                           DINT1140
          IF(DSINC.GT.0) KKL=1                                          DINT1150
          IF(DSINC LT 0) KKL=3                                          DINT1160
          K=SCY                                                         DINT1170
        END IF                                                          DINT1180
        RETURN                                                          DINT1190
      END IF                                                            DINT1200
      IF((ABS(DP).GE.DY).AND.(ABS(DP).LT.DU)) THEN                      DINT1210
        KJ=(ABS(FP)-FY)/(ABS(DP)-DY)                                    DINT1220
        IF(KJ.LT.SYU) RETURN                                            DINT1230
        IF(KJ.EQ.SYU) THEN                                              DINT1240
          IF(DSINC.GT 0) KKL=1                                          DINT1250
          IF(DSINC.LT.0) KKL=3                                          DINT1260
          RETURN                                                        DINT1270
        END IF                                                          DINT1280
        IF(KJ.GT.SYU) THEN                                              DINT1290
          X=(FY-DY*SYU-K*ABS(DPP)+ABS(FPP))/(K-SYU)                     DINT1300
          Y=ABS(FPP)+K*ABS(X)-K*ABS(DPP)                                DINT1310
          IF(ABS(X).LT.DY) THEN                                         DINT1320
            X=(K*ABS(DPP)-DC*SCY-ABS(FPP)+FC)/(K-SCY)                   DINT1330
            Y=ABS(FPP)+K*ABS(X)-K*ABS(DPP)                              DINT1340
            ADP=SIGN(X,DPP)                                             DINT1350
            APP=SIGN(Y,DPP)                                             DINT1360
            IF(DP.GT.0) FP=APP+SCY*(DY-ADP)+SYU*(DP-DY)                 DINT1370
            IF(DP.LT.0) FP=APP+SCY*(-DY-ADP)+SYU*(DP+DY)                DINT1380
            IF(DSINC.GT.0) KKL=1                                        DINT1390
            IF(DSINC.LT.0) KKL=3                                        DINT1400
            K=SYU                                                       DINT1410
            RETURN                                                      DINT1420
          END IF                                                        DINT1430
          ADP=SIGN(X,DPP)                                               DINT1440
          APP=SIGN(Y,DPP)                                               DINT1450
          FP=APP+SYU*(DP-ADP)                                           DINT1460
          IF(DSINC.GT.0) KKL=1                                          DINT1470
          IF(DSINC.LT.0) KKL=3                                          DINT1480
          X=SYU                                                         DINT1490
        END IF                                                          DINT1500
        RETURN                                                          DINT1510
      END IF                                                            DINT1520
      IF(ABS(DP).GE.DU .AND. ABS(DP).LT.DP) THEN                        DINT1530
        IF((ABS(DP).EQ.DU).AND.(ABS(FP).LT.FU)) RETURN                  DINT1540
        KJ=(ABS(FP)-FU)/(ABS(DP)-DU)                                    DINT1550
        IF((KJ.LT.0).AND.(KJ.LT.SUF)) RETURN                            DINT1560
        IF(KJ.EQ.SUP) THEN                                              DINT1570
          IF(DSINC.GT.0) KKL=1                                          DINT1580
          IF(DSINC.LT.0) KKL=3                                          DINT1590
          RETURN                                                        DINT1600
        END IF                                                          DINT1610
        IF(KJ.GT.SUP) THEN                                              DINT1620
          X=(FU-DU*SUP-K*ABS(DPP)-ABS(FPP))/(K-SUP)                     DINT1630
          Y=ABS(FPP)+X*ABS(X)-K*ABS(DPP)                                DINT1640
          IF(ABS(X).LT.DU) THEN                                         DINT1650
            X=(FY-DY*SYU-K*ABS(DPP)-ABS(FPP))/(K-SYU)                   DINT1660
            Y=ABS(FPP)+K*ABS(X)-K*ABS(DPP)                              DINT1670
            ADP=SIGN(X,DPP)                                             DINT1680
            APP=SIGN(Y,DPP)                                             DINT1690
            IF(DP.GT.0) FP=APP+SYU*(DU-ADP)+SUP*(DP-DU)                 DINT1700
            IF(DP.LT.0) FP=APP+SYU*(-DU-ADP)+SUP*(DP+DU)                DINT1710
            IF(DSINC.GT.0) KKL=1                                        DINT1720
            IF(DSINC.LT.0) KKL=3                                        DINT1730
            K=SUP                                                       DINT1740
            RETURN                                                      DINT1750
          END IF                                                        DINT1760
          ADP=SIGN(X,DPP)                                               DINT1770
          APP=SIGN(Y,DPP)                                               DINT1780
          FP=APP+SUP*(DP-ADP)                                           DINT1790
          IF(DSINC.GT.0) KKL=1                                          DINT1800
          IF(DSINC.LT.0) KKL=3                                          DINT1810
          K=SUP                                                         DINT1820
        END IF                                                          DINT1830
        RETURN                                                          DINT1840
      END IF                                                            DINT1850
C                                                                       DINT1860
      IF(ABS(DP).GE.DP  AND. ABS(DP).LT.(20*DY)) THEN                   DINT1870
        KJ=(ABS(FP)-FF)/(ABS(DP)-DP)                                    DINT1880
        DG=20*DY                                                        DINT1890
        SFG=(0.1*FU-FF)/(DG-DP)                                         DINT1900
        IF(KJ.EQ.SFG) THEN                                              DINT1910
          IF(DSINC GT.0) KKL=1                                          DINT1920
          IF(DSINC.LT.0) KKL=3                                          DINT1930
          RETURN                                                        DINT1940
        END IF                                                          DINT1950
        IF(KJ.GT.SFG) THEN                                              DINT1960
          X=(0.1*FU-DG*SFG-K*ABS(DPP)-ABS(FPP))/(K-SFG)                 DINT1970
          Y=ABS(FPP)+K*ABS(X)-K*ABS(DPP)                                DINT1980
          IF(ABS(X).LT.DU) THEN                                         DINT1990
CC  ... TRY WHETHER IT IS LOCATED IN BWN U AND P  .  ...                DINT2000
            X=(FU-DU*SUP-K*ABS(DPP)-ABS(FPP))/(K-SUP)                   DINT2010
            Y=ABS(FPP)+K*ABS(X)-K*ABS(DPP)                              DINT2020
            IF(ABS(X).LT.DU) THEN                                       DINT2030
              X=(FY-DY*SYU-K*ABS(DPP)-ABS(FPP))/(K-SYU)                 DINT2040
              Y=ABS(FPP)+K*ABS(X)-K*ABS(DPP)                            DINT2050
              ADP=SIGN(X,DPP)                                           DINT2060
              APP=SIGN(Y,DPP)                                           DINT2070
              IF(DP.GT.0) FP=APP+SYU*(DU-ADP)+SUP*(DP-DU)+SFG*(DP-DF)   DINT2080
              IF(DP.LT.0) FP=APP+SYU*(-DU-ADP)+SUP*(-DF+DU)+SFG*(DP+DF) DINT2090
              IF(DSINC GT.0) KKL=1                                      DINT2100
              IF(DSINC LT.0) KKL=3                                      DINT2110
              K=SFG                                                     DINT2120
              RETURN                                                    DINT2130
            END IF                                                      DINT2140
            ADP=SIGN(X,DPP)                                             DINT2150
            APP=SIGN(Y,DPP)                                             DINT2160
            IF(DP GT.0) FP=APP+SUP*(DP-ADP)+SFG*(DP-DF)                 DINT2170
            IF(DP.LT.0) FP=APP+SUP*(-DP-ADP)+SFG*(DP+DF)                DINT2180
            IF(DSINC.GT.0) KKL=1                                        DINT2190
            IF(DSINC.LT.0) KKL=3                                        DINT2200
            K=SFG                                                       DINT2210
            RETURN                                                      DINT2220
          END IF                                                        DINT2230
          IF(ABS(X) LT.DF  AND  ABS(X).GT.DU) THEN                      DINT2240
            X=(FP-DF*SUP-K*ABS(DPP)-ABS(FPP))/(K-SUP)                   DINT2250
```

```
            Y=ABS(FPP)+K*ABS(X)-K*ABS(DPP)                                      DINT2260
            ADP=SIGN(X,DPP)                                                     DINT2270
            APP=SIGN(Y,DPP)                                                     DINT2280
            IF(DP.GT.0) FP=APP+SUP*(DP-ADP)+SPG*(DP-DP)                         DINT2290
            IF(DP.LT.0) FP=APP+SUP*(-DP-ADP)-SPG*(DP-DP)                        DINT2300
            IF(DSINC.GT.0) KKL=1                                                DINT2310
            IF(DSINC.LT.0) KKL=3                                                DINT2320
            K=SPG                                                               DINT2330
            RETURN                                                              DINT2340
          END IF                                                                DINT2350
          ADP=SIGN(X,DPP)                                                       DINT2360
          APP=SIGN(Y,DPP)                                                       DINT2370
          FP=APP+SPG*(DP-ADP)                                                   DINT2380
          IF(DSINC.GT.0) KKL=1                                                  DINT2390
          IF(DSINC.LT.0) KKL=3                                                  DINT2400
          K=SPG                                                                 DINT2410
        END IF                                                                  DINT2420
        RETURN                                                                  DINT2430
      END IF                                                                    DINT2440
C                                                                               DINT2450
      DH=20*DY+(0. 0.1*FU)/(-0 0001*SCY)                                        DINT2460
      SGH= 0.0001*SCY                                                           DINT2470
      SPG=(0.1*FU-FP)/(20*DY-DP)                                                DINT2480
      IF(ABS(DP).GE.(20*DY)  AND. ABS(DP).LT.DH) THEN                           DINT2490
        IF(ABS(DP).EQ.(20*DY) .AND. ABS(FP).LT.(0.1*FU)) RETURN                 DINT2500
        KJ=(ABS(FP)-(0.1*FU))/(ABS(DP)-(20*DY))                                 DINT2510
        IF(KJ.LT 0 .AND. KJ.LT.SGH) RETURN                                      DINT2520
        IF(KJ.EQ.SGH) THEN                                                      DINT2530
          IF(DSINC GT.0) KKL=1                                                  DINT2540
          IF(DSINC.LT.0) KKL=3                                                  DINT2550
          RETURN                                                                DINT2560
        END IF                                                                  DINT2570
        IF(KJ.GT.SGH) THEN                                                      DINT2580
          X=(0.1*FU-20*DY*SGH+K*ABS(DPP)-ABS(FPP))/(K-SGH)                      DINT2590
          Y=ABS(FPP)+K*ABS(X)-K*ABS(DPP)                                        DINT2600
          IF(ABS(X).LT DP) THEN                                                 DINT2610
CC ... TRY WHETHER IT IS LOCATED IN BTN P AND G .....                           DINT2620
            X=(0.1*FU-DG*SPG-K*ABS(DPP)-ABS(FPP))/(K-SPG)                       DINT2630
            Y=ABS(FPP)+K*ABS(X)-K*ABS(DPP)                                      DINT2640
            IF(ABS(X).LT.DP) THEN                                               DINT2650
              X=(FU-DU+SUP+K*ABS(DPP)-ABS(FPP))/(K-SUP)                         DINT2660
              Y=ABS(FPP)+K*ABS(X)-K*ABS(DPP)                                    DINT2670
              ADP=SIGN(X,DPP)                                                   DINT2680
              APP=SIGN(Y,DPP)                                                   DINT2690
              IF(DP.GT.0) FP=APP+SUP*(DP-ADP)+SPG*(DG-DP)+SGH*(DP-DG)           DINT2700
              IF(DP.LT.0) FP=APP+SUP*(-DP-ADP)+SPG*(-DG+DP)+SGH*(DP+DG)         DINT2710
              IF(DSINC.GT.0) KKL=1                                              DINT2720
              IF(DSINC.LT.0) KKL=3                                              DINT2730
              K=SGH                                                             DINT2740
              RETURN                                                            DINT2750
            END IF                                                              DINT2760
            ADP=SIGN(X,DPP)                                                     DINT2770
            APP=SIGN(Y,DPP)                                                     DINT2780
            IF(DP.GT.0) FP=APP+SPG*(DP-ADP)+SGH*(DP-DP)                         DINT2790
            IF(DP.LT.0) FP=APP+SPG*(-DP-ADP)+SGH*(DP+DP)                        DINT2800
            FP=SIGN(FP,DP)                                                      DINT2810
            IF(DSINC.GT.0) KKL=1                                                DINT2820
            IF(DSINC.LT.0) KKL=3                                                DINT2830
            K=SGH                                                               DINT2840
            RETURN                                                              DINT2850
          END IF                                                                DINT2860
          IF(ABS(X).LT.(20*DY)) THEN                                            DINT2870
            K=(FP-DP+SPG+K*ABS(DPP)-ABS(FPP))/(K-SPG)                           DINT2880
            Y=ABS(FPP)+K*ABS(X)-K*ABS(DPP)                                      DINT2890
            ADP=SIGN(X,DPP)                                                     DINT2900
            APP=SIGN(Y,DPP)                                                     DINT2910
            IF(DP.GT.0) FP=APP+SPG*(20*DY-ADP)+SGH*(DP-20*DY)                   DINT2920
            IF(DP.LT.0) FP=APP-SPG*(-20*DY ADP)+SGH*(DP-20*DY)                  DINT2930
            IF(DSINC.GT.0) KKL=1                                                DINT2940
            IF(DSINC.LT.0) KKL=3                                                DINT2950
            K=SGH                                                               DINT2960
            RETURN                                                              DINT2970
          END IF                                                                DINT2980
          ADP=SIGN(X,DPP)                                                       DINT2990
          APP=SIGN(Y,DPP)                                                       DINT3000
          FP=APP+SGH*(DP-ADP)                                                   DINT3010
          IF(DSINC GT.0) KKL=1                                                  DINT3020
          IF(DSINC.LT.0) KKL=3                                                  DINT3030
          K=SGH                                                                 DINT3040
        END IF                                                                  DINT3050
        RETURN                                                                  DINT3060
      END IF                                                                    DINT3070
      IF(ABS(DP).GE.DH) THEN                                                    DINT3080
        K=0.                                                                    DINT3090
        IF(DSINC.GT.0) KKL=1                                                    DINT3100
        IF(DSINC.LT.0) KKL=3                                                    DINT3110
        RETURN                                                                  DINT3120
      END IF                                                                    DINT3130
      RETURN                                                                    DINT3140
      END                                                                       DINT3150
C********* HYST12 ************************************************** HYST0010
C         THIS SUBROUTINE IS USED FOR NONLINEAR BEAM-COLUMN ANALYSIS,           HYST0020
C         WHICH CAN ANALYSIS THE POSTPUCKING, UNLOADING, AND RELOADING          HYST0030
C         HYSTERISTIC LOOPS.                                                    HYST0040
C                                                                               HYST0050
C     ********************************************************************** HYST0060
      SUBROUTINE HYST12( IOPT ,NSEG  ,YS    ,EM    ,LIBN ,HN     ,UU     ,  HYST0070
     &            JSTOR,IDOF  ,WW    ,ZZ    ,INEB ,INEH ,ST     ,IREV1 ,  HYST0080
     &            IREV2,IREV3 ,IREV4 ,IECOP,ECCX0,ECCY0,NUMP  ,ELS    ,     HYST0090
     &            SMALL,RATIX0,RATIY0,TOTA ,IMD  ,ISFE ,DELTP2,S2     ,     HYST0100
     &            SP    ,ISTART,IR    ,G     ,ISTIP,IAUTO,IMATER,RATIO3,     HYST0110
     &            QRNEE,RNEEE ,                                               HYST0120
     &                  NP,IFM, BETA, ECX, ECY, U, V, U0, V0, GCD,            HYST0130
     &                  EL, D1, DD, D00, TL, XLOC, FLOC, PGLOB, XGLO,         HYST0140
     &                  E0, E1, E2, TE, PR, TIR, SEASAT, STF,MDIAG,           HYST0150
     &                  EEASAT, TRJT, DP, DUMMY, PSK, ECCX, ECCY,             HYST0160
     &                  FF1,FACTOR,DPCOPY,ASAT,B,DE,                          HYST0170
     &                  A0, B ,STR,DEP,EOP,STRP,TEP,EREP,SREP,FJL,FLP)        HYST0180
C                                                                               HYST0190
C  IOPT   - 1 : FORMULATE THE INITIAL MEMBER STIFFNESS                          HYST0200
C           - 2 : FORMULATE THE NONLINEAR MEMBER STIFFNESS                      HYST0210
C  RATIX0 - INITIAL IMPERFECTION OF THE MEMBER IN X DIRECTION                   HYST0220
C  RATIY0 - INITIAL IMPERFECTION OF THE MEMBER IN Y DIRECTION                   HYST0230
C  IECOP  - ECCENTRIC LOAD OPTION                                               HYST0240
C  WW     - ECCENTRIC DISTANCE IN X DIRECTION                                   HYST0250
C  ZZ     - ECCENTRIC DISTANCE IN Y DIRECTION                                   HYST0260
C  NUMP   - TOTAL SECTIONAL POINT ON THE SEGMENT CROSS-SECTION                  HYST0270
C  ASAT   - MEMBER GLOBAL STIFFNESS MATRIX                                      HYST0280
C  EEASAT - CONDENSED MATRIX                                                    HYST0290
C  CONMT  - RESIDUAL MATRIX                                                     HYST0300
C  DD     - MEMBER'S DISPLACEMENT INCREMENT AT EACH D.O.F.                       HYST0310
C  NSEG   - TOTAL NO OF SEGMENTS IN THE MEMBER                                  HYST0320
C  IDOF   - TOTAL D.O.F. OF THE MEMBER                                          HYST0330
C  ELS    - TOTAL LENGTH OF THE MEMBER                                          HYST0340
C  LIBN   - THE CROSS-SECTION LIBRARY NO.                                       HYST0350
C  FLP    - FLAG FOR JUDING LOCAL BUCKLING                                      HYST0360
C           0 - NO LOCAL BUCKLING                                               HYST0370
C           1-6 : LOCAL BUCKLING OCCURRED                                       HYST0380
C  NOMEM  - SYSTEM MEMBER NO.                                                   HYST0390
C  RNEEE  - INDEX FOR LIMITING NORM OF DISPLACEMENT                             HYST0400
C           0 : EEE .LT. QRNEE                                                  HYST0410
C           1 : EEE .GT. QRNEE                                                  HYST0420
C  ECX    - DISTANCE FROM REFERENCE COORDINATE (U,V,W) ORIGIN TO                HYST0430
C           ECCENTRIC LOAD LOCATION IN U DIRECTION                              HYST0440
C  ECY    - DISTANCE FROM REFERENCE COORDINATE (U,V,W) ORIGIN TO                HYST0450
C           ECCENTRIC LOAD LOCATION IN V DIRECTION                              HYST0460
C                                                                               HYST0470
      REAL INEB,INEH                                                            HYST0480
      DIMENSION BETA(NSEG),ECX(2),ECY(2),IFM(IDOF),MDIAG(IDOF+1)                HYST0490
      DIMENSION U(NSEG,NUMP),V(NSEG,NUMP),U0(NUMP),V0(NUMP)                     HYST0500
      DIMENSION GCD(NSEG,2,3),D1(IDOF),DD(IDOF),ASAT(IDOF,IDOF)                 HYST0510
      DIMENSION D00(IDOF),EL(NSEG),TL(NSEG),XLOC(NSEG,12),DP(IDOF)              HYST0520
      DIMENSION FLOC(NSEG,12),PGLOB(NSEG,12),XGLO(NSEG,12)                      HYST0530
      DIMENSION E1(NSEG,NUMP),E2(NSEG,NUMP),TE(NUMP),PR(NSEG,9)                 HYST0540
      DIMENSION E(NUMP),DE(NSEG,NUMP),SEASAT(IDOF,IDOF)                         HYST0550
      DIMENSION TIR(NSEG,9),SK(12,12),EREP(NSEG,NUMP)                           HYST0560
      DIMENSION DPCOPY(IDOF),FJL(NUMP),E0(NSEG,NUMP),NP(NSEG,12)                HYST0570
      DIMENSION FACTOR(IDOF),EEASAT(12,12),TRJT(12,12),STF(JSTOR)               HYST0580
      DIMENSION DUMMY(IDOF,IDOF),SREP(NSEG,NUMP),STR(NSEG,NUMP)                 HYST0590
      DIMENSION R(12,12),PSK(NSEG,12,12),TEP(NSEG,NUMP)                         HYST0600
      DIMENSION ECCX(2),ECCY(2),A0(NUMP),B(NUMP),FF1(IDOF)                      HYST0610
      DIMENSION DEP(NSEG,NUMP),EOP(NSEG,NUMP),STRP(NSEG,NUMP)                   HYST0620
C                                                                               HYST0630
C DEFINE PARAMETERS                                                             HYST0640
      FLP=0.                                                                    HYST0650
      EY=YS/EM                                                                  HYST0660
C                                                                               HYST0670
C -- INITIALIZE THE MEMBER'S ASAT MATRIX --                                     HYST0680
C                                                                               HYST0690
      DO 17 I=1,IDOF                                                            HYST0700
      DO 17 J=1,IDOF                                                            HYST0710
        ASAT(I,J)=0.                                                            HYST0720
   17 CONTINUE                                                                  HYST0730
C                                                                               HYST0740
      GO TO (740, 740, 200),IOPT                                                HYST0750
  740 CONTINUE                                                                  HYST0760
C                                                                               HYST0770
C DEFINE LOCAL BUCKLING FLAG                                                    HYST0780
C                                                                               HYST0790
      DO 666 IK=1,NUMP                                                          HYST0800
  666   FJL(IK)=0                                                               HYST0810
C                                                                               HYST0820
C DEFINE SEGMENT GLOBAL AND LOCAL D O.F. RELATION                               HYST0830
C                                                                               HYST0840
      L1=1                                                                      HYST0850
      DO 300 I=1,12                                                             HYST0860
       IF (I GT.6) L1=NSEG                                                      HYST0870
300    NP(L1,I)=IDOF-12+I                                                       HYST0880
C                                                                               HYST0890
      DO 304 I=1,NSEG-1                                                         HYST0900
        DO 304 J=7,12                                                           HYST0910
          K=I*6-6                                                               HYST0920
          NP(I,J)=K+(J-6)                                                       HYST0930
  304 CONTINUE                                                                  HYST0940
      DO 306 I=2,NSEG                                                           HYST0950
        DO 306 J=1,6                                                            HYST0960
          K=I*6-6                                                               HYST0970
          NP(I,J)=K+(J-6)                                                       HYST0980
  306 CONTINUE                                                                  HYST0990
C                                                                               HYST1000
C DEFINE FORCE-MOMENT INDEX                                                     HYST1010
C                                                                               HYST1020
C                                                                               HYST1030
      DO 9012 I=1,IDOF                                                          HYST1040
        IFM(I)=0                                                                HYST1050
 9012 CONTINUE                                                                  HYST1060
          IFM(IDOF-12+4)=1                                                      HYST1070
          IFM(IDOF-12+5)=1                                                      HYST1080
          IFM(IDOF-12+6)=1                                                      HYST1090
          IFM(IDOF-12-10)=1                                                     HYST1100
          IFM(IDOF-12+11)=1                                                     HYST1110
          IFM(IDOF-12+12)=1                                                     HYST1120
      IF ( IDOF .EQ. 12 ) GO TO 4535                                            HYST1130
      DO 309 I=1,NSEG-1                                                         HYST1140
          J=I*6                                                                 HYST1150
          IFM(J-2)=1                                                            HYST1160
          IFM(J-1)=1                                                            HYST1170
          IFM(J  )=1                                                            HYST1180
  308 CONTINUE                                                                  HYST1190
 4535 CONTINUE                                                                  HYST1200
C                                                                               HYST1210
C DEFINE SECTION POINTS AND SECTIONAL LOCAL COORDINATES                         HYST1220
C                                                                               HYST1230
      CALL LIB(LIBN,HH,UU,WW,ZZ,INEB,INEH,ST,IREV1,IREV2,IREV3,IREV4,           HYST1240
     &         IECOP,U0,V0,A0,B,ECCX0,ECCY0,NUMP)                               HYST1250
      DO 50 I=1,NSEG                                                            HYST1260
        BETA(I)=0.                                                              HYST1270
        DO 50 J=1,NUMP                                                          HYST1280
          U(I,J)=U0(J)                                                          HYST1290
          V(I,J)=V0(J)                                                          HYST1300
   50 CONTINUE                                                                  HYST1310
      IF( IECOP  EQ. 1 ) THEN                                                   HYST1320
          ECX(1)=ECCX0                                                          HYST1330
          ECX(2)=ECCX0                                                          HYST1340
          ECY(1)=ECCY0                                                          HYST1350
          ECY(2)=ECCY0                                                          HYST1360
      ENDIF                                                                     HYST1370
C                                                                               HYST1380
C READ EACH NODAL POINTS' GLOBAL COORDINATES                                    HYST1390
C                                                                               HYST1400
      DO 203 I=1,NSEG                                                           HYST1410
        DO 203 J=1,2                                                            HYST1420
          DO 203 K=1,3                                                          HYST1430
            GCD(I,J,K)=0.                                                       HYST1440
  203 CONTINUE                                                                  HYST1450
      TEMPP=0                                                                   HYST1460
      DO 307 I=1,NSEG                                                           HYST1470
        IF(NSEG .EQ. 1 ) THEN                                                   HYST1480
              SEL=ELS                                                           HYST1490
              EL(I)=SEL                                                         HYST1500
              GCD(I,1,3)=TEMPP                                                  HYST1510
              GCD(I,2,3)=TEMPP+SEL                                              HYST1520
        ELSE IF( I .EQ. 1 .OR. I .EQ. ((NSEG-2)/2)+1                            HYST1530
     &           .OR. I .EQ. ((NSEG-2)/2)+2 .OR. I .EQ. NSEG                    HYST1540
     &           .AND. NSEG .NE. 1 ) THEN                                       HYST1550
              SEL=SMALL                                                         HYST1560
              EL(I)=SEL                                                         HYST1570
              GCD(I,1,3)=TEMPP                                                  HYST1580
              GCD(I,2,3)=TEMPP+SEL                                              HYST1590
              TEMPP=TEMPP+SEL                                                   HYST1600
        ELSE IF( I .NE. 1 .OR. I .NE. ((NSEG-2)/2)+1                            HYST1610
     &           .OR. I .NE. ((NSEG-2)/2)+2  OR. I NE. NSEG                     HYST1620
     &           .AND. NSEG .NE. 1 ) THEN                                       HYST1630
              SEL=(ELS-4*SMALL)/(NSEG-4)                                        HYST1640
              EL(I)=SEL                                                         HYST1650
              GCD(I,1,3)=TEMPP                                                  HYST1660
              GCD(I,2,3)=TEMPP+SEL                                              HYST1670
              TEMPP=TEMPP+SEL                                                   HYST1680
        ENDIF                                                                   HYST1690
  307 CONTINUE                                                                  HYST1700
C                                                                               HYST1710
C INITIALIZ THE MEMBER GLOBAL DISPLACEMENT AT EACH D.O.F                        HYST1720
C                                                                               HYST1730
      DO 157 I=1,IDOF                                                           HYST1740
       D1(I)=0                                                                  HYST1750
       DD(I)=0                                                                  HYST1760
       D00(I)=0.                                                                HYST1770
       DPCOPY(I)=0                                                              HYST1780
  157 CONTINUE                                                                  HYST1790
C                                                                               HYST1800
C GENERATE THE INITIAL IMPERFECTION DEFLECTION                                  HYST1810
C                                                                               HYST1820
      PI=3 1415926                                                              HYST1830
      DO 115 I=1,NSEG                                                           HYST1840
        D00(NP(I,1))=(RATIX0*ELS)*SIN(PI*GCD(I,1,3)/ELS)                        HYST1850
        D00(NP(I,7))=(RATIX0*ELS)*SIN(PI*GCD(I,2,3)/ELS)                        HYST1860
        D00(NP(I,2))=(RATIY0*ELS)*SIN(PI*GCD(I,1,3)/ELS)                        HYST1870
        D00(NP(I,8))=(RATIY0*ELS)*SIN(PI*GCD(I,2,3)/ELS)                        HYST1880
  115 CONTINUE                                                                  HYST1890
C                                                                               HYST1900
C GENERATE THE SEGMENT LEHGTH (INCLUDING THE INITIAL IMPERFECTION               HYST1910
C                                                                               HYST1920
      DO 912 I=1,NSEG                                                           HYST1930
        EL(I)=SQRT(EL(I)**2+(D00(NP(I,9))-D00(NP(I,2)))**2                      HYST1940
     &       -(D00(NP(I,7))-D00(NP(I,1)))**2 )                                  HYST1950
        TL(I)=EL(I)                                                             HYST1960
  912 CONTINUE                                                                  HYST1970
C                                                                               HYST1980
C INITIALIZE THE LOCAL DISPLACEMENT AND LOCAL INTERNAL FORCES                   HYST1990
C AND GLOBAL INTERNAL FORCE                                                     HYST2000
C                                                                               HYST2010
      DO 329 I=1,NSEG                                                           HYST2020
```

```
      DO 329 J=1,12                                                     HYST2030
         XLOC(I,J)=0.                                                   HYST2040
         PLOC(I,J)=0.                                                   HYST2050
         XGLO(I,J)=0.                                                   HYST2060
         PGLOB(I,J)=0.                                                  HYST2070
  329 CONTINUE                                                          HYST2080
C                                                                       HYST2090
C INITIALIZE THE SECTIONAL STRAIN AND BETA VALUES                       HYST2100
C                                                                       HYST2110
      DO 325 I=1,NSEG                                                   HYST2120
         BETA(I)=0                                                      HYST2130
         DO 325 J=1,NUMP                                                HYST2140
            E0(I,J)=0                                                   HYST2150
            E1(I,J)=EY                                                  HYST2160
            E2(I,J)=-EY                                                 HYST2170
            TE(J)=EM                                                    HYST2180
  325 CONTINUE                                                          HYST2190
C                                                                       HYST2200
      DO 112 I=1,NSEG                                                   HYST2210
C                                                                       HYST2220
C FINE THE ROTATION MATRIX DUE TO INITIAL IMPERFECTION EFFECT           HYST2230
C                                                                       HYST2240
       PR(I,1)=1                                                        HYST2250
       PR(I,2)=0.                                                       HYST2260
       PR(I,3)=0.                                                       HYST2270
       PR(I,4)=0.                                                       HYST2280
       PR(I,5)=1                                                        HYST2290
       PR(I,6)=0.                                                       HYST2300
       PR(I,7)=0.                                                       HYST2310
       PR(I,8)=0.                                                       HYST2320
       PR(I,9)=1.                                                       HYST2330
C                                                                       HYST2340
       K00=0                                                            HYST2350
110    XA=D1(NP(I,1))*GCD(I,1,1)+D00(NP(I,1))                           HYST2360
       YA=D1(NP(I,2))*GCD(I,1,2)+D00(NP(I,2))                           HYST2370
       ZA=D1(NP(I,3))*GCD(I,1,3)+D00(NP(I,3))                           HYST2380
       ROZA=D1(NP(I,6))+D00(NP(I,6))                                    HYST2390
       XB=D1(NP(I,7))*GCD(I,2,1)+D00(NP(I,7))                           HYST2400
       YB=D1(NP(I,8))*GCD(I,2,2)+D00(NP(I,8))                           HYST2410
       ZB=D1(NP(I,9))*GCD(I,2,1)+D00(NP(I,9))                           HYST2420
       ROZB=D1(NP(I,12))+D00(NP(I,12))                                  HYST2430
       DXA=DD(NP(I,1))                                                  HYST2440
       DYA=DD(NP(I,2))                                                  HYST2450
       DZA=DD(NP(I,3))                                                  HYST2460
       DROXA=DD(NP(I,4))                                                HYST2470
       DROYA=DD(NP(I,5))                                                HYST2480
       DROZA=DD(NP(I,6))                                                HYST2490
       DXB=DD(NP(I,7))                                                  HYST2500
       DYB=DD(NP(I,8))                                                  HYST2510
       DZB=DD(NP(I,9))                                                  HYST2520
       DROXB=DD(NP(I,10))                                               HYST2530
       DROYB=DD(NP(I,11))                                               HYST2540
       DROZB=DD(NP(I,12))                                               HYST2550
       IF (K00 EQ.1) GOTO 135                                           HYST2560
C                                                                       HYST2570
C VARIABLES FOR THE ROTATIONAL MATRIX DUE TO INITIAL IMPERFECTION       HYST2580
C                                                                       HYST2590
      VXC=PR(I,4)                                                       HYST2600
      VYC=PR(I,5)                                                       HYST2610
      VZC=PR(I,6)                                                       HYST2620
      WXC=PR(I,7)                                                       HYST2630
      WYC=PR(I,8)                                                       HYST2640
      WZC=PR(I,9)                                                       HYST2650
C                                                                       HYST2660
C CALCULATE ROTATIONAL MATRIX OF SEGMENT                                HYST2670
C                                                                       HYST2680
      CALL ROMA(XA,YA,ZA,ROZA,XB,YB,ZB,ROZB,VX0,VY0,VZ0,WX0,            HYST2690
     *          WY0,WZ0,I,R,PR,BETA,TL,NSEG)                            HYST2700
C                                                                       HYST2710
       L0=0                                                             HYST2720
       DO 110 J=1,3                                                     HYST2730
       DO 110 K=1,3                                                     HYST2740
       L0=L0+1                                                          HYST2750
110    TIR(I,L0)=R(J,K)                                                 HYST2760
C                                                                       HYST2770
C GENERATE THE INITIAL ROTATIONAL MATRIX OR PREVIOUS ROTATIONAL         HYST2780
C MATRIX                                                                HYST2790
C                                                                       HYST2800
C                                                                       HYST2810
C FIND SEGMENTS' PRINCIPAL AXIS AND MATERIAL PROPERTIES                 HYST2820
C ( SUBROUTINE STRATE AND JUDEL ARE CALLED BY PRINC )                   HYST2830
C                                                                       HYST2840
      CALL PRINC(0.,0.,0.,I,ST,TOTA,LIBN,EM,EY,G,YS,IOPT,NUMP,NSEG,     HYST2850
     &           EA,EIU,EIV,GKT,IREV1,IREV2,IREV3,IREV4,UC0,VC0,IMATER, HYST2860
     &           A0,B,BETA,E0,E1,E2,U,V,U0,V0,TE,E,DE,INEB,INEH,PJL,    HYST2870
     &           RATIO3,IR,STR,DEP,EOP,STRP,TEP,EREP,SREP,PLP)          HYST2880
C                                                                       HYST2890
C CALCULATE ECCENTRICITIES OF TWO END-SEGMENTS                          HYST2900
C                                                                       HYST2910
      IF (I.EQ 1.OR.I.EQ.NSEG.AND.IECOP.EQ.1) THEN                      HYST2920
         CALL ECCENT(NSEG,UC0,VC0,I,0.,0.,ECX,ECY,ECCX,ECCY)            HYST2930
      ENDIF                                                             HYST2940
C                                                                       HYST2950
C CALCULATE INITIAL ROTATION MATRIX OF SEGMENT                          HYST2960
C                                                                       HYST2970
       K00=1                                                            HYST2980
       GOTO 130                                                         HYST2990
C                                                                       HYST3000
C VARIABLES FOR THE INITIAL ROTATIONAL MATRIX                           HYST3010
C                                                                       HYST3020
135   VX0=TIR(I,4)                                                      HYST3030
      VY0=TIR(I,5)                                                      HYST3040
      VZ0=TIR(I,6)                                                      HYST3050
      WX0=TIR(I,7)                                                      HYST3060
      WY0=TIR(I,8)                                                      HYST3070
      WZ0=TIR(I,9)                                                      HYST3080
C                                                                       HYST3090
C CALCULATE ROTATIONAL MATRIX OF SEGMENT                                HYST3100
C                                                                       HYST3110
      CALL ROMA(XA,YA,ZA,ROZA,XB,YB,ZB,ROZB,VX0,VY0,VZ0,WX0,            HYST3120
     *          WY0,WZ0,I,R,PR,BETA,TL,NSEG)                            HYST3130
      DEC=0.                                                            HYST3140
      DCU=0.                                                            HYST3150
      DCV=0.                                                            HYST3160
C                                                                       HYST3170
C FORMULATE THE SEGMENT'S LOACL STIFFNESS MATRIX                        HYST3180
C                                                                       HYST3190
       CALL ELESTI(NSEG,I,ISTIP,EA,EIU,EIV,GKT,EL,PLOC,PSX,SX)          HYST3200
C                                                                       HYST3210
C TRANSFER SEGMENT'S LOCAL STIFFNESS TO GLOBAL STIFFNESS                HYST3220
C                                                                       HYST3230
      CALL MULTM(0,12,DUMMY,SX,R)                                       HYST3240
      CALL MULTM(1,12,SX,R,DUMMY)                                       HYST3250
C                                                                       HYST3260
C ASSEMBLING SEGMENT GLOBAL STIFFNESS TO MEMBER                         HYST3270
C CENTROID STIFFNESS MATRIX                                             HYST3280
C                                                                       HYST3290
      DO 85 J=1,12                                                      HYST3300
      DO 85 JJ=1,12                                                     HYST3310
        ASAT(NP(I,J),NP(I,JJ))=ASAT(NP(I,J),NP(I,JJ))+SK(J,JJ)         HYST3320
   85 CONTINUE                                                          HYST3330
C                                                                       HYST3340
  112 CONTINUE                                                          HYST3350
      GOTO 555                                                          HYST3360
C                                                                       HYST3370
C------------------- ( IOPT =2 ) ---------------------------------------HYST3380
  200 CONTINUE                                                          HYST3390
      IF( ISTART .LT. 5 ) THEN                                          HYST3400
         ISTART=ISTART+1                                                HYST3410
      ELSE                                                              HYST3420
         ISTART=5                                                       HYST3430
      ENDIF                                                             HYST3440
C                                                                       HYST3450
      DO 426 I=1,IDOF                                                   HYST3460
  426   FACTOR(I)=0                                                     HYST3470
C                                                                       HYST3480
C DISPLACEMENT INCREMENT TRANSFORMATION DUE TO ECCENTRIC LOADS          HYST3490
C                                                                       HYST3500
      IF( IECOP .EQ. 1 ) THEN                                           HYST3510
        DO 913 I=1,12                                                   HYST3520
          DP(I)=DD(I)                                                   HYST3530
  913   CONTINUE                                                        HYST3540
        DO 525 II=1,12                                                  HYST3550
          DD(II)=0.                                                     HYST3560
          DO 525 KK=1,12                                                HYST3570
            DD(II)=DD(II)+TPJT(II,KK)*DP(KK)                            HYST3580
  525   CONTINUE                                                        HYST3590
      ENDIF                                                             HYST3600
C                                                                       HYST3610
C CONVERT MEMBER END DISPLACEMENT INCREMENT TO THE CORRESPONDING        HYST3620
C MEMBER'S DOF DIRECTION                                                HYST3630
C                                                                       HYST3640
      DO 837 I=1,12                                                     HYST3650
        DD(IDOF-12+I)=DD(I)                                             HYST3660
  837 CONTINUE                                                          HYST3670
C                                                                       HYST3680
C BACK SUBSTITUTE TO GET MEMBER'S INTERNAL DOF DISPLACEMENT INCREMENT   HYST3690
C                                                                       HYST3700
      L1=IDOF-12+1                                                      HYST3710
      L2=IDOF                                                           HYST3720
                                                                        HYST3730
      DO 838 I=L1,L2                                                    HYST3740
  838   FACTOR(I)=DD(I)                                                 HYST3750
      IMD=MDIAG(IDOF+1)-1                                               HYST3760
C                                                                       HYST3770
      CALL GAUSS2(4,IDOF,L1,L2,MDIAG,STF,DPCOPY,FACTOR,JSTOR)           HYST3780
      DO 901 I=1,L1-1                                                   HYST3790
        DP(I)=0                                                         HYST3800
  901   DPCOPY(I)=0                                                     HYST3810
      DO 902 I=L1,L2                                                    HYST3820
  902   DP(I)=DPCOPY(I)                                                 HYST3830
      DO 848 I=IDOF-12+1,IDOF                                           HYST3840
  848   FACTOR(I)=DD(I)                                                 HYST3850
C                                                                       HYST3860
C                                                                       HYST3870
C     ----------------------------                                      HYST3880
C     --- BACK SUBSTITUTE BEGIN ---                                     HYST3890
C     ----------------------------                                      HYST3900
      IF ( IDOF .NE  12 ) THEN                                          HYST3910
       L2=IDOF-12                                                       HYST3920
       CALL GAUSS2(5,IDOF,1,L2,MDIAG,STF,DPCOPY,FACTOR,JSTOR)           HYST3930
      ENDIF                                                             HYST3940
C                                                                       HYST3950
C                                                                       HYST3960
C ADD GLOBAL DISPLACEMENT RATE TO TOTAL DISPLACEMENT                    HYST3970
C                                                                       HYST3980
      DO 903 I=1,IDOF                                                   HYST3990
        DD(I)=FACTOR(I)                                                 HYST4000
  903   D1(I)=D1(I)+DD(I)                                               HYST4010
C                                                                       HYST4020
      DO 219 I=1,NSEG                                                   HYST4030
      DO 219 J=1,12                                                     HYST4040
        XGLO(I,J)=DD(NP(I,J))                                           HYST4050
  219 CONTINUE                                                          HYST4060
C                                                                       HYST4070
C FIND EACH SEGMENT INTERNAL FORCE IN GLOBAL DIRECTIONS                 HYST4080
C                                                                       HYST4090
C                                                                       HYST4100
C ADJUST THE INITIAL IMPERFECTION DEFLECTION FOR NEW CYCLE, ISPE-1      HYST4110
C                                                                       HYST4120
      IF( ISPE .EQ. 1 ) THEN                                            HYST4130
       PI=3.1415926                                                     HYST4140
       DO 210 I=1,IDOF                                                  HYST4150
         D00(I)=0.                                                      HYST4160
  210  CONTINUE                                                         HYST4170
       ELS=0.                                                           HYST4180
       DO 211 I=1,NSEG                                                  HYST4190
        SZA=D1(NP(I,3))*GCD(I,1,3)                                      HYST4200
        SZB=D1(NP(I,9))*GCD(I,2,3)                                      HYST4210
        EL(I)=SZB-SZA                                                   HYST4220
  211   ELS=ELS+EL(I)                                                   HYST4230
C                                                                       HYST4240
C     ------------------------------------------------------------      HYST4250
C     ASSUME INITIAL IMPERFECTION RATIO BASED ON MEMBER MIDDLE POINT    HYST4260
C     DISPLACEMENT                                                      HYST4270
C     ------------------------------------------------------------      HYST4280
       IDUM=((NSEG/2)-1)*6 + 1                                          HYST4290
       RATIX0=D1(IDUM)/ELS                                              HYST4300
       RATIY0=D1(IDUM+1)/ELS                                            HYST4310
C                                                                       HYST4320
       DO 215 I=1,NSEG                                                  HYST4330
        D00(NP(I,1))=(RATIX0*ELS)*SIN(PI*                               HYST4340
     &               (D1(NP(I,3))*GCD(I,1,3)-D1(NP(1,3)))/ELS)          HYST4350
        D00(NP(I,7))=(RATIX0*ELS)*SIN(PI*                               HYST4360
     &               (D1(NP(I,9))*GCD(I,2,3)-D1(NP(1,3)))/ELS)          HYST4370
        D00(NP(I,2))=(RATIY0*ELS)*SIN(PI*                               HYST4380
     &               (D1(NP(I,3))*GCD(I,1,3)-D1(NP(1,3)))/ELS)          HYST4390
        D00(NP(I,8))=(RATIY0*ELS)*SIN(PI*                               HYST4400
     &               (D1(NP(I,9))*GCD(I,2,3)-D1(NP(1,3)))/ELS)          HYST4410
  215  CONTINUE                                                         HYST4420
C                                                                       HYST4430
C GENERATE THE SEGMENT LENGTH (INCLUDING THE INITIAL IMPERFECTION       HYST4440
C                                                                       HYST4450
       DO 922 I=1,NSEG                                                  HYST4460
        EL(I)=SQRT(EL(I)**2+(D00(NP(I,8))-D00(NP(I,2)))**2              HYST4470
     &           +(D00(NP(I,7))-D00(NP(I,1)))**2 )                      HYST4480
        TL(I)=EL(I)                                                     HYST4490
  922  CONTINUE                                                         HYST4500
       WRITE(108,915)                                                   HYST4510
       WRITE(108,916)                                                   HYST4520
       WRITE(108,917) (I,EL(I),I=1,NSEG)                                HYST4530
  915  FORMAT(//1X,'THE SEGMENT LENGTH:')                               HYST4540
  916  FORMAT(1X,'SEGMENT    LENGTH')                                   HYST4550
  917  FORMAT(I5,F15.6)                                                 HYST4560
       ENDIF                                                            HYST4570
C                                                                       HYST4580
C ----------------------------------------------------                  HYST4590
C     FIND EACH SEGMENT INTERNAL FORCE IN GLOBAL DIRECTIONS             HYST4600
C ----------------------------------------------------                  HYST4610
      DO 90 I=1,NSEG                                                    HYST4620
        CALL CLDAF(E,PR,I,NSEG,XGLO,PSX,SX,PGLOB)                       HYST4630
C                                                                       HYST4640
C FIND SEGMENT DEFORMATION AND INCREMENTS IN MEMBER COORDINATES         HYST4650
C DIRECTIONS                                                            HYST4660
C                                                                       HYST4670
        IF (ISPE.EQ.0.OR.ISPE.EQ.1) THEN                                HYST4680
         IF ( ISPE .EQ. 0 ) GOTO 330                                    HYST4690
           DO 335 J=1,12                                                HYST4700
             IF (J EQ 3.OR J.EQ.6.OR J.EQ.9.OR J EQ.12) GOTO 335        HYST4710
             D1(NP(I,J))=0.                                             HYST4720
  335      CONTINUE                                                     HYST4730
  330      XA=D1(NP(I,1))+GCD(I,1,1)+D00(NP(I,1))                       HYST4740
           YA=D1(NP(I,2))+GCD(I,1,2)+D00(NP(I,2))                       HYST4750
           ZA=D1(NP(I,3))+GCD(I,1,3)+D00(NP(I,3))                       HYST4760
           ROZA=D1(NP(I,6))+D00(NP(I,6))                                HYST4770
           XB=D1(NP(I,7))+GCD(I,2,1)+D00(NP(I,7))                       HYST4780
           YB=D1(NP(I,8))+GCD(I,2,2)+D00(NP(I,8))                       HYST4790
           ZB=D1(NP(I,9))+GCD(I,2,3)+D00(NP(I,9))                       HYST4800
           ROZB=D1(NP(I,12))+D00(NP(I,12))                              HYST4810
           DXA=DD(NP(I,1))                                              HYST4820
           DYA=DD(NP(I,2))                                              HYST4830
           DZA=DD(NP(I,3))                                              HYST4840
           DROXA=DD(NP(I,4))                                            HYST4850
           DROYA=DD(NP(I,5))                                            HYST4860
           DROZA=DD(NP(I,6))                                            HYST4870
           DXB=DD(NP(I,7))                                              HYST4880
           DYB=DD(NP(I,8))                                              HYST4890
           DZB=DD(NP(I,9))                                              HYST4900
           DROXB=DD(NP(I,10))                                           HYST4910
           DROYB=DD(NP(I,11))                                           HYST4920
           DROZB=DD(NP(I,12))                                           HYST4930
        ENDIF                                                           HYST4940
```

```
C                                                                       HYST4950
C VARIABLES FOR THE INITIAL ROTATIONAL MATRIX                           HYST4960
C                                                                       HYST4970
      VX0=TIR(I,4)                                                      HYST4980
      VY0=TIR(I,5)                                                      HYST4990
      VZ0=TIR(I,6)                                                      HYST5000
      WX0=TIR(I,7)                                                      HYST5010
      WY0=TIR(I,8)                                                      HYST5020
      WZ0=TIR(I,9)                                                      HYST5030
C                                                                       HYST5040
C CALCULATE SEGMENT CURVATURE RATE IN PRINCIPAL AXES DIRECTIONS         HYST5050
C                                                                       HYST5060
      CALL CURVA(NSEG,I,DEC,DCU,DCV,E,EL,PR,DXA,DYA,DZA,DROXA,DROYA,    HYST5070
     &           DROZA,DXB,DYB,DZB,DROXB,DROYB,DROZB )                  HYST5080
C                                                                       HYST5090
C CALCULATE EACH SEGMEN'S PRINCIPAL AXIS AND MATERIAL PROPERTIES        HYST5100
C                                                                       HYST5110
      CALL PRINC(DEC,DCU,DCV,I,ST,TOTA,LIBN,EM,EY,G,YS,IOPT,NUMP,NSEG,  HYST5120
     &           EA,EIU,EIV,GKT,IREV1,IREV2,IREV3,IREV4,UC0,VC0,IMATER, HYST5130
     &           A0,B,BETA,E0,E1,E2,U,V,U0,V0,TE,E,DE,INEB,INEH,PJL,    HYST5140
     &           RATIO3,IR,STR,DEP,E0P,STRP,TEP,EREP,SREP,FLP)          HYST5150
C                                                                       HYST5160
C CALCULATE ECCENTRICITIES OF TWO END-SEGMENTS                          HYST5170
C                                                                       HYST5180
      IF (I.EQ.1.OR.I.EQ.NSEG.AND.IECOP.EQ.1) THEN                      HYST5190
        CALL ECCENT(NSEG,UC0,VC0,I,ROZA,ROZB,ECX,ECY,ECCX,ECCY)         HYST5200
      ENDIF                                                             HYST5210
C                                                                       HYST5220
C     CALCULATE SEGMENT CURRENT ROTATIONAL MATRIX                       HYST5230
C                                                                       HYST5240
      CALL ROMA(XA,YA,ZA,ROZA,XB,YB,ZB,ROZB,VX0,VY0,VZ0,WX0,            HYST5250
     *          WY0,WZ0,I,R,PR,BETA,TL,NSEG)                          HYST5260
C                                                                       HYST5270
C CALCULATE EACH SEGMENT LOCAL FORCE                                    HYST5280
C                                                                       HYST5290
      DO 710 II=1,12                                                    HYST5300
        PLOC(I,II)=0.                                                   HYST5310
      DO 710 KK=1,12                                                    HYST5320
  710   PLOC(I,II)=PLOC(I,II)+R(II,KK)*PGLOB(I,KK)                     HYST5330
C                                                                       HYST5340
C CALCULATE ELEMENT STIFFNESS MATRIX                                    HYST5350
C                                                                       HYST5360
       CALL ELESTI(NSEG,I,ISTIP,EA,EIU,EIV,GKT,EL,PLOC,PSK,SK )         HYST5370
C                                                                       HYST5380
C CALCULATE SYSTEM STIFFNESS MATRIX                                     HYST5390
C                                                                       HYST5400
      CALL MULTM(0,12,DUMMY,SK,R)                                       HYST5410
      CALL MULTM(1,12,SK,R,DUMMY)                                       HYST5420
C                                                                       HYST5430
C ASSEMBLING SEGMENT GLOBAL STIFFNESS TO MEMBER                         HYST5440
C CENTROID STIFFNESS MATRIX                                             HYST5450
C                                                                       HYST5460
      DO 805 J=1,12                                                     HYST5470
      DO 805 JJ=1,12                                                    HYST5480
        ASAT(NP(I,J),NP(I,JJ))=ASAT(NP(I,J),NP(I,JJ))+SK(J,JJ)         HYST5490
  805 CONTINUE                                                          HYST5500
C                                                                       HYST5510
   90 CONTINUE                                                          HYST5520
C                                                                       HYST5530
C CALCULATE MEMBER'S LOAD INCREMENTS BASED ON :K|*DD                    HYST5540
C                                                                       HYST5550
      DO 5021 II=1,IDOF                                                 HYST5560
        DP(II)=0.                                                       HYST5570
        FF1(II)=0.                                                      HYST5580
      DO 5021 K=1,IDOF                                                  HYST5590
        DP(II)=DP(II)+SEASAT(II,K)*DD(K)                                HYST5600
 5021 CONTINUE                                                          HYST5610
C                                                                       HYST5620
C CALCULATE MEMBER'S TOTAL FORCE IN THE GLOBAL DIRECTION                HYST5630
C                                                                       HYST5640
      DO 804 J=1,NSEG                                                   HYST5650
        DO 804 K=1,12                                                   HYST5660
          FF1(NP(J,K))=FF1(NP(J,K))+PGLOB(J,K)                          HYST5670
  804 CONTINUE                                                          HYST5680
C                                                                       HYST5690
C     ------------------------------------------------                  HYST5700
C     CALCULATE STABILITY MEMBER'S STIFFNESS PARAMETER AND              HYST5710
C     DISPLACEMENT NORM                                                 HYST5720
C     ------------------------------------------------                  HYST5730
C                                                                       HYST5740
      IF ( IAUTO  EQ. 1 ) THEN                                          HYST5750
C                                                                       HYST5760
        DO 977 MM=1,IDOF                                                HYST5770
          DPCOPY(MM)=D1(MM)                                             HYST5780
        DO 977 NN=1,IDOF                                                HYST5790
  977     DUMMY(MM,NN)=SEASAT(MM,NN)                                    HYST5800
C                                                                       HYST5810
        IF ( ISTART .EQ. 2 ) THEN                                       HYST5820
         CALL SNORM(1,FF1,DD,DP,DUMMY,DELTP2,S2,IDOF,IPM)               HYST5830
        ENDIF                                                           HYST5840
C                                                                       HYST5850
        IF ( ISTART .GE. 2 ) THEN                                       HYST5860
          CALL SNORM(1,FF1,DD,DP,DUMMY,DELTPJ,SJ,IDOF,IPM)              HYST5870
          CALL SNORM(2,DPCOPY,DD,DD,DUMMY,EEE,DMY,IDOF,IPM)             HYST5880
          IF (SJ*DELTP2.NE.0.) SP=((DELTPJ/DELTP2)**2)*(S2/SJ)          HYST5890
          RNEEE=ABS(EEE/QRNEE)                                          HYST5900
        ENDIF                                                           HYST5910
      ENDIF                                                             HYST5920
C                                                                       HYST5930
C MEMORIZE THE STABILITY ELEMENT STIFFNESS                              HYST5940
C                                                                       HYST5950
555   DO 987 I=1,IDOF                                                  HYST5960
        DO 987 J=1,IDOF                                                 HYST5970
          SEASAT(I,J)=ASAT(I,J)                                         HYST5980
  987 CONTINUE                                                          HYST5990
C                                                                       HYST6000
C CONVERT ASAT MATRIX TO HALF STORAGE AND BANDED MODE                   HYST6010
C                                                                       HYST6020
      CALL SKY(ASAT,STF,JSTOR,IDOF,MDIAG,IMD)                           HYST6030
C                                                                       HYST6040
C FIND CONDENSED STIFFNESS MATRIX CORRESPONDING TO TWO-END 12 DOF       HYST6050
C                                                                       HYST6060
      IF ( IDOF .NE. 12 ) THEN                                          HYST6070
       L2=IDOF-12                                                       HYST6080
       CALL GAUSS2(1,IDOF,1,L2,MDIAG,STF,DPCOPY,FACTOR,JSTOR)           HYST6090
      ENDIF                                                             HYST6100
C                                                                       HYST6110
C CONDENSE ASAT MATRIX TO GET CONDENSED MATRIX CORRESPONDING TO         HYST6120
C TWO ENDS TWELVE DOF DIRECTIONS                                        HYST6130
C                                                                       HYST6140
      DO 205 J=L1,IDOF,-1                                               HYST6150
        DO 205 I=J,L1,-1                                                HYST6160
          IJ=MDIAG(J)-J+I                                               HYST6170
          IF(IJ  LT  MDIAG(J+1)  .OR. IJ .EQ. MDIAG(J) ) THEN           HYST6180
              EEASAT(I-L1+1,J-L1+1)=STF(IJ)                             HYST6190
              EEASAT(J-L1+1,I-L1+1)=EEASAT(I-L1+1,J-L1+1)               HYST6200
          ENDIF                                                         HYST6210
  205   CONTINUE                                                        HYST6220
C                                                                       HYST6230
      IF( IECOP  EQ. 1 ) CALL TRANK(EEASAT,ECCX,ECCY,TPJT,SK)           HYST6240
C                                                                       HYST6250
      RETURN                                                            HYST6260
      END                                                               HYST6270
C                                                                       SNOR0010
C ------------------------------ ------------------------------        SNOR0020
C                                                                       SNOR0030
      SUBROUTINE SNORM(IOPT,BR,DSR,DBR,TK,DELTP,DEN,NOP,IPM)            SNOR0040
C-----------------------------------------------------------------      SNOR0050
C CALCULATE NORM OF LOAD INCREMENT AND NORM OF DISPLACEMENT INCREMENT   SNOR0060
C                                                                       SNOR0070
C IOPT : 1 - CALCULATE DELTP FOR NORM OF LOAD AND DEN                   SNOR0080
C IOPT   2 - CALCULATE DELTP FOR NORM OF DISPLACEMENT                   SNOR0090
C                                                                       SNOR0100
C SR    : DISPLACEMENT VECTOR                                           SNOR0110
C BR    : LOAD VECTOR                                                   SNOR0120
C DSR   : DISPLACEMENT INCREMENT VECTOR                                 SNOR0130
C DBR   : LOAD VECTOR  INCREMENT VECTOR                                 SNOR0140
C TK    : STIFFNESS MATRIX                                              SNOR0150
C DELTP: NORM OF LOAD OR DISPLACEMENT                                   SNOR0160
C DEN   : DEN = TRANSPOSE OF :DSR:*:TK:*:DSR:                          SNOR0170
CIPM  : TRANVERSE AND ROTATION DOF IDENTIFICATION INDEX                 SNOR0180
C       0 FOR TRANVERSE                                                 SNOR0190
C       1 FOR ROTATION                                                  SNOR0200
C                                                                       SNOR0210
C NOTE: SR(MM), BR(MM), DSR(MM), DBR(MM), TK(MM,MM)                     SNOR0220
C       IPM(NN)                                                         SNOR0230
C-----------------------------------------------------------------      SNOR0240
C                                                                       SNOR0250
      DIMENSION DSR(NOP),DBR(NOP),BR(NOP),TK(NOP,NOP),IPM(NOP)          SNOR0260
C                                                                       SNOR0270
C SKIM STIFFNESS TERMS                                                  SNOR0280
C                                                                       SNOR0290
      DO 80 I=1,NOP                                                     SNOR0300
      DO 80 J=1,NOP                                                     SNOR0310
        IF( ABS(TK(I,J)) .LT. 1.E-7 ) TK(I,J)=0                         SNOR0320
   80 CONTINUE                                                          SNOR0330
C                                                                       SNOR0340
C SEARCH MAXIMUM FORCE OR DISPLACEMENT INCREMENT                        SNOR0350
C                                                                       SNOR0360
      FMAX=0                                                            SNOR0370
      TMMAX=0.                                                          SNOR0380
      DO 25 I=1,NOP                                                     SNOR0390
        IF(IPM(I) .EQ. 0 ) THEN                                         SNOR0400
          IF(ABS(BR(I))  GT. ABS(FMAX) ) THEN                           SNOR0410
                FMAX=BR(I)                                              SNOR0420
          ENDIF                                                         SNOR0430
        ELSE IF(IPM(I) .NE. 0 ) THEN                                    SNOR0440
          IF(ABS(BR(I)) .GT. ABS(TMMAX) ) THEN                          SNOR0450
                TMMAX=BR(I)                                             SNOR0460
          ENDIF                                                         SNOR0470
        ENDIF                                                           SNOR0480
   25 CONTINUE                                                          SNOR0490
C                                                                       SNOR0500
C FIND NORM OF DBR                                                      SNOR0510
C                                                                       SNOR0520
      DELT=0.                                                           SNOR0530
      DO 32 I=1,NOP                                                     SNOR0540
       IF(IPM(I) .EQ. 0 ) THEN                                          SNOR0550
         IF(FMAX .EQ. 0. ) GO TO 32                                     SNOR0560
         DELT=DELT+(ABS(DBR(I)/FMAX))**2                                SNOR0570
       ELSE IF(IPM(I) .NE. 0 ) THEN                                     SNOR0580
         IF(TMMAX .EQ. 0. ) GO TO 32                                    SNOR0590
         DELT=DELT+(ABS(DBR(I)/TMMAX))**2                               SNOR0600
       ENDIF                                                            SNOR0610
   32 CONTINUE                                                          SNOR0620
      DELTP=SQRT(DELT/NOP)                                              SNOR0630
      IF (IOPT.EQ.2) GOTO 200                                           SNOR0640
C                                                                       SNOR0650
C :DSRT:**(T)*:TK:=:DSRT:                                               SNOR0660
C                                                                       SNOR0670
      DEN=0.                                                            SNOR0680
      DO 68 J=1,NOP                                                     SNOR0690
       DO 68 K=1,NOP                                                    SNOR0700
         DEN=DEN+DSR(K)*TK(K,J)*DSR(K)                                  SNOR0710
   68 CONTINUE                                                          SNOR0720
C                                                                       SNOR0730
200   RETURN                                                            SNOR0740
      END                                                               SNOR0750
C                                                                       GAUS0010
C                                                                       GAUS0020
C-----------------------------------------------------------------      GAUS0030
C  MATRIX SOLUTION OF EQUATION AX=P, WHERE THE UPPER TRIANGULAR MATRIX  GAUS0040
C    OF THE SYMMETRIC MATRIX A IS STORED BY SKYLINES. P IS STORED AS    GAUS0050
C    A FULL MATRIX.  THE MATRIX P IS REPLACED BY X AFTER BACK           GAUS0060
C    SUBSTITUTION IS COMPLEATE.  BOTH A AND P ARE ALTERED....           GAUS0070
C  GUYAN OPTION: IF GUYAN IS .TRUE. MASS MATRIX IS ALSO CONDENSED       GAUS0080
C-----------------------------------------------------------------      GAUS0090
      SUBROUTINE GAUSS2(IOPT,N,L1,L2,MD,A,P,FACTOR,ISTOR)               GAUS0100
      DIMENSION A(ISTOR),FACTOR(N),P(N),MD(N+1)                         GAUS0110
C                                                                       GAUS0120
      IJI(I,J)=MD(J)+J-I                                                GAUS0130
C.....................................................................  GAUS0140
C   MATRIX STORAGE SCHEME:                                              GAUS0150
C  THE MATRIX A IS BANDED AND SYMMETRIC. THUS ONLY THE SKYLINE ABOVE    GAUS0160
C   AND THE MAIN DIAGONAL NEEDS TO BE STORED.                           GAUS0170
C  THE MATRIX A IS STORED IN A LINEAR ARRAY BY COLUMNS.  THE VALUE ON   GAUS0180
C   THE MAIN DIAGONAL IS STORED IN THE LINEAR ARRAY FIRST.              GAUS0190
C   ADDRESSES OF THE LINEAR ARRAY ELEMENTS CONTAINING THE MAIN DIAGONAL GAUS0200
C   TERMS ARE STORED IN MD.                                             GAUS0210
C   EXAMPLE: LET B BE THE 5X5 FULL SYMETRIC MATRIX BELOW..              GAUS0220
C       :  1  :  3  :  :  : 12 :                                       GAUS0230
C       :     :  2  :  5  :  : 11 :  NOTE: B(1,3),B(1,4) AND B(2,4) ARE ZERO. GAUS0240
C   B=  :     :     :  4  :  7  : 10 :                                  GAUS0250
C       :     : SYMM.     :  6  :  9 :  MD= 1, 2, 4, 6, 8, 13          GAUS0260
C       :     :     :     :     :  8 :                                  GAUS0270
C   THUS, B(3,3)=A( MD(3) )= A(4)                                       GAUS0280
C         B(I,J)=A( MD(J)+J-I )  IF J>I AND (MD(J)+J-I)<MD(J+1)         GAUS0290
C               =0               IF J>I AND (MD(J)+J-I)>MD(J+1)       GAUS0300
C  MATRIX P IS NOT SYMMETRIC, AND IS STORED AS A FULL MATRIX            GAUS0310
C.....................................................................  GAUS0320
C   VARIABLE TABLE:                                                     GAUS0330
C   IOPT  = OPTION NUMBER                                               GAUS0340
C   N     = NUMBER OF DEGREES OF FREEDOM OF A MATRIX                    GAUS0350
C   L1    = FIRST ROW TO BE WORKED WITH                                 GAUS0360
C   L2    = LAST ROW TO BE WORKED WITH                                  GAUS0370
C   MD    = ADDRESS OF THE MAIN DIAGONAL TERMS OF A                     GAUS0380
C   P     = LAOD MATRIX ON INPUT, SOLN (X) OF AX=P ON COMPLEATION OF    GAUS0390
C           IOPT=4                                                      GAUS0400
C   FACTOR= TEMPORARY STORAGE MATRIX                                    GAUS0410
C   L     = ROW NUMBER FOR ELIMINATION                                  GAUS0420
C   I     = ROW NUMBER                                                  GAUS0430
C   J     = COLUMN NUMBER                                               GAUS0440
C.....................................................................  GAUS0450
C                                                                       GAUS0460
      L1=MAX(L1,1)                                                      GAUS0470
      L2=MIN(L2,N)                                                      GAUS0480
C                                                                       GAUS0490
      GO TO (10,370,370,310,410),IOPT                                   GAUS0500
C                                                                       GAUS0510
   10 CONTINUE                                                          GAUS0520
C--------- IOPT=1, REDUCE A AND P ------------------------------------ GAUS0530
C         GAUSSIAN ELIMINIATION IS USED TO REDUCE THE A AND P MATRICIES GAUS0540
C         FROM ROW L1 TO ROW L2.  IF L1 IS NOT 1, IT IS ASSUMED THAT    GAUS0550
C         A AND P ARE ALLREADY REDUCED FROM 1 TO L1...                  GAUS0560
C-----------------------------------------------------------------      GAUS0570
C                                                                       GAUS0580
      DO 100 L=L1,L2                                                    GAUS0590
        IF (L.EQ.N) GO TO 100                                           GAUS0600
C                                                                       GAUS0610
C------- TEST FOR ZERO PIVOT..... ..... ...                            GAUS0620
        IF (A(MD(L)).EQ 0) THEN                                         GAUS0630
          LJ=L                                                          GAUS0640
          GOTO 375                                                      GAUS0650
        ENDIF                                                           GAUS0660
C                                                                       GAUS0670
C  ----- CALCULATE THE FACTORS EACH ROW IS MULTIPLIED BY                GAUS0680
        DO 30 I=(L+1),N                                                 GAUS0690
          LI=IJI(L,I)                                                   GAUS0700
          IF ( LI.LT.MD(I+1) ) THEN                                     GAUS0710
                FACTOR(I)=A(LI) /A( MD(L) )                             GAUS0720
          ELSE                                                          GAUS0730
                FACTOR(I)=0                                             GAUS0740
          ENDIF                                                         GAUS0750
   30     CONTINUE                                                      GAUS0760
C                                                                       GAUS0770
C------- REDUCE A BY COLUMNS (J) EACH ROW (I) ..                        GAUS0780
        DO 40 J=(L+1),N                                                 GAUS0790
          LJ=IJI(L,J)                                                   GAUS0800
          IF (LJ.GE.MD(J+1)) GO TO 40                                   GAUS0810
          DO 35 I=(L+1),J                                               GAUS0820
            IJ=IJI(I,J)                                                 GAUS0830
            A(IJ)=A(IJ)-FACTOR(I)*A(LJ)                                 GAUS0840
```

```
   35        CONTINUE                                                      GAUS0850
   40    CONTINUE                                                          GAUS0860
C                                                                          GAUS0870
C------- REDUCE P BY COLUMNS (J) EACH ROW (I)...                          GAUS0880
         DO 45 I=(L-1),N                                                   GAUS0890
   45     P(I)=P(I)-FACTOR(I)*P(L)                                        GAUS0900
C                                                                          GAUS0910
  100 CONTINUE                                                             GAUS0920
      RETURN                                                               GAUS0930
C                                                                          GAUS0940
  310 CONTINUE                                                             GAUS0950
C---------- IOPT=4, REDUCED MULTIPLICATION ------------------------------GAUS0960
C          SOLVE FOR V IN S*D=P WHERE S HAS BEEN REDUCED BY GAUSSIAN       GAUS0970
C          ELIMINATION.                                                    GAUS0980
C          THE RESULTING LOAD IS STORED IN P.                              GAUS0990
C          THE DISPLACEMENTS ARE INPUT IN D                                GAUS1000
C          MULTIPLACATION ONLY TAKES PLACE BETWEEN ROWS L1 AND L2.         GAUS1010
C---- ---- ------------------------------------------------------------GAUS1020
C                                                                          GAUS1030
C---- - ONLY ONE LOAD CASE...                                             GAUS1040
C       ---- ZERO P ----                                                  GAUS1050
        DO 320 J=L1,L2                                                     GAUS1060
          IF (A(MD(J)).EQ.0) THEN                                          GAUS1070
            LJ=J                                                           GAUS1080
            GOTO 375                                                       GAUS1090
          ENDIF                                                            GAUS1100
  320     P(J)=0                                                           GAUS1110
C                                                                          GAUS1120
C- -------- - MULTIPLY S_RED * DISP = P                                   GAUS1130
      DO 340 I=L1,L2,+1                                                    GAUS1140
      DO 340 J=L1,L2,+1                                                    GAUS1150
        IF( I .LE. J ) IJ=MD(J)+J-I                                        GAUS1160
        IF( I .GT. J ) IJ=MD(I)+I-J                                        GAUS1170
  340     P(I)=P(I)+A(IJ)*FACTOR(J)                                        GAUS1180
C                                                                          GAUS1190
  400   CONTINUE                                                           GAUS1200
C                                                                          GAUS1210
      RETURN                                                               GAUS1220
  410 CONTINUE                                                             GAUS1230
C---------- IOPT=5, BACK SUBSTUTION -------------------------------------GAUS1240
C          SOLVE FOR D'IN S'*D'=P'                                         GAUS1250
C                                                                          GAUS1260
C    L1-->|        :      |  |   |  |                                     GAUS1270
C         |        S'     |D'|   |P'|                                     GAUS1280
C    L2-->|--------------*|--|-  |--|                                     GAUS1290
C         |        S      |D |   |P |                                     GAUS1300
C    L3-->|        :      |  |   |  |                                     GAUS1310
C          WHERE S IS THE CONDENSED MATRIX BY GAUSS ELEMINATION,           GAUS1320
C          AND IT IS A NON_POSITIVE DEFINITIVE MATRIX                      GAUS1330
C          D AND P ARE THE REDUCED DISPLACEMENT AND LOAD MATRIX            GAUS1340
C          P' ARE INPUT IN ARRAY P                                         GAUS1350
C          D ARE INPUT IN ARRARY FACTOR                                    GAUS1360
C------------------------------------------------------------------------GAUS1370
C                                                                          GAUS1380
C- ---- ONLY ONE LOAD CASE....                                            GAUS1390
C                                                                          GAUS1400
        DO 720 J=L1,L2                                                     GAUS1410
          IF (A(MD(J)).EQ.0) THEN                                          GAUS1420
            LJ=J                                                           GAUS1430
            GOTO 375                                                       GAUS1440
          ENDIF                                                            GAUS1450
  720   CONTINUE                                                           GAUS1460
C                                                                          GAUS1470
C-------CALCULATE DISPLACEMENT MATRIX , BACK SUBSTUTION ONLY              GAUS1480
C       TAKES PLACE BETWEEN L1 AND L2                                      GAUS1490
        L3=N                                                               GAUS1500
        DO 350 I=L2,L1,-1                                                  GAUS1510
          FACTOR(I)=0.                                                     GAUS1520
          DO 350 J=I,L3,+1                                                 GAUS1530
            IJ=MD(J)+J-I                                                   GAUS1540
            IF( IJ .EQ. MD(I) ) THEN                                       GAUS1550
              FACTOR(I)=FACTOR(I)+P(I)/A(MD(I))                            GAUS1560
            ELSE IF( IJ .LT. MD(J+1) ) THEN                                GAUS1570
              FACTOR(I)=FACTOR(I)- A(IJ)*FACTOR(J)/A(MD(I))                GAUS1580
            ENDIF                                                          GAUS1590
  350   CONTINUE                                                           GAUS1600
        GOTO 370                                                           GAUS1610
  375   WRITE(108,2000) LJ,MD(LJ),A(MD(LJ))                               GAUS1620
 2000   FORMAT (///1X,50('*')/                                            GAUS1630
     &            ' *  ER','ROR IN SUBROUTINE GAUSS',       T51,'*'/     GAUS1640
     &            ' *  PIVOT IS LE 0 ------ ROW:',I5,       T51,'*'/     GAUS1650
     &            ' *                    STORAGE LCN:',I5,  T51,'*'/     GAUS1660
     &            ' *                          PIVOT:',1P,G15.6,T51,'*'/ GAUS1670
     &            ' *  SOLUTION IS ABORTED ',               T51,'*'/     GAUS1680
     &          1X,50('*')/)                                               GAUS1690
        STOP                                                               GAUS1700
  370   RETURN                                                             GAUS1710
        END                                                                GAUS1720
C                                                                          CURV0010
      SUBROUTINE CURVA( NSEG,ISEG,DEC,DCU,DCV,R,EL,PR,DXA,DYA,DZA,        CURV0020
     &                  DROXA,DROYA,DROZA,DXB,DYB,DZB,DROXB,DROYB,DROZB) CURV0030
      DIMENSION R(12,12),DA(6),DDA(6),DB(6),DDB(6),PR(NSEG,9),EL(NSEG)     CURV0040
C                                                                          CURV0050
      DO 52 I=1,12                                                         CURV0060
      DO 52 J=1,12                                                         CURV0070
        R(I,J)=0.                                                          CURV0080
   52 CONTINUE                                                             CURV0090
C                                                                          CURV0100
      I=1                                                                  CURV0110
   72 R(I,I)=PR(ISEG,1)                                                    CURV0120
      R(I,I+1)=PR(ISEG,2)                                                  CURV0130
      R(I,I+2)=PR(ISEG,3)                                                  CURV0140
      R(I+1,I)=PR(ISEG,4)                                                  CURV0150
      R(I+1,I+1)=PR(ISEG,5)                                                CURV0160
      R(I-1,I+2)=PR(ISEG,6)                                                CURV0170
      R(I+2,I)=PR(ISEG,7)                                                  CURV0180
      R(I+2,I-1)=PR(ISEG,8)                                                CURV0190
      R(I+2,I-2)=PR(ISEG,9)                                                CURV0200
      I=I+3                                                                CURV0210
      IF( I .EQ. 13 ) GO TO 62                                             CURV0220
      GO TO 72                                                             CURV0230
   62 DA(1)=DXA                                                            CURV0240
      DA(2)=DYA                                                            CURV0250
      DA(3)=DZA                                                            CURV0260
      DA(4)=DROXA                                                          CURV0270
      DA(5)=DROYA                                                          CURV0280
      DA(6)=DROZA                                                          CURV0290
      DB(1)=DXB                                                            CURV0300
      DB(2)=DYB                                                            CURV0310
      DB(3)=DZB                                                            CURV0320
      DB(4)=DROXB                                                          CURV0330
      DB(5)=DROYB                                                          CURV0340
      DB(6)=DROZB                                                          CURV0350
C                                                                          CURV0360
      DO 710 II=1,6                                                        CURV0370
        DDA(II)=0                                                          CURV0380
      DO 710 KK=1,6                                                        CURV0390
  710   DDA(II)=DDA(II)+R(II,KK)*DA(KK)                                    CURV0400
      DO 720 II=1,6                                                        CURV0410
        DDB(II)=0                                                          CURV0420
      DO 720 KK=1,6                                                        CURV0430
  720   DDB(II)=DDB(II)+R(II,KK)*DB(KK)                                    CURV0440
C                                                                          CURV0450
C  ---------------------------------------                                 CURV0460
C  -- FIND SEGMENT BASIC DEFORMATION INCREMENTS --                        CURV0470
C  ---------------------------------------                                 CURV0480
      DEC=(DDB(3)-DDA(3))/EL(ISEG)                                         CURV0490
      DCU=(DDB(4)-DDA(4))/EL(ISEG)                                         CURV0500
      DCV=(DDB(5)-DDA(5))/EL(ISEG)                                         CURV0510
C                                                                          CURV0520
      RETURN                                                               CURV0530
      END                                                                  CURV0540
C                                                                          CLDA0010
      SUBROUTINE CLDAF(PRM,PR,I,NSEG,XGLO,PSX,SK,PGLOB)                    CLDA0020
C------------------------------------------------------------------------CLDA0030
C     CALCULATE SEGMENT GLOBAL FORCES                                      CLDA0040
C------------------------------------------------------------------------CLDA0050
      DIMENSION XGLO(NSEG,12),PSX(NSEG,12),SK(12,12),PR(NSEG,9),           CLDA0060
     &    PGLOB(NSEG,12),DLOCAL(12),PRM(12,12),DFLOC(12),DPGLO(12)         CLDA0070
C                                                                          CLDA0080
C PREVIOUS STEP ELEMENT STIFFNESS                                          CLDA0090
C                                                                          CLDA0100
      DO 98 II=1,12                                                        CLDA0110
      DO 98 JJ=1,12                                                        CLDA0120
   98   SK(II,JJ)=PSX(I,II,JJ)                                            CLDA0130
      DO 32 L=1,12                                                         CLDA0140
      DO 32 J=1,12                                                         CLDA0150
   32   PRM(L,J)=0.                                                       CLDA0160
  959 K=1                                                                  CLDA0170
   42 PRM(K,K)=PR(I,1)                                                     CLDA0180
      PRM(K,K+1)=PR(I,2)                                                   CLDA0190
      PRM(K,K+2)=PR(I,3)                                                   CLDA0200
      PRM(K+1,K)=PR(I,4)                                                   CLDA0210
      PRM(K+1,K+1)=PR(I,5)                                                 CLDA0220
      PRM(K+1,K+2)=PR(I,6)                                                 CLDA0230
      PRM(K+2,K)=PR(I,7)                                                   CLDA0240
      PRM(K+2,K+1)=PR(I,8)                                                 CLDA0250
      PRM(K+2,K+2)=PR(I,9)                                                 CLDA0260
      K=K+3                                                                CLDA0270
      IF (K.EQ.13) GO TO 92                                                CLDA0280
      GO TO 42                                                             CLDA0290
   92 DO 610 II=1,12                                                       CLDA0300
        DLOCAL(II)=0                                                       CLDA0310
      DO 610 KK=1,12                                                       CLDA0320
  610   DLOCAL(II)=DLOCAL(II)+PRM(II,KK)*XGLO(I,KK)                        CLDA0330
      DO 620 II=1,12                                                       CLDA0340
        DFLOC(II)=0                                                        CLDA0350
      DO 620 KK=1,12                                                       CLDA0360
  620   DFLOC(II)=DFLOC(II)+SK(II,KK)*DLOCAL(KK)                           CLDA0370
      DO 640 II=1,12                                                       CLDA0380
       DPGLO(II)=0                                                         CLDA0390
       DO 630 KK=1,12                                                      CLDA0400
  630  DPGLO(II)=DPGLO(II)+PRM(KX,II)*DFLOC(KK)                            CLDA0410
       PGLOB(I,II)=DPGLO(II)+PGLOB(I,II)                                   CLDA0420
  640 CONTINUE                                                             CLDA0430
       RETURN                                                              CLDA0440
       END                                                                 CLDA0450
C                                                                          LIBN0010
      SUBROUTINE LIB(NNOO,HH,UU,WW,ZZ,INEB,INEH,ST,IA1,IA2,IA3,            LIBN0020
     &               IA4,IECOP,U0,V0,A0,B,ECCX0,ECCY0,NUMP)                LIBN0030
C------------------------------------------------------------------------LIBN0040
C  SECT1 : BOX SECTION                                                     LIBN0050
C  SECT2 : TUBE SECTION WITH ONE LAYER                                     LIBN0060
C  SECT3 : RECTANGULAR CROSS SECTION                                       LIBN0070
C  SECT4 . WIDE-FLANGE SECTION                                             LIBN0080
C  SECT5   NOT AVAILABLE NOW                                               LIBN0090
C  SECT6 : TUBE SECTION WITH TWO LAYERS                                    LIBN0100
C  SECT7 : ANGLE SECTION WITH EQUAL LEGS                                   LIBN0110
C  SECT8 : ANGLE SECTION WITH EQUAL LEGS( OPPOSIVE SITE )                  LIBN0120
C- ---------- ------------------------------------------------------------LIBN0130
      REAL INEB,INEH                                                       LIBN0140
      DIMENSION U0(NUMP),V0(NUMP),A0(NUMP),B(NUMP)                         LIBN0150
C                                                                          LIBN0160
C THIS SUBROUTINE IS FOR SEARCHING MEMBER'S CROSS-SECTION LIBRARY          LIBN0170
C                                                                          LIBN0180
      IF(NNOO .EQ. 1 ) THEN                                                LIBN0190
          CALL SECT1(HH,UU,WW,ZZ,INEB,INEH,ST,                             LIBN0200
     &               U0,V0,A0,B,ECCX0,ECCY0,NUMP,IECOP)                    LIBN0210
      ELSE IF(NNOO .EQ. 2 ) THEN                                           LIBN0220
          CALL SECT2(HH,WW,ZZ,INEB,ST,U0,V0,A0,B,ECCX0,                    LIBN0230
     &               ECCY0,NUMP,IECOP)                                     LIBN0240
      ELSE IF(NNOO .EQ. 3 ) THEN                                           LIBN0250
          CALL SECT3(HH,UU,WW,ZZ,INEB,INEH,U0,V0,A0,B,ECCX0,               LIBN0260
     &               ECCY0,NUMP,IECOP)                                     LIBN0270
      ELSE IF(NNOO .EQ. 4 ) THEN                                           LIBN0280
          CALL SECT4(HH,UU,WW,ZZ,INEB,ST,U0,V0,A0,B,ECCX0,                 LIBN0290
     &               ECCY0,NUMP,IA1,IA2,IA3,IA4,IECOP)                     LIBN0300
      ELSE IF(NNOO .EQ. 5 ) THEN                                           LIBN0310
          CALL SECT5(HH,UU,WW,INEB,INEH,ST,U0,V0,A0,B,NUMP,IECOP)          LIBN0320
      ELSE IF(NNOO .EQ. 6 ) THEN                                           LIBN0330
          CALL SECT6(HH,WW,ZZ,INEB,ST,U0,V0,A0,B,ECCX0,                    LIBN0340
     &               ECCY0,NUMP,IECOP)                                     LIBN0350
      ELSE IF(NNOO .EQ. 7 ) THEN                                           LIBN0360
          CALL SECT7(HH,WW,ZZ,ST,U0,V0,A0,B,ECCX0,ECCY0,NUMP,              LIBN0370
     &               IA1,IA2,IA3,IA4,IECOP)                                LIBN0380
      ELSE IF(NNOO .EQ. 8 ) THEN                                           LIBN0390
          CALL SECT8(HH,WW,ZZ,ST,U0,V0,A0,B,ECCX0,ECCY0,NUMP,              LIBN0400
     &               IA1,IA2,IA3,IA4)                                      LIBN0410
      ENDIF                                                                LIBN0420
      RETURN                                                               LIBN0430
      END                                                                  LIBN0440
C                                                                          SECT0010
C                                                                          SECT0020
      SUBROUTINE SECT1(HH,UU,ECCX0,ECCY0,INEB,INEH,ST,                     SECT0030
     &                 U0,V0,A0,B,ECCX0,ECCY0,NUMP,IECOP)                  SECT0040
C------------------------------------------------------------------------SECT0050
C     THIS SUBROUTINE IS FOR DEFINING SECTIONAL COORDINATES FOR            SECT0060
C     THE BOX CROSS-SECTION                                                SECT0070
C        HH   - BOX HIGH                                                   SECT0080
C        UU   - BOX WIDTH                                                  SECT0090
C        INEB - TOTAL NO. OF ELEMENTS FOR HALF WIDTH                       SECT0100
C        INEH - TOTAL NO. OF ELEMENTS FOR HALF HIGH                        SECT0110
C        ST   - THICKNESS                                                  SECT0120
C------------------------------------------------------------------------SECT0130
      REAL INEB,INEH                                                       SECT0140
      DIMENSION U0(NUMP),V0(NUMP),A0(NUMP),B(NUMP)                         SECT0150
C                                                                          SECT0160
      UU=UU-2*ST                                                           SECT0170
      NUMP=4*(INEB+INEH)                                                   SECT0180
      ELH=((HH/2.)-ST)/(INEH-1)                                            SECT0190
      ELB=(UU/2.)/INEB                                                     SECT0200
      U0(1)=(UU/2.)+(ST/2.)                                                SECT0210
      V0(1)=-((HH/2.)-(ST/2.))                                             SECT0220
      A0(1)=ST*ST                                                          SECT0230
      B(1)=ST                                                              SECT0240
      DO 52 I=2,INEH                                                       SECT0250
         V0(I)=-((HH/2.)-(ST+ELH/2.)-(I-2)*ELH)                            SECT0260
         U0(I)=(UU/2.)+(ST/2.)                                             SECT0270
         A0(I)=ELH*ST                                                      SECT0280
         B(I)=ELH                                                          SECT0290
   52 CONTINUE                                                             SECT0300
      DO 53 I=1,INEB                                                       SECT0310
         J=I+INEH                                                          SECT0320
         V0(J)=-((HH/2.)-(ST/2.))                                          SECT0330
         U0(J)=((UU/2.)-(ELB/2.))-(I-1)*ELB                                SECT0340
         A0(J)=ELB*ST                                                      SECT0350
         B(J)=ELB                                                          SECT0360
   53 CONTINUE                                                             SECT0370
      NEOQ=INEB+INEH                                                       SECT0380
      DO 54 I=1,NEOQ                                                       SECT0390
         J=NEOQ+I                                                          SECT0400
         U0(J)=U0(I)                                                       SECT0410
         V0(J)=-V0(I)                                                      SECT0420
         A0(J)=A0(I)                                                       SECT0430
         B(J)=B(I)                                                         SECT0440
         JJ=(NEOQ*2)+I                                                     SECT0450
         U0(JJ)= U0(I)                                                     SECT0460
         V0(JJ)=V0(I)                                                      SECT0470
         A0(JJ)=A0(I)                                                      SECT0480
         B(JJ)=B(I)                                                        SECT0490
         JJJ=(NEOQ*3)+I                                                    SECT0500
         U0(JJJ)=-U0(I)                                                    SECT0510
         V0(JJJ)=-V0(I)                                                    SECT0520
         A0(JJJ)=A0(I)                                                     SECT0530
         B(JJJ)=B(I)                                                       SECT0540
   54 CONTINUE                                                             SECT0550
      IF( IECOP .EQ. 1 ) THEN                                              SECT0560
         ECCX0=ECCX0                                                       SECT0570
         ECCY0=ECCY0                                                       SECT0580
      ENDIF                                                                SECT0590
      RETURN                                                               SECT0600
      END                                                                  SECT0610
```

```
C                                                                       SECT0010
      SUBROUTINE SECT2(RAD,ECCXO,ECCYO,INEB,ST,U0,V0,A0,B,ECCX0,        SECT0020
     &                 ECCY0,NUMP,IECOP)                                SECT0030
C-----------------------------------------------------------------------SECT0040
C     THIS SUBROUTINE IS FOR DEFINING SECTIONAL COORDINATES FOR          SECT0050
C     THE TUBE CROSS-SECTION                                            SECT0060
C       RAD    -  RADIUS OF THE TUBE (BASED ON CENTER LINE             SECT0070
C       INEB   -  TOTAL NO. OF ELEMENTS FOR 1/4 OF THE CIRCLE          SECT0080
C       ST     -  THICKNESS                                            SECT0090
C-----------------------------------------------------------------------SECT0100
      REAL INEB                                                         SECT0110
      DIMENSION U0(NUMP),V0(NUMP),A0(NUMP),B(NUMP)                      SECT0120
      NUMP=4*INEB                                                       SECT0130
      PI=2.*3.1415926                                                   SECT0140
      SITA=(PI/4.)/(2*INEB)                                             SECT0150
      TEMP=2.*SITA                                                      SECT0160
      DO 25 I=1,INEB                                                    SECT0170
        V0(I)=SQRT(RAD*RAD/(1.+TAN(SITA)*TAN(SITA)))                    SECT0180
        U0(I)=V0(I)*TAN(SITA)                                           SECT0190
   25   SITA=SITA+TEMP                                                  SECT0200
      DO 30 I=INEB+1,2*INEB                                             SECT0210
        V0(I)=V0(I-INEB)                                                SECT0220
   30   U0(I)=-U0(I-INEB)                                               SECT0230
      DO 35 I=2*INEB+1,3*INEB                                           SECT0240
        V0(I)=-V0(I-2*INEB)                                             SECT0250
   35   U0(I)=U0(I-2*INEB)                                              SECT0260
      DO 40 I=3*INEB+1,4*INEB                                           SECT0270
        V0(I)=-V0(I-3*INEB)                                             SECT0280
   40   U0(I)=-U0(I-3*INEB)                                             SECT0290
      DO 50 I=1,NUMP                                                    SECT0300
        B(I)=(PI*RAD)/NUMP                                              SECT0310
        A0(I)=B(I)*ST                                                   SECT0320
   50 CONTINUE                                                          SECT0330
      IF(IECOP .EQ. 1 ) THEN                                            SECT0340
        ECCX0=ECCXO                                                     SECT0350
        ECCY0=ECCYO                                                     SECT0360
      ENDIF                                                             SECT0370
      RETURN                                                            SECT0380
      END                                                               SECT0390
C                                                                       SECT0010
      SUBROUTINE SECT3(HH,UU,ECCXO,ECCYO,INEB,INEH,U0,V0,A0,B,ECCX0,    SECT0020
     &                 ECCY0,NUMP,IECOP)                                SECT0030
C-----------------------------------------------------------------------SECT0040
C     THIS SUBROUTINE IS FOR DEFINING SECTIONAL COORDINATES FOR          SECT0050
C     THE RECTANGULAR CROSS-SECTION                                     SECT0060
C       HH   - SECTIONAL HIGH                                           SECT0070
C       UU   - SECTIONAL WIDTH                                          SECT0080
C       INEB - TOTAL NO. OF ELEMENTS FOR HALF WIDTH                     SECT0090
C       INEH - TOTAL NO. OF ELEMENTS FOR HALF HIGH                      SECT0100
C       B(1) - HH/2 (LARGER LENGTH)                                     SECT0110
C       B(2) - UU/2 (SMALLER LENGTH)                                    SECT0120
C       NOTE : B(1) AND B(2) ARE FOR CALCULATING TORSIONAL RIGIDITY     SECT0130
C-----------------------------------------------------------------------SECT0140
      REAL INEB,INEH                                                    SECT0150
      DIMENSION U0(NUMP),V0(NUMP),A0(NUMP),B(NUMP)                      SECT0160
C                                                                       SECT0170
      NUMP=4*(INEB*INEH)                                                SECT0180
      ELH=HH/(2.*INEH)                                                  SECT0190
      ELB=UU/(2.*INEB)                                                  SECT0200
C                                                                       SECT0210
      DO 50 M=1,INEB                                                    SECT0220
       II=(M-1)*INEH + 1                                                SECT0230
       U0(II)=(UU/2)-(ELB/2)-(M-1)*ELB                                  SECT0240
       V0(II)=(HH/2)-(ELH/2)                                            SECT0250
       A0(II)=ELH*ELB                                                   SECT0260
       J=1                                                              SECT0270
       DO 60 I=(M-1)*INEH+2, M*INEH                                     SECT0280
          J=J+1                                                         SECT0290
          U0(I)=U0(II)                                                  SECT0300
          V0(I)=(HH/2)-(J-1)*ELH-(ELH/2)                                SECT0310
          A0(I)=ELH*ELB                                                 SECT0320
   60  CONTINUE                                                         SECT0330
   50 CONTINUE                                                          SECT0340
C                                                                       SECT0350
      IQRT=INEB*INEH                                                    SECT0360
      DO 70 I=IQRT+1,2*IQRT                                             SECT0370
         U0(I)=U0(I-IQRT)                                               SECT0380
         V0(I)=-V0(I-IQRT)                                              SECT0390
         A0(I)=ELH*ELB                                                  SECT0400
   70 CONTINUE                                                          SECT0410
      DO 80 I=2*IQRT+1,3*IQRT                                           SECT0420
         U0(I)=-U0(I-2*IQRT)                                            SECT0430
         V0(I)=-V0(I-2*IQRT)                                            SECT0440
         A0(I)=ELH*ELB                                                  SECT0450
   80 CONTINUE                                                          SECT0460
      DO 90 I=3*IQRT+1,4*IQRT                                           SECT0470
         U0(I)=-U0(I-3*IQRT)                                            SECT0480
         V0(I)=-V0(I-3*IQRT)                                            SECT0490
         A0(I)=ELH*ELB                                                  SECT0500
   90 CONTINUE                                                          SECT0510
C                                                                       SECT0520
      IF( HH .GE. UU ) THEN                                             SECT0530
         B(1)=HH/2                                                      SECT0540
         B(2)=UU/2                                                      SECT0550
      ELSE IF( HH .LT. UU ) THEN                                        SECT0560
         B(1)=UU/2                                                      SECT0570
         B(2)=HH/2                                                      SECT0580
      ENDIF                                                             SECT0590
C                                                                       SECT0600
      IF( IECOP .EQ. 1 ) THEN                                           SECT0610
         ECCX0=ECCXO                                                    SECT0620
         ECCY0=ECCYO                                                    SECT0630
      ENDIF                                                             SECT0640
      RETURN                                                            SECT0650
      END                                                               SECT0660
C                                                                       SECT0010
      SUBROUTINE SECT4(HH,UU,ECCXO,ECCYO,INEB,ST,U0,V0,A0,B,ECCX0,      SECT0020
     &                 ECCY0,NUMP,IREV1,IREV2,IREV3,IREV4,IECOP)        SECT0030
C-----------------------------------------------------------------------SECT0040
C     THIS SUBROUTINE IS FOR DEFINING SECTIONAL COORDINATES FOR          SECT0050
C     THE WIDE-FLANGE SECTION                                           SECT0060
C          ST    - FLANGE THICKNESS                                     SECT0070
C          INEB  - WEB THICKNESS                                        SECT0080
C          HH    - SECTION HIGH                                         SECT0090
C          UU    - SECTION WIDTH                                        SECT0100
C          IREV1- NO. OF FLANGE COLUMN IN THE REFERENCE AXIS U DIRECTIONSECT0110
C          IREV2- NO. OF FLANGE ROW    IN THE REFERENCE AXIS U DIRECTIONSECT0120
C          IREV3- NO. OF WEB COLUMN    IN THE REFERENCE AXIS V DIRECTIONSECT0130
C          IREV4- NO. OF WEB ROW       IN THE REFERENCE AXIS V DIRECTIONSECT0140
C-----------------------------------------------------------------------SECT0150
      REAL INEB                                                         SECT0160
      DIMENSION U0(NUMP),V0(NUMP),A0(NUMP),B(NUMP)                      SECT0170
C                                                                       SECT0180
      NUMP=IREV1*IREV2*2 + IREV3*IREV4                                  SECT0190
      WEBHIGH = HH - 2*ST                                               SECT0200
      SS = UU/IREV2                                                     SECT0210
      TT = ST/IREV1                                                     SECT0220
      SSS = INEB/IREV3                                                  SECT0230
      TTT = WEBHIGH/IREV4                                               SECT0240
      B(1)= (2*UU*ST**3 + WEBHIGH*INEB**3)/3                            SECT0250
C                                                                       SECT0260
C....CALCULATE TOP FLANGE ELEMENT COORDINATES                           SECT0270
      K=0                                                               SECT0280
      DO 50 I=1, IREV1                                                  SECT0290
        DO 50 J=1,IREV2                                                 SECT0300
           K=K+1                                                        SECT0310
           U0(K)=( UU/2) + 0.5*SS + (J-1)*SS                            SECT0320
           V0(K)=( HH/2) - 0.5*TT - (I-1)*TT                            SECT0330
           A0(K)=SS*TT                                                  SECT0340
   50 CONTINUE                                                          SECT0350
C                                                                       SECT0360
C ...CALCULATE BOTTOM FLANGE ELEMENT COORDINATES                        SECT0370
      DO 60 I=1, IREV1                                                  SECT0380
        DO 60 J=1,IREV2                                                 SECT0390
           K=K+1                                                        SECT0400
           U0(K)=(-UU/2) + 0.5*SS + (J-1)*SS                            SECT0410
           V0(K)=(-HH/2) + 0.5*TT + (I-1)*TT                            SECT0420
           A0(K)=SS*TT                                                  SECT0430
   60 CONTINUE                                                          SECT0440
C                                                                       SECT0450
C...CALCULATE WEB ELEMENT COORDINATES                                   SECT0460
      DO 70 I=1, IREV4                                                  SECT0470
        DO 70 J=1,IREV3                                                 SECT0480
           K=K+1                                                        SECT0490
           U0(K)=(-INEB/2) + 0.5*SSS + (J-1)*SSS                        SECT0500
           V0(K)=(WEBHIGH/2) - 0.5*TTT   (I-1)*TTT                      SECT0510
           A0(K)=SSS*TTT                                                SECT0520
   70 CONTINUE                                                          SECT0530
C                                                                       SECT0540
      IF( IECOP .EQ. 1 ) THEN                                           SECT0550
         ECCX0=ECCXO                                                    SECT0560
         ECCY0=ECCYO                                                    SECT0570
      ENDIF                                                             SECT0580
      RETURN                                                            SECT0590
      END                                                               SECT0600
C                                                                       SECT0010
      SUBROUTINE SECT5(BB,T,D,NCS,NPS,T1,U0,V0,A0,B,NUMP,IECOP)         SECT0020
C-----------------------------------------------------------------------SECT0030
C     THIS SUBROUTINE IS FOR DEFINING SECTIONAL COORDINATES FOR          SECT0040
C     THE OPEN WEB JOIST PLATE-CHORD CROSS-SECTION(BOTTOM)              SECT0050
C          BB   - PLATE WIDTH                                           SECT0060
C          T    - PLATE THICKNESS                                       SECT0070
C          T1   - CHORD THICKNESS                                       SECT0080
C          NPS  - TOTAL NO. OF ELEMENTS FOR HALF PLATE WIDTH            SECT0090
C          NCS  - TOTAL NO. OF ELEMENTS FOR EACH ANGLE CHORD'S TOE      SECT0100
C          D    - THE ANGLE'S LENGTH                                    SECT0110
C-----------------------------------------------------------------------SECT0120
      REAL NCS,NPS                                                      SECT0130
      DIMENSION U0(NUMP),V0(NUMP),A0(NUMP),B(NUMP)                      SECT0140
      IF (IECOP .EQ. 1 ) THEN                                           SECT0150
        WRITE(108,*) '1','SECT5 LIBRARY FOR ECCENTRIC LOAD IS NOT',     SECT0160
     $               'AVAILABLE..... ... ..'                            SECT0170
        IECOP=0                                                         SECT0180
      ENDIF                                                             SECT0190
      NUMP=4*NCS*2*NPS*2                                                SECT0200
      S=-((-.5*BB*T**2)-(4.*T1*T*D+D*T1**2 +T1*D**2*2.*T*T1**2-T1**3))  SECT0210
     $   /(BB*T+4.*T1*D*2.*T1**2)                                       SECT0220
      DS=(D-T1)/(2.*NCS)                                                SECT0230
      U0(1)=(BB/2.)-(T1/2.)                                             SECT0240
      V0(1)=-(S-(T+0.5*T1))                                             SECT0250
      A0(1)=T1**2                                                       SECT0260
      B(1)=T1                                                           SECT0270
      DO 52 I=2,NCS+1                                                   SECT0280
        U0(I)=(0.5*BB-D-DS)+2.*DS*(I-1)                                 SECT0290
        V0(I)=-(S-(T+0.5*T1))                                           SECT0300
        A0(I)=2.*DS*T1                                                  SECT0310
        B(I)=2.*DS                                                      SECT0320
   52 CONTINUE                                                          SECT0330
       DO 54 I=NCS+2, 2*NCS+1                                           SECT0340
        U0(I)=0.5*BB-0.5*T1                                             SECT0350
        V0(I)=(D+T-S)-DS-2.*DS*(I-NCS-1)                                SECT0360
        A0(I)=2.*DS*T1                                                  SECT0370
        B(I)=2.*DS                                                      SECT0380
   54 CONTINUE                                                          SECT0390
      U0(2*NCS+2)=-U0(1)                                                SECT0400
      V0(2*NCS+2)=V0(1)                                                 SECT0410
      A0(2*NCS+2)=A0(1)                                                 SECT0420
      B(2*NCS+2)=B(1)                                                   SECT0430
      DO 56 I=2*NCS+3, 3*NCS+2                                          SECT0440
        U0(I)=-(0.5*BB-D-DS)-2.*DS*(I-2*NCS-2)                          SECT0450
        V0(I)=-(S-(T+0.5*T1))                                           SECT0460
        A0(I)=2.*DS*T1                                                  SECT0470
        B(I)=2.*DS                                                      SECT0480
   56 CONTINUE                                                          SECT0490
      DO 58 I=3*NCS+3, 4*NCS+2                                          SECT0500
        U0(I)=-0.5*BB+0.5*T1                                            SECT0510
        V0(I)=(D-T-S)-DS-2.*DS*(I-3*NCS-2)                              SECT0520
        A0(I)=2.*DS*T1                                                  SECT0530
        B(I)=2.*DS                                                      SECT0540
   58 CONTINUE                                                          SECT0550
      DBB=0.5*BB/(2.*NPS)                                               SECT0560
      DO 60 I=4*NCS+3, 4*NCS+3*NPS-1                                    SECT0570
        U0(I)=0.5*BB+DBB-2.*DBB*(I-4*NCS-2)                             SECT0580
        V0(I)=-(S-0.5*T)                                                SECT0590
        A0(I)=2.*DBB*T                                                  SECT0600
        B(I)=2.*DBB                                                     SECT0610
   60 CONTINUE                                                          SECT0620
      DO 64 I=4*NCS+NPS+3, 4*NCS+2*NPS+2                                SECT0630
        U0(I)=-0.5*BB-DBB+2.*DBB*(I-4*NCS-NPS-2)                        SECT0640
        V0(I)=-(S-0.5*T)                                                SECT0650
        A0(I)=2.*DBB*T                                                  SECT0660
        B(I)=2.*DBB                                                     SECT0670
   64 CONTINUE                                                          SECT0680
      RETURN                                                            SECT0690
      END                                                               SECT0700
C                                                                       SECT0010
      SUBROUTINE SECT6(RAD,ECCXO,ECCYO,INEB,ST,U0,V0,A0,B,ECCX0,        SECT0020
     &                 ECCY0,NUMP,IECOP)                                SECT0030
C-----------------------------------------------------------------------SECT0040
C     THIS SUBROUTINE IS FOR DEFINING SECTIONAL COORDINATES FOR          SECT0050
C     THE TUBE CROSS-SECTIONWHICH HAS TWO LAYERS                        SECT0060
C       RAD    -  RADIUS OF THE TUBE (BASED ON CENTER LINE             SECT0070
C       INEB   -  TOTAL NO. OF ELEMENTS FOR 1/4 OF THE CIRCLE FOR      SECT0080
C                 EACH LAYER                                            SECT0090
C       ST     -  THICKNESS                                            SECT0100
C-----------------------------------------------------------------------SECT0110
      REAL INEB                                                         SECT0120
      DIMENSION U0(NUMP),V0(NUMP),A0(NUMP),B(NUMP)                      SECT0130
      NUMP=8*INEB                                                       SECT0140
      NNUMP=NUMP/2                                                      SECT0150
      RAD1=RAD+(ST/4.)                                                  SECT0160
      RAD2=RAD-(ST/4.)                                                  SECT0170
      PI=2.*3.1415926                                                   SECT0180
      SITA=(PI/4.)/(2*INEB)                                             SECT0190
      TEMP=2.*SITA                                                      SECT0200
      DO 25 I=1,INEB                                                    SECT0210
        V0(I)=SQRT(RAD1*RAD1/(1.+TAN(SITA)*TAN(SITA)))                  SECT0220
        U0(I)=V0(I)*TAN(SITA)                                           SECT0230
        V0(I+NNUMP)=SQRT(RAD2*RAD2/(1.+TAN(SITA)*TAN(SITA)))            SECT0240
        U0(I+NNUMP)=V0(I+NNUMP)*TAN(SITA)                               SECT0250
   25   SITA=SITA+TEMP                                                  SECT0260
      DO 30 I=INEB+1,2*INEB                                             SECT0270
        V0(I)=V0(I-INEB)                                                SECT0280
        U0(I)=-U0(I-INEB)                                               SECT0290
        V0(I+NNUMP)=V0(I-INEB+NNUMP)                                    SECT0300
   30   U0(I+NNUMP)=-U0(I-INEB+NNUMP)                                   SECT0310
      DO 35 I=2*INEB+1,3*INEB                                           SECT0320
        V0(I)=-V0(I-2*INEB)                                             SECT0330
        U0(I)=U0(I-2*INEB)                                              SECT0340
        V0(I+NNUMP)=-V0(I-2*INEB+NNUMP)                                 SECT0350
   35   U0(I+NNUMP)=U0(I-2*INEB+NNUMP)                                  SECT0360
      DO 40 I=3*INEB+1,4*INEB                                           SECT0370
        V0(I)= V0(I-3*INEB)                                             SECT0380
        U0(I)=-U0(I-3*INEB)                                             SECT0390
        V0(I+NNUMP)= V0(I-3*INEB+NNUMP)                                 SECT0400
   40   U0(I+NNUMP)=-U0(I-3*INEB+NNUMP)                                 SECT0410
      DO 50 I=1,NNUMP                                                   SECT0420
        B(I)=(PI*RAD1)/NNUMP                                            SECT0430
        A0(I)=B(I)*(ST/2.)                                              SECT0440
        B(I+NNUMP)=(PI*RAD2)/NNUMP                                      SECT0450
        A0(I+NNUMP)=B(I+NNUMP)*(ST/2.)                                  SECT0460
   50 CONTINUE                                                          SECT0470
      IF(IECOP .EQ. 1 ) THEN                                            SECT0480
       ECCX0=ECCXO                                                      SECT0490
       ECCY0=ECCYO                                                      SECT0500
      ENDIF                                                             SECT0510
      RETURN                                                            SECT0520
      END                                                               SECT0530
C                                                                       SECT0010
      SUBROUTINE SECT7(D,ECCXO,ECCYO,T,U0,V0,A0,B,ECCX0,ECCY0,NUMP,     SECT0020
     &                 L1,N1,L2,N2,IECOP)                               SECT0030
C-----------------------------------------------------------------------SECT0040
```

```
C     THIS SUBROUTINE IS FOR DEFINING SECTIONAL COORDINATES FOR          SECT0050
C     THE ANGLE CROSS-SECTION                                           SECT0060
C          D    - ANGLE SIDE TOTAL LENGTH                             SECT0070
C          T    - THICKNESS                                           SECT0080
C          ECCX0- THE X-DISTANCE FROM ECCENTRIC LOAD TO SHAPE CENTER  SECT0090
C          ECCY0- THE Y-DISTANCE FROM ECCENTRIC LOAD TO SHAPE CENTER  SECT0100
C          L1   - NO. OF COLUMN IN X DIRECTION                       SECT0110
C          N1   - NO OF ROW IN X DIRECTION                           SECT0120
C          L2   - NO. OF ROW IN Y DIRECTION                          SECT0130
C          N2   - NO OF COLUMN IN Y DIRECTION                        SECT0140
C          IECOP- ECCENTRIC LOAD OPTION                              SECT0150
C                 1: ECCENTRIC LOAD APPLIED                          SECT0160
C                 0: WITHOUT ECCENTRIC LOAD                          SECT0170
C---- ------------------------------------------------------------- SECT0180
      DIMENSION U0(NUMP),V0(NUMP),A0(NUMP),B(NUMP)                    SECT0190
      BB=(D*T+D**2-T**2)/(4.*D-2.*T)                                  SECT0200
      SY1=(T/N1)*0.5                                                  SECT0210
      SX1=(D/L1)*0.5                                                  SECT0220
      DO 101 J=1,N1                                                   SECT0230
       II=J-1                                                         SECT0240
       DO 101 I=1,L1                                                  SECT0250
         M=II*L1+I                                                    SECT0260
         U0(M)=BB-(D-(2*I-1)*SX1)                                     SECT0270
         V0(M)=BB-(2*J-1)*SY1                                         SECT0280
         A0(M)=SY1*SX1*4.                                             SECT0290
         B(M)=2.*SX1                                                  SECT0300
  101 CONTINUE                                                        SECT0310
      SX2=(T/N2)*0.5                                                  SECT0320
      SY2=((D-T)/L2)*0.5                                              SECT0330
      DO 201 J=1,N2                                                   SECT0340
       II=J-1                                                         SECT0350
       DO 201 I=1,L2                                                  SECT0360
         M1=II*L2+I+(N1*L1)                                           SECT0370
         U0(M1)=BB-(2*J-1)*SX2                                        SECT0380
         V0(M1)=BB-(D-(2*I-1)*SY2)                                    SECT0390
         A0(M1)=SY2*SX2*4.                                            SECT0400
         B(M1)=2.*SY2                                                 SECT0410
  201 CONTINUE                                                        SECT0420
      NUMP=L1*N1+L2*N2                                                SECT0430
      IF(IECOP .EQ. 1 ) THEN                                          SECT0440
        ECCX0=ECCX0                                                   SECT0450
        ECCY0=ECCY0                                                   SECT0460
      ENDIF                                                           SECT0470
      RETURN                                                          SECT0480
      END                                                             SECT0490
C                                                                     SECT0500
      SUBROUTINE SECT0(D,ECCX0,ECCY0,T,U0,V0,A0,B,ECCX0,ECCY0,NUMP,   SECT0510
     $                 L1,N1,L2,N2)                                   SECT0520
C---------------------- ------------------------------------------- SECT0530
C     THIS SUBROUTINE IS FOR DEFINING SECTIONAL COORDINATES FOR       SECT0540
C     THE ANGLE CROSS-SECTION (BOTTOM CHORD)                          SECT0550
C          D    - ANGLE SIDE TOTAL LENGTH                             SECT0560
C          T    - THICKNESS                                           SECT0570
C          ECCX0- THE X-DISTANCE FROM ECCENTRIC LOAD TO SHAPE CENTER  SECT0580
C          ECCY0- THE Y-DISTANCE FROM ECCENTRIC LOAD TO SHAPE CENTER  SECT0590
C          L1   - NO. OF COLUMN IN X DIRECTION                       SECT0600
C          N1   - NO OF ROW IN X DIRECTION                           SECT0610
C          L2   - NO. OF ROW IN Y DIRECTION                          SECT0620
C          N2   - NO OF COLUMN IN Y DIRECTION                        SECT0630
C---------------------------------------------------------------------SECT0640
      DIMENSION U0(NUMP),V0(NUMP),A0(NUMP),B(NUMP)                    SECT0650
      BB=(D*T+D**2-T**2)/(4.*D-2.*T)                                  SECT0660
      SY1=(T/N1)*0.5                                                  SECT0670
      SX1=(D/L1)*0.5                                                  SECT0680
      DO 101 J=1,N1                                                   SECT0690
       II=J-1                                                         SECT0700
       DO 101 I=1,L1                                                  SECT0710
        M=II*L1+I                                                     SECT0720
        U0(M)=BB-(D-(2*I-1)*SX1)                                      SECT0730
        V0(M)=-(BB-(2*J-1)*SY1)                                       SECT0740
        A0(M)=SY1*SX1*4.                                              SECT0750
        B(M)=2.*SX1                                                   SECT0760
  101 CONTINUE                                                        SECT0770
      SX2=(T/N2)*0.5                                                  SECT0780
      SY2=((D-T)/L2)*0.5                                              SECT0790
      DO 201 J=1,N2                                                   SECT0800
       II=J-1                                                         SECT0810
       DO 201 I=1,L2                                                  SECT0820
        M1=II*L2+I+(N1*L1)                                            SECT0830
        U0(M1)=BB-(2*J-1)*SX2                                         SECT0840
        V0(M1)=-(BB-(D-(2*I-1)*SY2))                                  SECT0850
        A0(M1)=SY2*SX2*4.                                             SECT0860
        B(M1)=2.*SY2                                                  SECT0870
  201 CONTINUE                                                        SECT0880
       NUMP=L1*N1+L2*N2                                               SECT0890
       ECCX0=ECCX0                                                    SECT0900
       ECCY0=ECCY0                                                    SECT0910
      RETURN                                                          SECT0920
      END                                                             SECT0930
C                                                                     ROMA0010
C---------------------------------------------------------------------ROMA0020
C                                                                     ROMA0030
      SUBROUTINE ROMA(XA,YA,ZA,ROZA,XB,YB,ZB,ROZB,VX0,VY0,VZ0,WX0,    ROMA0040
     *               WY0,WZ0,ISEG,R,PR,BETA,TL,NSEG)                  ROMA0050
      DIMENSION R(12,12),PR(NSEG,9),BETA(NSEG),TL(NSEG)               ROMA0060
C                                                                     ROMA0070
      TL(ISEG)=SQRT((XB-XA)**2+(YB-YA)**2+(ZB-ZA)**2)                 ROMA0080
      WX=(XB-XA)/TL(ISEG)                                             ROMA0090
      WY=(YB-YA)/TL(ISEG)                                             ROMA0100
      WZ=(ZB-ZA)/TL(ISEG)                                             ROMA0110
      ROWOBA=BETA(ISEG)-0.5*(ROZA-ROZB)                               ROMA0120
      S=SIN(ROWOBA)                                                   ROMA0130
      C=COS(ROWOBA)                                                   ROMA0140
      VBAX=VX0*C-(VY0*WZ0-VZ0*WY0)*S                                  ROMA0150
      VBAY=VY0*C-(VZ0*WX0-VX0*WZ0)*S                                  ROMA0160
      VBAZ=VZ0*C-(VX0*WY0-VY0*WX0)*S                                  ROMA0170
      TM=VBAX*WX+VBAY*WY+VBAZ*WZ                                      ROMA0180
      TLV=SQRT((VBAX-TM*WX)**2+(VBAY-TM*WY)**2+(VBAZ-TM*WZ)**2)       ROMA0190
      VX=(VBAX-TM*WX)/TLV                                             ROMA0200
      VY=(VBAY-TM*WY)/TLV                                             ROMA0210
      VZ=(VBAZ-TM*WZ)/TLV                                             ROMA0220
      UX=VY*WZ-VZ*WY                                                  ROMA0230
      UY=VZ*WX-VX*WZ                                                  ROMA0240
      UZ=VX*WY-VY*WX                                                  ROMA0250
C                                                                     ROMA0260
C FIND THE ROTATION MATRIX AND THE SECTION PROPERTIES                 ROMA0270
C                                                                     ROMA0280
      DO 52 I=1,12                                                    ROMA0290
      DO 52 J=1,12                                                    ROMA0300
       R(I,J)=0.                                                      ROMA0310
   52 CONTINUE                                                        ROMA0320
      I=1                                                             ROMA0330
   72 R(I,I)=UX                                                       ROMA0340
      R(I,I+1)=UY                                                     ROMA0350
      R(I,I+2)=UZ                                                     ROMA0360
      R(I+1,I)=VX                                                     ROMA0370
      R(I+1,I+1)=VY                                                   ROMA0380
      R(I+1,I+2)=VZ                                                   ROMA0390
      R(I+2,I)=WX                                                     ROMA0400
      R(I+2,I+1)=WY                                                   ROMA0410
      R(I+2,I+2)=WZ                                                   ROMA0420
      I=I+3                                                           ROMA0430
      IF (I EQ 13) GO TO 62                                           ROMA0440
      GO TO 72                                                        ROMA0450
   62 PR(ISEG,1)=UX                                                   ROMA0460
      PR(ISEG,2)=UY                                                   ROMA0470
      PR(ISEG,3)=UZ                                                   ROMA0480
      PR(ISEG,4)=VX                                                   ROMA0490
      PR(ISEG,5)=VY                                                   ROMA0500
      PR(ISEG,6)=VZ                                                   ROMA0510
      PR(ISEG,7)=WX                                                   ROMA0520
      PR(ISEG,8)=WY                                                   ROMA0530
      PR(ISEG,9)=WZ                                                   ROMA0540
C                                                                     ROMA0550
      RETURN                                                          ROMA0560
      END                                                             ROMA0570
C                                                                     PRIN0010
C ------------------------------------------------                    PRIN0020
C SUBROUTINE PRINC FOR CALCULATE PRINCIPAL AXES                       PRIN0030
      SUBROUTINE PRINC(DEC,DCU,DCV,ISEG,ST,TOTA,LIBN,EM,EY,G,YS,IOPT, PRIN0040
     &         NUMP,NSEG,EA,EIU,EIV,GKT,IREV1,IREV2,IREV3,IREV4,UC0,  PRIN0050
     &         VC0,IMATER,A0,B,BETA,E0,E1,E2,U,V,U0,V0,TE,E,DE,INEB,  PRIN0060
     &         INEH,FJL,RATIO3,IR,STR,DEP,E0P,STRP,TEP,EREP,SREP,FLP) PRIN0070
C ------------------------------------------------                    PRIN0080
C                                                                     PRIN0090
C ECCX - X- DISTANCE FROM ECCENTRIC LOAD TO INSTANCE SHAPE-CENTER     PRIN0100
C ECCY - Y- DISTANCE FROM ECCENTRIC LOAD TO INSTANCE SHAPE-CENTER     PRIN0110
C ECX  - X- DISTANCE FROM ECCENTRIC LOAD TO ORIGINAL SHAPE-CENTER     PRIN0120
C ECY  - Y- DISTANCE FROM ECCENTRIC LOAD TO ORIGINAL SHAPE-CENTER     PRIN0130
C ------------------------------------------------------------------ PRIN0140
      REAL INEB,INEH                                                  PRIN0150
      DIMENSION A0(NUMP),B(NUMP),BETA(NSEG),DE(NSEG,NUMP)             PRIN0160
      DIMENSION E(NUMP),E0(NSEG,NUMP),E1(NSEG,NUMP),E2(NSEG,NUMP)     PRIN0170
      DIMENSION TE(NUMP),U(NSEG,NUMP),V(NSEG,NUMP),SREP(NSEG,NUMP)    PRIN0180
      DIMENSION U0(NUMP),V0(NUMP),STR(NSEG,NUMP),FJL(NUMP)            PRIN0190
      DIMENSION DEP(NSEG,NUMP),E0P(NSEG,NUMP),STRP(NSEG,NUMP)         PRIN0200
      DIMENSION TEP(NSEG,NUMP),EREP(NSEG,NUMP)                        PRIN0210
C                                                                     PRIN0220
C CALCULATE STRAIN RATE OF EACH SECTION ELEMENT                       PRIN0230
C                                                                     PRIN0240
      DO 50 I=1,NUMP                                                  PRIN0250
       DE(ISEG,I)=DEC*DCU*V(ISEG,I)-DCV*U(ISEG,I)                     PRIN0260
       E(I)=E0(ISEG,I)+DE(ISEG,I)                                     PRIN0270
       E0(ISEG,I)=E(I)                                                PRIN0280
   50 CONTINUE                                                        PRIN0290
      IF (IMATER.EQ.0) THEN                                           PRIN0300
       CALL JUDEL0(NSEG,NUMP,ISEG,EM,EY,RATIO3,DE,E,E1,E2,TE,STR )    PRIN0310
      ELSE IF (IMATER.EQ.1) THEN                                      PRIN0320
       CALL JUDEL2(NSEG,NUMP,ISEG,EM,EY,RATIO3,YS,DE,E,E1,E2,TE,STR)  PRIN0330
      ELSE IF (IMATER.EQ.2) THEN                                      PRIN0340
       CALL JUDEL3(NSEG,NUMP,ISEG,EM,IR,YS,DE,E,TE,DEP,E0P,STRP,      PRIN0350
     &           STR,TEP,EREP,SREP,IOPT)                             PRIN0360
      ELSE IF (IMATER.EQ.3) THEN                                      PRIN0370
C.... ELASTIC CASE SOLUTION                                           PRIN0380
       DO 79 I=1,NUMP                                                 PRIN0390
        TE(I)=EM                                                      PRIN0400
        STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)                      PRIN0410
   79  CONTINUE                                                       PRIN0420
      ENDIF                                                           PRIN0430
C                                                                     PRIN0440
C     JUDGE LOCAL BUCKLING FOR BOX SECTION                            PRIN0450
C                                                                     PRIN0460
      IF (LIBN.EQ.1.AND.IREV1.NE.0) CALL BOXLOC(FJL,STR,NSEG,         PRIN0470
     &                NUMP,ISEG,INEB,INEH,TE,ST,EM,U0,V0,FLP )        PRIN0480
C                                                                     PRIN0490
C CALCULATE THE SECTIONAL PROPERTIES ABOUT SECTIONAL REFERENCE         PRIN0500
C AXES WITH ORIGIN AT ( 0 , 0 )                                       PRIN0510
C                                                                     PRIN0520
      EA0=0.                                                          PRIN0530
      ESV0=0.                                                         PRIN0540
      ESU0=0                                                          PRIN0550
      EIU0=0.                                                         PRIN0560
      EIV0=0.                                                         PRIN0570
      EIUV0=0                                                         PRIN0580
      DO 60 I=1,NUMP                                                  PRIN0590
         IF( FJL(I) .EQ. 0 ) THEN                                     PRIN0600
             EA0=EA0+TE(I)*A0(I)                                      PRIN0610
             ESV0=ESV0+TE(I)*V0(I)*A0(I)                              PRIN0620
             ESU0=ESU0+TE(I)*U0(I)*A0(I)                              PRIN0630
             EIU0=EIU0+TE(I)*V0(I)*V0(I)*A0(I)                        PRIN0640
             EIV0=EIV0+TE(I)*U0(I)*U0(I)*A0(I)                        PRIN0650
             EIUV0=EIUV0+TE(I)*U0(I)*V0(I)*A0(I)                      PRIN0660
         ELSE IF( FJL(I) .EQ. 1 ) THEN                                PRIN0670
             EA0=EA0+RATIO3*EM*A0(I)                                  PRIN0680
             ESV0=ESV0+RATIO3*EM*V0(I)*A0(I)                          PRIN0690
             ESU0=ESU0+RATIO3*EM*U0(I)*A0(I)                          PRIN0700
             EIU0=EIU0+RATIO3*EM*V0(I)*V0(I)*A0(I)                    PRIN0710
             EIV0=EIV0+RATIO3*EM*U0(I)*U0(I)*A0(I)                    PRIN0720
             EIUV0=EIUV0+RATIO3*EM*U0(I)*V0(I)*A0(I)                  PRIN0730
         ENDIF                                                        PRIN0740
   60 CONTINUE                                                        PRIN0750
C                                                                     PRIN0760
C THE NEW CENTROID LOCATION CORRESPONDING TO SECTION REFERENCE AXES    PRIN0770
C                                                                     PRIN0780
      UC0=ESU0/EA0                                                    PRIN0790
      VC0=ESV0/EA0                                                    PRIN0800
C                                                                     PRIN0810
C THE SECTIONAL PROPERTIES ABOUT SECTIONAL REFERENCE AXES WITH         PRIN0820
C ORIGIN AT ( UC0,VC0 )                                               PRIN0830
C                                                                     PRIN0840
      EIU1=EIU0-VC0*VC0*EA0                                           PRIN0850
      EIV1=EIV0-UC0*UC0*EA0                                           PRIN0860
      EIUV1=EIUV0-UC0*VC0*EA0                                         PRIN0870
      IF( EIU1 .LE. 0. .OR. EIV1 .LE  0. ) THEN                       PRIN0880
       WRITE(108,*) ' ** INEFFICIENT SECTION ELEMENT ARRENGEMENT **'  PRIN0890
       WRITE(108,*) '    ----ERROR--------ERROR---------ERROR----    '  PRIN0900
       WRITE(108,*) ' **          EIU1 OR EIV1 IS NEGATIVE          **'  PRIN0910
      ENDIF                                                           PRIN0920
C                                                                     PRIN0930
      IF( ABS((EIV1-EIU1)/EIV1) .LE. 0.00001 .AND.                    PRIN0940
     &    ABS(EIUV1/EM) .LE. 0.00001 ) THEN                           PRIN0950
          BETA(ISEG)=0                                                PRIN0960
      ELSE IF( ABS((EIV1-EIU1)/EIV1) .LE. 0.00001 .AND.               PRIN0970
     &    ABS(EIUV1/EM) .GT. 0.00001 ) THEN                           PRIN0980
          BETA(ISEG)=0.5*ATAN2(2.*EIUV1,0.)                           PRIN0990
      ELSE IF( ABS((EIV1-EIU1)/EIV1) .GT. 0.00001 .AND.               PRIN1000
     &    ABS(EIUV1/EM) .LE. 0.00001 ) THEN                           PRIN1010
          BETA(ISEG)=0                                                PRIN1020
      ELSE IF( ABS((EIV1-EIU1)/EIV1) .GT. 0.00001 .AND.               PRIN1030
     &    ABS(EIUV1/EM) .GT. 0.00001 ) THEN                           PRIN1040
          BETA(ISEG)=0.5*ATAN2(2.*EIUV1,(EIV1-EIU1))                  PRIN1050
      ENDIF                                                           PRIN1060
C                                                                     PRIN1070
C             ------------------------------------------              PRIN1080
C             --- LIMIT BETA TO (90 DEGREE TO -90 DEGREE) ---         PRIN1090
C             ------------------------------------------              PRIN1100
              P180=3.1415926                                          PRIN1110
              P90 =3.1415926/2.                                       PRIN1120
              IF( BETA(ISEG) .GT. P90.AND. BETA(ISEG) .LE. P180  ) THEN PRIN1130
                  BETA(ISEG)=BETA(ISEG)-P180                          PRIN1140
              ELSE IF(BETA(ISEG).GE.-P180.AND.BETA(ISEG) LT -P90 ) THEN PRIN1150
                  BETA(ISEG)=BETA(ISEG)+P180                          PRIN1160
              ENDIF                                                   PRIN1170
C                                                                     PRIN1180
C             ------------------------------------------              PRIN1190
C             --- AVOID UNSTABLE BETA ANGLE OCCURRED ---              PRIN1200
C             --- NEAR 90, -90, AND 0 DEGREE         ---              PRIN1210
C             ------------------------------------------              PRIN1220
              IF( ABS(ABS(BETA(ISEG))-P90) .LT. 0.03 ) THEN           PRIN1230
                  IF(BETA(ISEG) .LT  0 ) BETA(ISEG)= P90              PRIN1240
                  IF(BETA(ISEG) .GT  0 ) BETA(ISEG)=-P90              PRIN1250
              ELSE IF( ABS(BETA(ISEG)) .LT. 0.03 ) THEN               PRIN1260
                  BETA(ISEG)=0                                        PRIN1270
              ENDIF                                                   PRIN1280
C                                                                     PRIN1290
              IF( BETA(ISEG) .EQ. 0 ) THEN                            PRIN1300
                 EA=EA0                                               PRIN1310
                 EIU=EIU1                                             PRIN1320
                 EIV=EIV1                                             PRIN1330
              ELSE IF( BETA(ISEG) .NE. 0 ) THEN                       PRIN1340
                 EA=EA0                                               PRIN1350
                 EIU=0.5*(EIU1+EIV1)-(EIUV1/SIN(2.*BETA(ISEG)))       PRIN1360
                 EIV=0.5*(EIU1+EIV1)+(EIUV1/SIN(2.*BETA(ISEG)))       PRIN1370
              ENDIF                                                   PRIN1380
C                                                                     PRIN1390
C SECTION RIGIDITIES WITH RESPECT TO THE INSTANTANEOUS PRINCIPAL      PRIN1400
C AXES U AND V                                                        PRIN1410
C                                                                     PRIN1420
C                                                                     PRIN1430
      CALL SECRIG(LIBN,ST,TOTA,BJ,IREV1,IREV2,IREV3,IREV4,B,NUMP)     PRIN1440
      GKT=G*BJ                                                        PRIN1450
```

```
C                                                                       PRIN1460
C CALCULATE PRESENT SECTIONAL ELEMENT COORDINATE                        PRIN1470
C ACCORDING TO INSTANTANEOUS AXES                                       PRIN1480
C                                                                       PRIN1490
      DO 52 J=1,NUMP                                                    PRIN1500
      U(ISEG,J)= COS(BETA(ISEG))*(U0(J)-UC0)-SIN(BETA(ISEG))*(V0(J)-VC0)PRIN1510
      V(ISEG,J)=-SIN(BETA(ISEG))*(U0(J)-UC0)+COS(BETA(ISEG))*(V0(J)-VC0)PRIN1520
   52 CONTINUE                                                          PRIN1530
      RETURN                                                            PRIN1540
      END                                                               PRIN1550
C                                                                       JUDE0010
      SUBROUTINE JUDEL0(NSEG,NUMP,ISEG,EM,EY,RATIO3,                   JUDE0020
     &                  DE,E,E1,E2,TE,STR )                             JUDE0030
C     ------------------------------------------                        JUDE0040
C     ELASTO-PLASTIC STRESS-STRAIN HYSTERESIS MODEL                     JUDE0050
C     ------------------------------------------                        JUDE0060
      DIMENSION DE(NSEG,NUMP),E(NUMP),E1(NSEG,NUMP)                     JUDE0070
      DIMENSION E2(NSEG,NUMP),TE(NUMP),STR(NSEG,NUMP)                   JUDE0080
C                                                                       JUDE0090
      DO 75 I=1,NUMP                                                    JUDE0100
       IF ( E(I) .LT. E1(ISEG,I) .AND.E(I) .GT. E2(ISEG,I)) THEN        JUDE0110
          TE(I)=EM                                                      JUDE0120
          STR(ISEG,I)=STR(ISEG,I)-TE(I)*DE(ISEG,I)                      JUDE0130
       ENDIF                                                            JUDE0140
C                                                                       JUDE0150
C     -------------------------------                                   JUDE0160
C     --   TENSION STRAIN CONTROL   --                                  JUDE0170
C     -------------------------------                                   JUDE0180
       IF (E(I).GE.E1(ISEG,I) .AND  DE(ISEG,I) .GE. 0.) THEN            JUDE0190
          E1(ISEG,I)=E(I)                                               JUDE0200
          E2(ISEG,I)=E(I)-2.*EY                                         JUDE0210
          TE(I)=RATIO3*EM                                               JUDE0220
          STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)                      JUDE0230
       ENDIF                                                            JUDE0240
       IF (E(I) GE.E1(ISEG,I) .AND. DE(ISEG,I)  LT. 0.) THEN            JUDE0250
          TE(I)=EM                                                      JUDE0260
          STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)                      JUDE0270
       ENDIF                                                            JUDE0280
C                                                                       JUDE0290
C     -------------------------------                                   JUDE0300
C     -- COMPRESSION STRAIN CONTROL --                                  JUDE0310
C     -------------------------------                                   JUDE0320
       IF (E(I) .LE.E2(ISEG,I) .AND. DE(ISEG,I) .LE. 0.) THEN           JUDE0330
          E2(ISEG,I)=E(I)                                               JUDE0340
          E1(ISEG,I)=E(I)+2.*EY                                         JUDE0350
          TE(I)=RATIO3*EM                                               JUDE0360
          STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)                      JUDE0370
       ENDIF                                                            JUDE0380
       IF (E(I).LE.E2(ISEG,I) .AND. DE(ISEG,I) .GT. 0.) THEN            JUDE0390
          TE(I)=EM                                                      JUDE0400
          STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)                      JUDE0410
       ENDIF                                                            JUDE0420
   75 CONTINUE                                                          JUDE0430
C                                                                       JUDE0440
      RETURN                                                            JUDE0450
      END                                                               JUDE0460
C                                                                       JUDE0470
C                                                                       JUDE0480
      SUBROUTINE JUDEL2(NSEG,NUMP,ISEG,EM,EY,RATIO3,YS,                 JUDE0490
     &                  DE,E,E1,E2,TE,STR )                             JUDE0500
C     ------------------------------------------                        JUDE0510
C     MODIFIED BILINEAR STESS-STRAIN HYSTERESIS MODEL                   JUDE0520
C     ------------------------------------------                        JUDE0530
      DIMENSION DE(NSEG,NUMP),E(NUMP),E1(NSEG,NUMP)                     JUDE0540
      DIMENSION E2(NSEG,NUMP),TE(NUMP),STR(NSEG,NUMP)                   JUDE0550
C                                                                       JUDE0560
      DO 75 I=1,NUMP                                                    JUDE0570
      IF( E(I) .LT. E1(ISEG,I) .AND.E(I) .GT. E2(ISEG,I)) THEN          JUDE0580
          TE(I)=EM                                                      JUDE0590
          STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)                      JUDE0600
C                                                                       JUDE0610
C     -------------------------------                                   JUDE0620
C     --   TENSION STRAIN CONTROL   --                                  JUDE0630
C     -------------------------------                                   JUDE0640
      ELSE IF(E(I).GE.E1(ISEG,I) .AND. DE(ISEG,I)  GE. 0.) THEN         JUDE0650
        IF(ABS(STR(ISEG,I)) .GE. YS) THEN                               JUDE0660
          E1(ISEG,I)=E(I)                                               JUDE0670
          E2(ISEG,I)=E(I)-2 *EY                                         JUDE0680
          TE(I)=RATIO3*EM                                               JUDE0690
          STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)                      JUDE0700
        ELSE IF(ABS(STR(ISEG,I)) .LT. YS) THEN                          JUDE0710
          TE(I)=EM                                                      JUDE0720
          STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)                      JUDE0730
        ENDIF                                                           JUDE0740
      ELSE IF(E(I) GE.E1(ISEG,I) .AND. DE(ISEG,I) .LT. 0.) THEN         JUDE0750
          TE(I)=EM                                                      JUDE0760
          STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)                      JUDE0770
C                                                                       JUDE0780
C     -------------------------------                                   JUDE0790
C     -- COMPRESSION STRAIN CONTROL --                                  JUDE0800
C     -------------------------------                                   JUDE0810
      ELSE IF(E(I) .LE.E2(ISEG,I)  AND. DE(ISEG,I) .LE. 0.) THEN        JUDE0820
        IF(ABS(STR(ISEG,I)) .GE  YS) THEN                               JUDE0830
          E2(ISEG,I)=E(I)                                               JUDE0840
          E1(ISEG,I)=E(I)+2.*EY                                         JUDE0850
          TE(I)=RATIO3*EM                                               JUDE0860
          STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)                      JUDE0870
        ELSE IF(ABS(STR(ISEG,I)) .LT. YS) THEN                          JUDE0880
          TE(I)=EM                                                      JUDE0890
          STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)                      JUDE0900
        ENDIF                                                           JUDE0910
      ELSE IF(E(I) .LE.E2(ISEG,I) .AND. DE(ISEG,I) .GT. 0.) THEN        JUDE0920
          TE(I)=EM                                                      JUDE0930
          STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)                      JUDE0940
      ENDIF                                                             JUDE0950
   75 CONTINUE                                                          JUDE0960
C                                                                       JUDE0970
      RETURN                                                            JUDE0980
      END                                                               JUDE0990
C                                                                       JUDE1000
      SUBROUTINE JUDEL3(NSEG,NUMP,ISEG,EM,IR,YS,DE,E,TE,                JUDE1010
     &                  DEP,EOP,STRP,STR,TEP,EREP,SREP,MOPT)            JUDE1020
C     ------------------------------------------                        JUDE1030
C     RAMBERG-OSGOOD STRESS-STRAIN HYSTERESIS MODEL                     JUDE1040
C     ------------------------------------------                        JUDE1050
      DIMENSION DE(NSEG,NUMP),E(NUMP),TE(NUMP),STR(NSEG,NUMP)           JUDE1060
      DIMENSION DEP(NSEG,NUMP),EOP(NSEG,NUMP),STRP(NSEG,NUMP),         JUDE1070
     &          TEP(NSEG,NUMP),EREP(NSEG,NUMP),SREP(NSEG,NUMP)         JUDE1080
C                                                                       JUDE1090
C     ---------------------------                                       JUDE1100
C     -- FOR INITIAL CONDITION --                                       JUDE1110
C     ---------------------------                                       JUDE1120
C                                                                       JUDE1130
      DO 75 I=1,NUMP                                                    JUDE1140
      IF( MOPT .EQ. 2 ) THEN                                            JUDE1150
          TE(I)=EM                                                      JUDE1160
          STR(ISEG,I)=STR(ISEG,I)-TE(I)*DE(ISEG,I)                      JUDE1170
          TEP(ISEG,I)=TE(I)                                             JUDE1180
          DEP(ISEG,I)=DE(ISEG,I)                                        JUDE1190
          EOP(ISEG,I)=E(I)                                              JUDE1200
          STRP(ISEG,I)=STR(ISEG,I)                                      JUDE1210
C                                                                       JUDE1220
C     -------------------------                                         JUDE1230
C     -- CHECK REVERSE POINT --                                         JUDE1240
C     -------------------------                                         JUDE1250
      ELSE IF(DEP(ISEG,I)* DE(ISEG,I) .LT. 0 ) THEN                     JUDE1260
          EREP(ISEG,I)=EOP(ISEG,I)                                      JUDE1270
          SREP(ISEG,I)=STRP(ISEG,I)                                     JUDE1280
          TE(I)=EM                                                      JUDE1290
          STR(ISEG,I)=STR(ISEG,I)+TE(I)*DE(ISEG,I)                      JUDE1300
          TEP(ISEG,I)=TE(I)                                             JUDE1310
          DEP(ISEG,I)=DE(ISEG,I)                                        JUDE1320
          EOP(ISEG,I)=E(I)                                              JUDE1330
          STRP(ISEG,I)=STR(ISEG,I)                                      JUDE1340
C                                                                       JUDE1350
C     ----------------------------------                                JUDE1360
C     -- FOR SKELETON AND BRANCH REGION --                              JUDE1370
C     ----------------------------------                                JUDE1380
      ELSE IF(DEP(ISEG,I)* DE(ISEG,I)  GE. 0 ) THEN                     JUDE1390
          STR(ISEG,I)=STR(ISEG,I)-TEP(ISEG,I)*DE(ISEG,I)                JUDE1400
C                                                                       JUDE1410
C..... FOR FIRST SKELETON CURVE                                         JUDE1420
          IF( EREP(ISEG,I) .EQ. 0  AND SREP(ISEG,I) .EQ. 0 ) THEN       JUDE1430
              IF( ABS(STR(ISEG,I)/YS)  LT  0.2 ) THEN                   JUDE1440
                TEMP1=1/EM                                              JUDE1450
              ELSE IF( STR(ISEG,I)  GE. 0 ) THEN                        JUDE1460
                TEMP1=(1 + IR*ABS(STR(ISEG,I)/YS)**(IR-1) )/EM          JUDE1470
              ELSE IF( STR(ISEG,I) .LT  0 ) THEN                        JUDE1480
                TEMP1=(1 + IR*ABS(STR(ISEG,I)/YS)**(IR-1) )/EM          JUDE1490
              ENDIF                                                     JUDE1500
              TE(I)=1/TEMP1                                             JUDE1510
              TEP(ISEG,I)=TE(I)                                         JUDE1520
              DEP(ISEG,I)=DE(ISEG,I)                                    JUDE1530
              EOP(ISEG,I)=E(I)                                          JUDE1540
              STRP(ISEG,I)=STR(ISEG,I)                                  JUDE1550
C                                                                       JUDE1560
C..... AFTER FIRST SKELETON CURVE                                       JUDE1570
          ELSE IF( EREP(ISEG,I) .NE. 0 .OR. SREP(ISEG,I) .NE. 0 ) THEN  JUDE1580
              TEMP2= STR(ISEG,I) - SREP(ISEG,I)                         JUDE1590
              IF ( ABS(TEMP2/(2*YS))  LT. 0.2 ) THEN                    JUDE1600
                TEMP1=1/EM                                              JUDE1610
              ELSE IF( TEMP2  GE. 0 ) THEN                              JUDE1620
                TEMP1=(1 + IR*ABS(TEMP2/(2*YS))**(IR-1) )/EM            JUDE1630
              ELSE IF( TEMP2 .LT. 0 ) THEN                              JUDE1640
                TEMP1=(1 + IR*ABS(TEMP2/(2*YS))**(IR-1) )/EM            JUDE1650
              ENDIF                                                     JUDE1660
              TE(I)=1/TEMP1                                             JUDE1670
              TEP(ISEG,I)=TE(I)                                         JUDE1680
              DEP(ISEG,I)=DE(ISEG,I)                                    JUDE1690
              EOP(ISEG,I)=E(I)                                          JUDE1700
              STRP(ISEG,I)=STR(ISEG,I)                                  JUDE1710
          ENDIF                                                         JUDE1720
      ENDIF                                                             JUDE1730
   75 CONTINUE                                                          JUDE1740
C                                                                       JUDE1750
      RETURN                                                            JUDE1760
      END                                                               JUDE1770
C                                                                       JUDE1780
C-----------------------------------------------------------------      BOXL0010
      SUBROUTINE BOXLOC(PJL,STR,NSEG,NUMP,ISEG,INEB,INEH,               BOXL0020
     &                  TE,ST,EM,U0,V0,PLP )                            BOXL0030
C     ---------------------------------------                           BOXL0040
C     JUDGE LOCAL BUCKLING STRESS FOR BOX SECTION                       BOXL0050
C     ---------------------------------------                           BOXL0060
C     NOTE   PLP=0   : NO LOCAL BUCKLING OCCURRED                       BOXL0070
C            PLP=1-5 : LOCAL BUCKLING OCCURRED                          BOXL0080
C                                                                       BOXL0090
      REAL INEB,INEH                                                    BOXL0100
      DIMENSION STR(NSEG,NUMP),U0(NUMP),V0(NUMP),PJL(NUMP),TE(NUMP)     BOXL0110
C                                                                       BOXL0120
      DO 60 I=1,NUMP                                                    BOXL0130
   60   PJL(I)=0                                                        BOXL0140
      IPT1=1                                                            BOXL0150
      INEQ=INEB+INEH                                                    BOXL0160
      IPT2=INEQ+1                                                       BOXL0170
      IPT3=2*INEQ+1                                                     BOXL0180
      IPT4=3*INEQ+1                                                     BOXL0190
C                                                                       BOXL0200
C     CALCULATE SECTION ELEMENT LENGTH                                  BOXL0210
C                                                                       BOXL0220
      ELH=ABS(V0(2)-V0(3))                                              BOXL0230
      ELB=ABS(U0(INEH+1)-U0(INEH+2))                                    BOXL0240
C     ---------------------------------                                 BOXL0250
C     -- JUDGE LOCAL BUCKLING INDEX PJL --                              BOXL0260
C     ---------------------------------                                 BOXL0270
C                                                                       BOXL0280
C     AVOID SECTION REACH TO FULLY YIELD COMPRESSION STRESS             BOXL0290
C                                                                       BOXL0300
      IF( STR(ISEG,IPT1) .LT. 0 .AND. STR(ISEG,IPT2)  LT. 0 .AND.       BOXL0310
     &    STR(ISEG,IPT3) .LT. 0  AND. STR(ISEG,IPT4)  LT. 0 ) THEN      BOXL0320
          IF( TE(IPT1) .NE. EM .AND. TE(IPT2) .NE. EM  AND.             BOXL0330
     &        TE(IPT3) .NE. EM .AND. TE(IPT4) .NE. EM       ) THEN      BOXL0340
            DO 260 I=1,NUMP                                             BOXL0350
  260         PJL(I)=1                                                  BOXL0360
            PLP=5                                                       BOXL0370
            RETURN                                                      BOXL0380
          ENDIF                                                         BOXL0390
      ENDIF                                                             BOXL0400
C                                                                       BOXL0410
      IF( STR(ISEG,IPT1) .LT. 0 .AND. STR(ISEG,IPT3) .LT. 0 ) THEN      BOXL0420
        IF(TE(IPT1) .NE. EM .AND. TE(IPT3) .NE. EM) THEN                BOXL0430
          DO 70 I=INEH+1,INEQ                                           BOXL0440
   70       PJL(I)=1                                                   BOXL0450
          DO 72 I=2*INEQ+INEH+1, 3*INEQ                                 BOXL0460
   72       PJL(I)=1                                                   BOXL0470
C                                                                       BOXL0480
          L1=2                                                          BOXL0490
          L2=INEH                                                       BOXL0500
          L3=INEQ+2                                                     BOXL0510
          L4=INEQ+INEH                                                  BOXL0520
          CALL SUBLI1(1,PLP,L1,L2,L3,L4,ELH,EM,ST,STR,PJL,ISEG,         BOXL0530
     &                NSEG,NUMP)                                        BOXL0540
C                                                                       BOXL0550
          L1=2*INEQ+2                                                   BOXL0560
          L2=2*INEQ+INEH                                                BOXL0570
          L3=3*INEQ+2                                                   BOXL0580
          L4=3*INEQ+INEH                                                BOXL0590
          CALL SUBLI1(1,PLP,L1,L2,L3,L4,ELH,EM,ST,STR,PJL,ISEG,         BOXL0600
     &                NSEG,NUMP)                                        BOXL0610
        ENDIF                                                           BOXL0620
      ENDIF                                                             BOXL0630
C                                                                       BOXL0640
      IF( STR(ISEG,IPT2) .LT. 0 .AND. STR(ISEG,IPT4) .LT. 0 ) THEN      BOXL0650
        IF(TE(IPT2) .NE. EM .AND  TE(IPT4) .NE. EM) THEN                BOXL0660
          DO 40 I=INEQ+INEH+1,2*INEQ                                    BOXL0670
   40       PJL(I)=1                                                   BOXL0680
          DO 42 I=3*INEQ+INEH+1, 4*INEQ                                 BOXL0690
   42       PJL(I)=1                                                   BOXL0700
C                                                                       BOXL0710
          L1=2                                                          BOXL0720
          L2=INEH                                                       BOXL0730
          L3=INEQ+2                                                     BOXL0740
          L4=INEQ+INEH                                                  BOXL0750
          CALL SUBLI1(2,PLP,L1,L2,L3,L4,ELH,EM,ST,STR,PJL,ISEG,         BOXL0760
     &                NSEG,NUMP)                                        BOXL0770
C                                                                       BOXL0780
          L1=2*INEQ+2                                                   BOXL0790
          L2=2*INEQ+INEH                                                BOXL0800
          L3=3*INEQ+2                                                   BOXL0810
          L4=3*INEQ+INEH                                                BOXL0820
          CALL SUBLI1(2,PLP,L1,L2,L3,L4,ELH,EM,ST,STR,PJL,ISEG,         BOXL0830
     &                NSEG,NUMP)                                        BOXL0840
        ENDIF                                                           BOXL0850
      ENDIF                                                             BOXL0860
C                                                                       BOXL0870
      IF( STR(ISEG,IPT1) .LT. 0 .AND. STR(ISEG,IPT2)  LT  0 ) THEN      BOXL0880
        IF(TE(IPT1)  NE. EM .AND. TE(IPT2) .NE. EM) THEN                BOXL0890
          DO 20 I=1,INEH                                                BOXL0900
   20       PJL(I)=1                                                   BOXL0910
          DO 22 I=INEQ+1, INEQ+INEH                                     BOXL0920
   22       PJL(I)=1                                                   BOXL0930
C                                                                       BOXL0940
          L1=INEQ-INEH+1                                                BOXL0950
          L2=2*INEQ                                                     BOXL0960
          L3=3*INEQ+INEH+1                                              BOXL0970
          L4=4*INEQ                                                     BOXL0980
          CALL SUBLI1(3,PLP,L1,L2,L3,L4,ELH,EM,ST,STR,PJL,ISEG,         BOXL0990
     &                NSEG,NUMP)                                        BOXL1000
C                                                                       BOXL1010
          L1=INEH+1                                                     BOXL1020
          L2=INEQ                                                       BOXL1030
                                                                        BOXL1040
```

```
      L3=2*INEQ+INEH+1                                                  BOXL1050
      L4=3*INEQ                                                         BOXL1060
      CALL SUBLI1(3,FLP,L1,L2,L3,L4,ELH,EM,ST,STR,FJL,ISEG,             BOXL1070
     &                 NSEG,NUMP)                                       BOXL1080
      ENDIF                                                             BOXL1090
     ENDIF                                                              BOXL1100
C                                                                       BOXL1110
     IF( STR(ISEG,IPT3) .LT. 0 .AND. STR(ISEG,IPT4) .LT. 0 ) THEN       BOXL1120
       IF(TE(IPT3) .NE. EM .AND. TE(IPT4) .NE. EM) THEN                 BOXL1130
         DO 30 I=2*INEQ+1,2*INEQ+INEH                                   BOXL1140
 30        FJL(I)=1                                                    BOXL1150
         DO 32 I=3*INEQ+1, 3*INEQ+INEH                                  BOXL1160
 32        FJL(I)=1                                                    BOXL1170
C                                                                       BOXL1180
         L1=INEH+1                                                      BOXL1190
         L2=INEQ                                                        BOXL1200
         L3=2*INEQ+INEH+1                                               BOXL1210
         L4=3*INEQ                                                      BOXL1220
         CALL SUBLI1(4,FLP,L1,L2,L3,L4,ELH,EM,ST,STR,FJL,ISEG,          BOXL1230
     &                 NSEG,NUMP)                                       BOXL1240
C                                                                       BOXL1250
         L1=INEQ+INEH+1                                                 BOXL1260
         L2=2*INEQ                                                      BOXL1270
         L3=3*INEQ+INEH+1                                               BOXL1280
         L4=4*INEQ                                                      BOXL1290
         CALL SUBLI1(4,FLP,L1,L2,L3,L4,ELH,EM,ST,STR,FJL,ISEG,          BOXL1300
     &                 NSEG,NUMP)                                       BOXL1310
       ENDIF                                                            BOXL1320
     ENDIF                                                              BOXL1330
C                                                                       BOXL1340
     RETURN                                                             BOXL1350
     END                                                                BOXL1360
     SUBROUTINE SUBLI1(KK,FLP,L1,L2,L3,L4,ELH,EM,ST,STR,FJL,ISEG,       SUBL0010
     &                 NSEG,NUMP)                                       SUBL0020
     DIMENSION STR(NSEG,NUMP),FJL(NUMP)                                 SUBL0030
         W=0                                                            SUBL0040
         TOTST=0                                                        SUBL0050
         J=0                                                            SUBL0060
         DO 74 I=L1,L2                                                  SUBL0070
            IF(STR(ISEG,I) .LT. 0 ) THEN                                SUBL0080
               J=J+1                                                    SUBL0090
               W=W+ELH                                                  SUBL0100
               TOTST=TOTST+STR(ISEG,I)                                  SUBL0110
            ENDIF                                                       SUBL0120
 74      CONTINUE                                                       SUBL0130
         DO 76 I=L3,L4                                                  SUBL0140
            IF(STR(ISEG,I) .LT. 0 ) THEN                                SUBL0150
               J=J+1                                                    SUBL0160
               W=W+ELH                                                  SUBL0170
               TOTST=TOTST+STR(ISEG,I)                                  SUBL0180
            ENDIF                                                       SUBL0190
 76      CONTINUE                                                       SUBL0200
C                                                                       SUBL0210
C    FIND AVERAGE STRESS AND CRITICAL STRESS SCR                        SUBL0220
C                                                                       SUBL0230
         IF ( J .NE. 0 ) THEN                                           SUBL0240
          SAVE=ABS(TOTST/J)                                             SUBL0250
          SCR=0.425*(3.14159*3.14159)*EM/(12*0.91*(W/ST)**2)            SUBL0260
C                                                                       SUBL0270
          IF ( SAVE .GT. SCR) THEN                                      SUBL0280
           DO 78 I=L1,L2                                                SUBL0290
 78          IF( STR(ISEG,I) .LT. 0 .AND. FJL(I) .NE. 1 ) FJL(I)=1     SUBL0300
           DO 80 I=L3,L4                                                SUBL0310
 80          IF( STR(ISEG,I) .LT. 0 .AND. FJL(I) .NE. 1 ) FJL(I)=1     SUBL0320
           FLP=KK                                                       SUBL0330
          ENDIF                                                         SUBL0340
         ENDIF                                                          SUBL0350
C                                                                       SUBL0360
       RETURN                                                           SUBL0370
       END                                                              SUBL0380
C                                                                       SKYA0010
      SUBROUTINE SKY(ASAT,STIFF,ISTOR,NOP,MDIAG,IMD)                    SKYA0020
C-----------------------------------------------------------------------SKYA0030
C--   CONVERT STIFFNESS MATRIX TO HALF STORAGE AND BANDED MODE        --SKYA0040
C-----------------------------------------------------------------------SKYA0050
      DIMENSION ASAT(NOP,NOP),STIFF(ISTOR),MDIAG(NOP+1)                 SKYA0060
C                                                                       SKYA0070
C CONVERT STIFFNESS MATRIX TO HALF STORAGE BANDED MODE - STIFF          SKYA0080
C THE MAIN DIAGONAL ADDRESS ARE STORED IN MDIAG ARRAY  - MDIAG          SKYA0090
C                                                                       SKYA0100
      L=0                                                               SKYA0110
      DO 70 J=1,NOP                                                     SKYA0120
      DO 80 I=1,J                                                       SKYA0130
        IF ( ASAT(I,J) .EQ. 0 .AND. I .NE. J ) GO TO 80                 SKYA0140
        IF ( ASAT(I,J) .EQ. 0 .AND. I .EQ. J ) THEN                     SKYA0150
          WRITE(108,*) 'THE STIFFNESS ',I,'TH DIAGONAL TERM ARE ZERO'   SKYA0160
     &                 //' - SOLUTION ABORTED'                         SKYA0170
          RETURN                                                        SKYA0180
        ENDIF                                                           SKYA0190
        M=I                                                             SKYA0200
        GO TO 100                                                       SKYA0210
 80   CONTINUE                                                          SKYA0220
100   DO 70 I=J,M,-1                                                    SKYA0230
        L=L+1                                                           SKYA0240
        IF ( I.EQ. J) MDIAG(J)=L                                        SKYA0250
        IF ( ASAT(I,J) .EQ. 0 .AND. I .EQ. J ) THEN                     SKYA0260
          WRITE(108,*) 'THE STIFFNESS ',I,'TH DIAGONAL TERM ARE ZERO'   SKYA0270
     &                 //' - SOLUTION ABORTED'                         SKYA0280
          RETURN                                                        SKYA0290
        ENDIF                                                           SKYA0300
        STIFF(L)=ASAT(I,J)                                              SKYA0310
        IPT=L                                                           SKYA0320
 70   CONTINUE                                                          SKYA0330
      MDIAG(NOP+1)=IPT+1                                                SKYA0340
      IMD=MDIAG(NOP+1)-1                                                SKYA0350
C                                                                       SKYA0360
      RETURN                                                            SKYA0370
      END                                                               SKYA0380
C                                                                       SECR0010
      SUBROUTINE SECRIG(LIBN,ST,TOTA,BJ,L1,N1,L2,N2,B,NUMP)             SECR0020
C     ---------------------------------------------------               SECR0030
C     FIND SECTION RIGIDITY                                             SECR0040
C                                                                       SECR0050
C     NOTE:                                                             SECR0060
C         THE TORSIONAL RIGITY IS ASSUMED ELASTIC CURRENTLY             SECR0070
C     ---------------------------------------------------               SECR0080
      DIMENSION B(NUMP)                                                 SECR0090
C                                                                       SECR0100
      BJJ=0.                                                            SECR0110
      BJ=0.                                                             SECR0120
      IF (LIBN.LE.2) THEN                                               SECR0130
C                                                                       SECR0140
C FOR SECT1,2 ELEMENT                                                   SECR0150
C                                                                       SECR0160
       DO 48 I=1,NUMP                                                   SECR0170
        BJJ=BJJ-B(I)                                                    SECR0180
 48    CONTINUE                                                         SECR0190
       BJ=4.*TOTA*TOTA*ST/BJJ                                           SECR0200
      ELSE IF(LIBN.EQ.3) THEN                                           SECR0210
C                                                                       SECR0220
C FOR SECTION 3 ELEMENT                                                 SECR0230
C                                                                       SECR0240
       STT1=1.-(B(2)**4/(12*B(1)**4))                                   SECR0250
       BJ=(B(1)*B(2)**3)*(5.3333-(3.36*B(2)/(B(1))))*STT1)              SECR0260
      ELSE IF(LIBN.EQ.4 OR.LIBN EQ 5) THEN                              SECR0270
C                                                                       SECR0280
C FOR SECT 4 ELEMENT                                                    SECR0290
C                                                                       SECR0300
       BJ=B(1)                                                          SECR0310
      ELSE IF(LIBN.EQ.6) THEN                                           SECR0320
C                                                                       SECR0330
C FOR SECT6 ELEMENT                                                     SECR0340
C                                                                       SECR0350
       NNUMP=NUMP/2                                                     SECR0360
       DO 58 I=1,NNUMP                                                  SECR0370
        BJJ=BJJ+((B(I)*B(I+NNUMP))/2.)                                  SECR0380
 58    CONTINUE                                                         SECR0390
       BJ=4.*TOTA*TOTA*ST/BJJ                                           SECR0400
      ELSE IF(LIBN.EQ.7.OR.LIBN.EQ.8) THEN                              SECR0410
C                                                                       SECR0420
C FOR SECT7 AND SECT8 ELEMENT                                           SECR0430
C                                                                       SECR0440
       STT1=ST/N1                                                       SECR0450
       DO 50 I=1,L1                                                     SECR0460
        SST1=0.                                                         SECR0470
        DO 60 J=1,N1                                                    SECR0480
         K=I+(J-1)*L1                                                   SECR0490
         SST1=STT1+SST1                                                 SECR0500
 60     CONTINUE                                                        SECR0510
        BJ=BJ+((B(I)*SST1**3)/3 )                                       SECR0520
 50    CONTINUE                                                         SECR0530
       STT2=ST/N2                                                       SECR0540
       DO 70 I=(L1*N1)+1,(L1*N1)+L2                                     SECR0550
        SST2=0.                                                         SECR0560
        DO 80 J=1,N2                                                    SECR0570
         K=I-(J-1)*L2                                                   SECR0580
         SST2=STT2+SST2                                                 SECR0590
 80     CONTINUE                                                        SECR0600
        BJ=BJ+((B(I)*SST2**3)/3 )                                       SECR0610
 70    CONTINUE                                                         SECR0620
      ENDIF                                                             SECR0630
C                                                                       SECR0640
      RETURN                                                            SECR0650
      END                                                               SECR0660
C                                                                       ECCE0010
      SUBROUTINE ECCENT(NSEG,UC0,VC0,ISEG,ROZA,ROZB,ECX,ECY,ECCX,ECCY)  ECCE0020
C     ----------------------------------                                ECCE0030
C     CALCULATE MEMBER END ECCENTRICITIES                               ECCE0040
C     ----------------------------------                                ECCE0050
      DIMENSION ECX(2),ECY(2),ECCX(2),ECCY(2)                           ECCE0060
C                                                                       ECCE0070
C ECCENTRICITIES ABOUT SECTION REFERENCE COORDINATES                    ECCE0080
C                                                                       ECCE0090
      IF( ISEG .EQ. 1) THEN                                             ECCE0100
         EXX=UC0-ECX(1)                                                 ECCE0110
         EYY=VC0-ECY(1)                                                 ECCE0120
      ELSE IF( ISEG .EQ. NSEG) THEN                                     ECCE0130
         EXX=UC0-ECX(2)                                                 ECCE0140
         EYY=VC0-ECY(2)                                                 ECCE0150
      ENDIF                                                             ECCE0160
C NOTE: EXX AND EYY ARE VECTORS FROM LOAD APPLIED POINT TO CENTROID     ECCE0170
C       IN SECTION REFERENCE COORDINATES DIRECTION                      ECCE0180
C       ECX IS THE DISTANCE FROM REFERENCE COORDINATE ORIGIN TO         ECCE0190
C       LOAD APPLIED LOCATION                                           ECCE0200
C                                                                       ECCE0210
C ECCENTRICITIES ABOUT MEMBER COORDINATES                               ECCE0220
C                                                                       ECCE0230
      IF( ISEG .EQ. 1 ) THEN                                            ECCE0240
        ECCX(1)=COS(ROZA)*EXX - SIN(ROZA)*EYY                           ECCE0250
        ECCY(1)=SIN(ROZA)*EXX + COS(ROZA)*EYY                           ECCE0260
      ELSE IF( ISEG .EQ. NSEG ) THEN                                    ECCE0270
        ECCX(2)=COS(ROZB)*EXX - SIN(ROZB)*EYY                           ECCE0280
        ECCY(2)=SIN(ROZB)*EXX + COS(ROZB)*EYY                           ECCE0290
      ENDIF                                                             ECCE0300
C                                                                       ECCE0310
C NOTE. ECCX AND ECCY ARE VECTORS FROM LOAD APPLIED POINT TO CENTROID   ECCE0320
C       IN MEMBER COORDINATES DIRECTION                                 ECCE0330
C                                                                       ECCE0340
      RETURN                                                            ECCE0350
      END                                                               ECCE0360
C                                                                       ELES0010
      SUBROUTINE ELESTI(NSEG,ISEG,ISTIF,EA,EIU,EIV,GKT,EL,PLOC,PSK,SK)  ELES0020
C     ----------------------------------                                ELES0030
C     CALCULATE STABILITY ELEMENT STIFFNESS                             ELES0040
C     ----------------------------------                                ELES0050
      DIMENSION EL(NSEG),PLOC(NSEG,12),PSK(NSEG,12,12),SK(12,12)        ELES0060
C                                                                       ELES0070
C INITIALIZE THE SEGMENT STIFFNESS                                      ELES0080
C                                                                       ELES0090
      AUN=1200.                                                         ELES0100
      DO 430 I=1,12                                                     ELES0110
      DO 430 J=1,12                                                     ELES0120
        SK(I,J)=0.                                                      ELES0130
  430 CONTINUE                                                          ELES0140
C                                                                       ELES0150
C FORMULATE THE ELEMENT STIFFNESS MATRIX                                ELES0160
C                                                                       ELES0170
C                                                                       ELES0180
      P=PLOC(ISEG,3)                                                    ELES0190
      COF=P/(10.*EL(ISEG))                                              ELES0200
      IF (ISTIF .EQ. 1) THEN                                            ELES0210
       SK(1,1)=12.*EIV/(EL(ISEG)**3)-COF*12.                            ELES0220
       SK(2,2)=12.*EIU/(EL(ISEG)**3)-COF*12.                            ELES0230
       SK(3,3)=EA/EL(ISEG)                                              ELES0240
       SK(4,4)=4.*EIU/EL(ISEG)-COF*(4.*EL(ISEG)*EL(ISEG)/3.)            ELES0250
       SK(5,5)=4.*EIV/EL(ISEG)-COF*(4.*EL(ISEG)*EL(ISEG)/3.)            ELES0260
       SK(6,6)=GKT/EL(ISEG)                                             ELES0270
       SK(1,5)=6.*EIV/(EL(ISEG)**2)-COF*EL(ISEG)                        ELES0280
       SK(2,4)=-6.*EIU/(EL(ISEG)**2)-COF*(-EL(ISEG))                    ELES0290
       SK(7,7)=SK(1,1)                                                  ELES0300
       SK(8,8)=SK(2,2)                                                  ELES0310
       SK(9,9)=SK(3,3)                                                  ELES0320
       SK(10,10)=SK(4,4)                                                ELES0330
       SK(11,11)=SK(5,5)                                                ELES0340
       SK(12,12)=SK(6,6)                                                ELES0350
       SK(7,11)=-SK(1,5)                                                ELES0360
       SK(8,10)=-SK(2,4)                                                ELES0370
       SK(1,7)=-SK(1,1)                                                 ELES0380
       SK(2,8)=-SK(2,2)                                                 ELES0390
       SK(3,9)=-SK(3,3)                                                 ELES0400
       SK(4,10)=0.5*SK(4,4)-COF*(-EL(ISEG)*EL(ISEG)/3.)                 ELES0410
       SK(5,11)=0.5*SK(5,5)-COF*(-EL(ISEG)*EL(ISEG)/3.)                 ELES0420
       SK(6,12)=-SK(6,6)                                                ELES0430
       SK(1,11)=SK(1,5)                                                 ELES0440
       SK(2,10)=SK(2,4)                                                 ELES0450
       SK(4,8)=-SK(2,10)                                                ELES0460
       SK(5,7)=-SK(1,11)                                                ELES0470
      ELSE IF(ISTIF .EQ. 0) THEN                                        ELES0480
       IF ( ABS(P) .LT. 1.) THEN                                        ELES0490
        SK(1,1)=12.*EIV/(EL(ISEG)**3)-COF*12.                           ELES0500
        SK(2,2)=12.*EIU/(EL(ISEG)**3) COF*12.                           ELES0510
        SK(3,3)=EA/EL(ISEG)                                             ELES0520
        SK(4,4)=4.*EIU/EL(ISEG)-COF*(4.*EL(ISEG)*EL(ISEG)/3.)           ELES0530
        SK(5,5)=4.*EIV/EL(ISEG) COF*(4.*EL(ISEG)*EL(ISEG)/3.)           ELES0540
        SK(6,6)=GKT/EL(ISEG)                                            ELES0550
        SK(1,5)=6.*EIV/(EL(ISEG)**2)-COF*EL(ISEG)                       ELES0560
        SK(2,4)=-6.*EIU/(EL(ISEG)**2)-COF*(-EL(ISEG))                   ELES0570
        SK(7,7)=SK(1,1)                                                 ELES0580
        SK(8,8)=SK(2,2)                                                 ELES0590
        SK(9,9)=SK(3,3)                                                 ELES0600
        SK(10,10)=SK(4,4)                                               ELES0610
        SK(11,11)=SK(5,5)                                               ELES0620
        SK(12,12)=SK(6,6)                                               ELES0630
        SK(7,11)=-SK(1,5)                                               ELES0640
        SK(8,10)=-SK(2,4)                                               ELES0650
        SK(1,7)=-SK(1,1)                                                ELES0660
        SK(2,8)=-SK(2,2)                                                ELES0670
        SK(3,9)=-SK(3,3)                                                ELES0680
        SK(4,10)=0.5*SK(4,4)-COF*(-EL(ISEG)*EL(ISEG)/3.)                ELES0690
        SK(5,11)=0.5*SK(5,5)-COF*(-EL(ISEG)*EL(ISEG)/3.)                ELES0700
        SK(6,12)=-SK(6,6)                                               ELES0710
        SK(1,11)=SK(1,5)                                                ELES0720
        SK(2,10)=SK(2,4)                                                ELES0730
        SK(4,8)=-SK(2,10)                                               ELES0740
        SK(5,7)=-SK(1,11)                                               ELES0750
       ELSE IF (P .GT. 1.) THEN                                         ELES0760
        SKU=SQRT(P/EIU)                                                 ELES0770
        CU=COS(SKU*EL(ISEG))*AUN                                        ELES0780
        SU=SIN(SKU*EL(ISEG))*AUN                                        ELES0790
        SKV=SQRT(P/EIV)                                                 ELES0800
        CV=COS(SKV*EL(ISEG))*AUN                                        ELES0810
        SV=SIN(SKV*EL(ISEG))*AUN                                        ELES0820
```

```
      SK(1,1)=EIV*((SKV**3)*SV/(2.*AUN-2.*CV-SKV*EL(ISEG)*SV))                  ELES0830
      SK(2,2)=EIU*((SKU**3)*SU/(2.*AUN-2.*CU-SKU*EL(ISEG)*SU))                  ELES0840
      SK(3,3)=EA/EL(ISEG)                                                       ELES0850
      SK(4,4)=EIU*SKU*((SU-SKU*EL(ISEG)*CU)/                                    ELES0860
     *           (2.*AUN-2.*CU-SKU*EL(ISEG)*SU))                               ELES0870
      SK(5,5)=EIV*SKV*((SV-SKV*EL(ISEG)*CV)/                                    ELES0880
     *           (2.*AUN-2.*CV-SKV*EL(ISEG)*SV))                               ELES0890
      SK(6,6)=GKT/EL(ISEG)                                                      ELES0900
      SK(1,5)=EIV*((SKV**2)*(1.*AUN-CV)/                                        ELES0910
     *           (2.*AUN-2.*CV-SKV*EL(ISEG)*SV))                               ELES0920
      SK(2,4)=-EIU*((SKU**2)*(1.*AUN-CU)/                                       ELES0930
     *           (2.*AUN-2.*CU-SKU*EL(ISEG)*SU))                               ELES0940
      SK(7,7)=SK(1,1)                                                           ELES0950
      SK(8,8)=SK(2,2)                                                           ELES0960
      SK(9,9)=SK(3,3)                                                           ELES0970
      SK(10,10)=SK(4,4)                                                         ELES0980
      SK(11,11)=SK(5,5)                                                         ELES0990
      SK(12,12)=SK(6,6)                                                         ELES1000
      SK(7,11)=-SK(1,5)                                                         ELES1010
      SK(8,10)=-SK(2,4)                                                         ELES1020
      SK(1,7)=-SK(1,1)                                                          ELES1030
      SK(2,8)=-SK(2,2)                                                          ELES1040
      SK(3,9)=-SK(3,3)                                                          ELES1050
      SK(4,10)=EIU*SKU*((SKU*EL(ISEG)*AUN-SU)/                                  ELES1060
     *           (2.*AUN-2.*CU-SKU*EL(ISEG)*SU))                               ELES1070
      SK(5,11)=EIV*SKV*((SKV*EL(ISEG)*AUN-SV)/                                  ELES1080
     *           (2.*AUN-2.*CV-SKV*EL(ISEG)*SV))                               ELES1090
      SK(6,12)=-SK(6,6)                                                         ELES1100
      SK(1,11)=SK(1,5)                                                          ELES1110
      SK(2,10)=SK(2,4)                                                          ELES1120
      SK(4,8)=-SK(2,10)                                                         ELES1130
      SK(5,7)=-SK(1,11)                                                         ELES1140
     ELSE IF(P .LT. -1.) THEN                                                   ELES1150
      SKU=SQRT(-P/EIU)                                                          ELES1160
      CU=COSH(SKU*EL(ISEG))*AUN                                                 ELES1170
      SU=SINH(SKU*EL(ISEG))*AUN                                                 ELES1180
      SKV=SQRT(-P/EIV)                                                          ELES1190
      CV=COSH(SKV*EL(ISEG))*AUN                                                 ELES1200
      SV=SINH(SKV*EL(ISEG))*AUN                                                 ELES1210
      SK(1,1)=EIV*((SKV**3)*SV/(2.*AUN-2.*CV+SKV*EL(ISEG)*SV))                  ELES1220
      SK(2,2)=EIU*((SKU**3)*SU/(2.*AUN-2.*CU+SKU*EL(ISEG)*SU))                  ELES1230
      SK(3,3)=EA/EL(ISEG)                                                       ELES1240
      SK(4,4)=EIU*SKU*((SKU*EL(ISEG)*CU-SU)/                                    ELES1250
     *           (2.*AUN-2.*CU-SKU*EL(ISEG)*SU))                               ELES1260
      SK(5,5)=EIV*SKV*((SKV*EL(ISEG)*CV-SV)/                                    ELES1270
     *           (2.*AUN-2.*CV+SKV*EL(ISEG)*SV))                               ELES1280
      SK(6,6)=GKT/EL(ISEG)                                                      ELES1290
      SK(1,5)=EIV*((SKV**2)*(CV-1.*AUN)/                                        ELES1300
     *           (2.*AUN-2.*CV+SKV*EL(ISEG)*SV))                               ELES1310
      SK(2,4)=-EIU*((SKU**2)*(CU-1.*AUN)/                                       ELES1320
     *           (2.*AUN-2.*CU+SKU*EL(ISEG)*SU))                               ELES1330
      SK(7,7)=SK(1,1)                                                           ELES1340
      SK(8,8)=SK(2,2)                                                           ELES1350
      SK(9,9)=SK(3,3)                                                           ELES1360
      SK(10,10)=SK(4,4)                                                         ELES1370
      SK(11,11)=SK(5,5)                                                         ELES1380
      SK(12,12)=SK(6,6)                                                         ELES1390
      SK(7,11)=-SK(1,5)                                                         ELES1400
      SK(8,10)=-SK(2,4)                                                         ELES1410
      SK(1,7)=-SK(1,1)                                                          ELES1420
      SK(2,8)=-SK(2,2)                                                          ELES1430
      SK(3,9)=-SK(3,3)                                                          ELES1440
      SK(4,10)=EIU*SKU*((SU-SKU*EL(ISEG)*AUN)/                                  ELES1450
     *           (2.*AUN-2.*CU+SKU*EL(ISEG)*SU))                               ELES1460
      SK(5,11)=EIV*SKV*((SV-SKV*EL(ISEG)*AUN)/                                  ELES1470
     *           (2.*AUN-2.*CV+SKV*EL(ISEG)*SV))                               ELES1480
      SK(6,12)=-SK(6,6)                                                         ELES1490
      SK(1,11)=SK(1,5)                                                          ELES1500
      SK(2,10)=SK(2,4)                                                          ELES1510
      SK(4,8)=-SK(2,10)                                                         ELES1520
      SK(5,7)=-SK(1,11)                                                         ELES1530
     ENDIF                                                                      ELES1540
     ENDIF                                                                      ELES1550
     DO 90 I=1,12                                                               ELES1560
       II=I+1                                                                   ELES1570
       DO 90 J=II,12                                                            ELES1580
         SK(J,I)=SK(I,J)                                                        ELES1590
  90 CONTINUE                                                                   ELES1600
C                                                                               ELES1610
C MEMORIZE THE ELEMENT STIFFNESS FOR BEING USED IN SUBROUTINE                   ELES1620
C CLDAP IN THE NEXT STEP                                                        ELES1630
C                                                                               ELES1640
      DO 190 II=1,12                                                            ELES1650
      DO 190 JJ=1,12                                                            ELES1660
        PSK(ISEG,II,JJ)=SK(II,JJ)                                               ELES1670
  190 CONTINUE                                                                  ELES1680
      RETURN                                                                    ELES1690
      END                                                                       ELES1700
C                                                                               TRAN0010
      SUBROUTINE TRANK(FFASAT,ECCX,ECCY,TPJT,ASATP)                             TRAN0020
C     ------------------------------------------------------------              TRAN0030
C     TRANSFER MEMBER STIFFNESS MATRIX INTO ECCENTRICITY LOAD                   TRAN0040
C     APPLY DIRECTION                                                           TRAN0050
C     ------------------------------------------------------------              TRAN0060
      DIMENSION ECCX(2),ECCY(2),TPJT(12,12),FFASAT(12,12),ASATP(12,12)          TRAN0070
C                                                                               TRAN0080
      DO 250 I=1,12                                                             TRAN0090
      DO 250 J=1,12                                                             TRAN0100
        TPJT(I,J)=0.                                                            TRAN0110
        IF( I  EQ. J ) TPJT(I,J)=1.                                             TRAN0120
  250 CONTINUE                                                                  TRAN0130
      TPJT(3, 4)= ECCY(1)                                                       TRAN0140
      TPJT(3, 5)=-ECCX(1)                                                       TRAN0150
      TPJT(1, 6)=-ECCY(1)                                                       TRAN0160
      TPJT(2, 6)= ECCX(1)                                                       TRAN0170
      TPJT(9,10)= ECCY(2)                                                       TRAN0180
      TPJT(9,11)=-ECCX(2)                                                       TRAN0190
      TPJT(7,12)=-ECCY(2)                                                       TRAN0200
      TPJT(8,12)= ECCX(2)                                                       TRAN0210
C                                                                               TRAN0220
C !EASAT!*!TPJT!                                                                TRAN0230
C                                                                               TRAN0240
      CALL MULTM(0,12,ASATP,FFASAT,TPJT)                                        TRAN0250
C                                                                               TRAN0260
C!TPJ!*!EASAT!*!TPJT!                                                           TRAN0270
C                                                                               TRAN0280
      CALL MULTM(1,12,FFASAT,TPJT,ASATP)                                        TRAN0290
      RETURN                                                                    TRAN0300
      END                                                                       TRAN0310
C-------------------------------------------------------------------------------SKYL0010
      SUBROUTINE SKYLIN(JFLAG,ICPT,NDOF,IBUG,TITLE,MDIAG,                       SKYL0020
     &                  IELDOF,LM,SE.MD,NAME)                                   SKYL0030
      DIMENSION MDIAG(NDOF+1),SE(IELDOF*(IELDOF+1)/2)                           SKYL0040
      DIMENSION MD(IELDOF+1),LM(IELDOF)                                         SKYL0050
      LOGICAL BTEST,PT                                                          SKYL0060
      CHARACTER*(*) NAME,TITLE(2)                                               SKYL0070
      CHARACTER*1 CX(100)                                                       SKYL0080
C                                                                               SKYL0090
      IF (IOPT.EQ.1) THEN                                                       SKYL0100
         DO 10 I=1,NDOF+1                                                       SKYL0110
   10       MDIAG(I)=I                                                         SKYL0120
         IF ( BTEST(IBUG,12) ) THEN                                             SKYL0130
           WRITE(108,*) 'INITIAL VALUES'                                        SKYL0140
           WRITE(108,320) (I,MDIAG(I),I=1,NDOF)                                 SKYL0150
         ENDIF                                                                  SKYL0160
C                                                                               SKYL0170
      ELSE IF (IOPT.EQ.2) THEN                                                  SKYL0180
C.......... DETERMINE THE SKYLINE OF THIS ELEMENT, AND MODIFY THE               SKYL0190
C             GLOBAL SKYLINE...                                                 SKYL0200
C....... I    - THE ROW OF THE MEMBER STIFFNESS MATRIX                          SKYL0210
C....... J    - THE COLUMN OF THE MEMBER STIFFNESS MATRIX                       SKYL0220
C..... .. IJ   - THE ADDRESS OF THE MEMBER STIFFNESS TERM I,J                   SKYL0230
C    . .. LMI  - THE ROW OF THE GLOBAL STIFFNESS MATRIX CORRESPONDING           SKYL0240
C                TO ROW I OF THE MEMBER STIFFNESS MATRIX                        SKYL0250
C.  .  . LMJ  - THE COLUMN OF THE GLOBAL STIFFNESS MATRIX CORRESPONDING         SKYL0260
C                TO COLUMN J OF THE MEMBER STIFFNESS MATRIX                     SKYL0270
C....  . MDIAG(JJ) = THE JJ'TH ELEMENT OF THE GLOBAL SKYLINE MATRIX.           SKYL0280
C              AT THIS POINT THE GLOBAL SKYLINE MATRIX CONTAINS THE             SKYL0290
C              LOWEST ROW NUMBER THAT CONTAINS A NON-ZERO STIFFNESS             SKYL0300
C              TERM.                                                            SKYL0310
C                                                                               SKYL0320
         DO 250 I=1,IELDOF                                                      SKYL0330
           LMI=LM(I)                                                            SKYL0340
           IF (LMI.LE.0) GO TO 250                                              SKYL0350
           DO 240 J=1,IELDOF                                                    SKYL0360
             LMJ=LM(J)                                                          SKYL0370
             IF (LMJ.LE.0) GO TO 240                                            SKYL0380
               IJ=MD(J)+J-I                                                     SKYL0390
               IF (SE(IJ) EQ.0  AND  JFLAG .EQ. 0 ) GO TO 240                   SKYL0400
C              IF (SE(IJ) EQ.0  ) GO TO 240                                     SKYL0410
               IF (LMI .LE.LMJ) THEN                                            SKYL0420
                   MDIAG(LMJ)=MIN(MDIAG(LMJ),LMI)                               SKYL0430
               ELSE                                                             SKYL0440
                   MDIAG(LMI)=MIN(MDIAG(LMI),LMJ)                               SKYL0450
               ENDIF                                                            SKYL0460
  240      CONTINUE                                                             SKYL0470
  250    CONTINUE                                                               SKYL0480
C-------------------------------------------------------------------------------SKYL0490
C                                                                               SKYL0500
      ELSE IF (IOPT.EQ.3) THEN                                                  SKYL0510
C.......... CALCULATE THE INDICIES OF THE MAIN DIAGONAL TERMS.                  SKYL0520
      IF ( BTEST(IBUG,12) ) THEN                                                SKYL0530
         WRITE(108,*) 'AFTER COLUMN HEIGHTS'                                    SKYL0540
         WRITE(108,320) (I,MDIAG(I),I=1,NDOF)                                   SKYL0550
      ENDIF                                                                     SKYL0560
      MDIAG(1)=1                                                                SKYL0570
      ICOLHT=1                                                                  SKYL0580
      DO 310 IROW=2,(NDOF+1)                                                    SKYL0590
         ILOCN=ICOLHT + MDIAG(IROW-1)                                           SKYL0600
         ICOLHT=IROW - (MDIAG(IROW)-1)                                          SKYL0610
  310    MDIAG(IROW)=ILOCN                                                      SKYL0620
      IF ( BTEST(IBUG,12) ) THEN                                                SKYL0630
         WRITE(108,*) 'ADDRESS OF MAIN DIAGONAL ELEMENTS'                       SKYL0640
         WRITE(108,320) (I,MDIAG(I),I=1,NDOF)                                   SKYL0650
  320    FORMAT (' MDIAG(',I3,')=',I5)                                          SKYL0660
      ENDIF                                                                     SKYL0670
  321 FORMAT (2I6)                                                              SKYL0680
C                                                                               SKYL0690
      IF (.NOT.(BTEST(IBUG,12).OR.BTEST(IBUG,9)) ) GOTO 370                     SKYL0700
C.......... PRINT A MAP OF THE STIFFNESS ARRAY                                  SKYL0710
      NL= MDIAG(NDOF+1)-MDIAG(1) +1                                             SKYL0720
      NP=NDOF/100                                                               SKYL0730
      NR=MOD(NDOF,100)                                                          SKYL0740
      IF (NR.GT.0) NP=NP+1                                                      SKYL0750
      DO 370 IP=1,NP                                                            SKYL0760
         JC1=(IP-1)*100                                                         SKYL0770
         JC2=MIN(IP*100,NDOF)                                                   SKYL0780
         J2=100                                                                 SKYL0790
         IF (IP.EQ.NP) J2=NR                                                    SKYL0800
         WRITE(108,290) TITLE(1),TITLE(2),NAME,NL,(I,I=JC1,JC2,10)              SKYL0810
         DO 270 I=1,NDOF                                                        SKYL0820
            PT=.FALSE.                                                          SKYL0830
            DO 260 JJ=1,J2                                                      SKYL0840
              J=JJ-(IP-1)*100                                                   SKYL0850
              IJ=MDIAG(J)+J-I                                                   SKYL0860
              IF (IJ.LT.MDIAG(J-1) .AND. I.LT.J) THEN                           SKYL0870
                   CX(JJ)='X'                                                   SKYL0880
                   PT=.TRUE.                                                    SKYL0890
              ELSE IF (I.EQ.J) THEN                                             SKYL0900
                   CX(JJ)='D'                                                   SKYL0910
                   PT=.TRUE.                                                    SKYL0920
              ELSE                                                              SKYL0930
                   CX(JJ)=' '                                                   SKYL0940
              ENDIF                                                             SKYL0950
  260       CONTINUE                                                            SKYL0960
  270    IF (PT) WRITE(108,295) I,(CX(J),J=1,J2)                                SKYL0970
C                                                                               SKYL0980
  290 FORMAT ('1 STRUCTURE..... ',A,/                                           SKYL0990
     &        '  SOLUTION..... ',A,//                                          SKYL1000
     &        '   SKYLINE OF THE ',A,' MATRIX ',//                             SKYL1010
     &  '  THE MATRIX REQUIRES',I6,' STORAGE LOCATIONS',                        SKYL1020
     &  /2X,12(I4,'.....'))                                                     SKYL1030
  295 FORMAT (I6,132A1)                                                         SKYL1040
C                                                                               SKYL1050
      ENDIF                                                                     SKYL1060
  370 RETURN                                                                    SKYL1070
      END                                                                       SKYL1080
C                                                                               NODD0010
C-------------------------------------------------------------------------------NODD0020
C                                                                               NODD0030
      SUBROUTINE NODDOF(NDOF,NCOND,NFREE,NREST,MAXNOD,NNODE,NCOS,               NODD0040
     &       SCALE,NSUPT,NCONST,ID,IDOF,INDEX,COORD,COSINE,CONST,               NODD0050
     &       JTPLG,JTCOS,BUG,MSTRJT)                                            NODD0060
C                                                                               NODD0070
      DIMENSION COORD(MAXNOD,3),ID(MAXNOD),INDEX(MAXNOD),IFM(6),JFM(6),         NODD0080
     &          IDOF(MAXNOD,6) ,COSINE(3,3,NCOS),CONST(MAXNOD,6),               NODD0090
     &          JTPLG(MAXNOD),JTCOS(MAXNOD),MSTRJT(MAXNOD,6)                    NODD0100
      DIMENSION VX(3),VY(3),VZ(3)                                               NODD0110
      LOGICAL FLAG,PT,BUG,TEST,BTEST                                            NODD0120
      CHARACTER*15 TYPE                                                         NODD0130
      CHARACTER*2  CX(6)                                                        NODD0140
C-------------------------------------------------------------------------------NODD0150
C VARIABLE TABLE.                                                               NODD0160
C COORD  - JOINT COORDINATES  X,Y,Z                                             NODD0170
C ID     - JOINT NUMBER                                                         NODD0180
C IDOF   - GLOBAL DEGREES OF FREEDOM ASSOC. WITH JOINT FX,FY,FZ,MX,MY,MZ       NODD0190
C JTCOS  - ROW IN COSINE MATRIX ASSOC WITH JOINT                                NODD0200
C COSINE - COSINE MATRIX                                                        NODD0210
C CONST  - GLOBAL CONSTRAINT MATRIX FOR EACH NODE                               NODD0220
C JTPLG  - CONTAINS JOINT FLAG'S --  MUST BE 24 BIT'S LONG .... ....            NODD0230
C          BIT'S  0 TO  5 RESTRAIN   DOF FX,FY,FZ,MX,MY,MZ                      NODD0240
C          BIT'S  6 TO 11 CONDENSE   DOF FX,FY,FZ,MX,MY,MZ                      NODD0250
C          BIT'S 12 TO 17 CONSTRAIN  DOF FX,FY,FZ,MX,MY,MZ                      NODD0260
C          BIT'S 18 TO 23 ELIMINATE  DOF FX,FY,FZ,MX,MY,MZ                      NODD0270
C MSTRJT - INTERNAL MASTER JT # OF CONSTRAINED JOINTS. .....                    NODD0280
C-------------------------------------------------------------------------------NODD0290
C                                                                               NODD0300
C..... - INPUT NODES                                                            NODD0310
      IF (BUG) WRITE(108,1000) 'INPUT NODES'                                    NODD0320
      I=0                                                                       NODD0330
   20 I=I+1                                                                     NODD0340
      READ(105, *) ID(I),(COORD(I,J),J=1,3),JTCOS(I),IGEN                       NODD0350
      IF (ID(I).LE.0) THEN                                                      NODD0360
         NNODE=I-1                                                              NODD0370
         GO TO 26                                                               NODD0380
      ENDIF                                                                     NODD0390
      IF (BUG) WRITE(108,30) I,ID(I),(COORD(I,J),J=1,3),JTCOS(I)                NODD0400
      IF (IGEN.GT.0) THEN                                                       NODD0410
           READ(105, *) IDINC,COORD1,COORD2,COORD3                              NODD0420
           DO 25 K=1,IGEN                                                       NODD0430
             I=I+1                                                              NODD0440
             ID(I)=ID(I-1)+IDINC                                                NODD0450
             IF (ID(I) LE.0) THEN                                               NODD0460
                NNODE=I-1                                                       NODD0470
                GO TO 26                                                        NODD0480
             ENDIF                                                              NODD0490
             COORD(I,1)=COORD(I-1,1)+COORD1                                     NODD0500
             COORD(I,2)=COORD(I-1,2)+COORD2                                     NODD0510
             COORD(I,3)=COORD(I-1,3)+COORD3                                     NODD0520
             JTCOS(I)=JTCOS(I-1)                                                NODD0530
   25        IF (BUG) WRITE(108,30) I,ID(I),(COORD(I,J),J=1,3),JTCOS(I)         NODD0540
      ENDIF                                                                     NODD0550
      IF (I.LT.NNODE) GO TO 20                                                  NODD0560
   26 CONTINUE                                                                  NODD0570
C                                                                               NODD0580
C....... SORT NODES                                                             NODD0590
      CALL ISORT(ID,INDEX,NNODE)                                                NODD0600
C                                                                               NODD0610
C        MOVE NODE INFO. TO TEMPORARY STORGAE                                   NODD0620
      DO 40 I=1,NNODE                                                           NODD0630
       J=INDEX(I)                                                               NODD0640
       IDOF(I,1)=ID(J)                                                          NODD0650
```

```
C       IDOF(I,2)=JTCOS(I)                                              NODD0660
C-------------------------------------------------------------          NODD0670
C-- CORRECT BY GER JENG-FUH JTCOS(I) REPLACED BY JTCOS(J) --            NODD0680
C-------------------------------------------------------------          NODD0690
        IDOF(I,2)=JTCOS(J)                                              NODD0700
        DO 40 K=1,3                                                     NODD0710
   40     CONST(I,K)=COORD(J,K)*SCALE                                   NODD0720
C                                                                       NODD0730
C         MOVE NODES INFO BACK TO MAIN STORAGE, DELETE DUPLICATE INFO.  NODD0740
        NNOLD=NNODE                                                     NODD0750
        I=0                                                             NODD0760
        J=0                                                             NODD0770
   43   I=I+1                                                          NODD0780
        J=J+1                                                           NODD0790
        ID(I)=IDOF(J,1)                                                 NODD0800
        IF (I.GE.2) THEN                                                NODD0810
          IF (ID(I).EQ.ID(I-1) ) THEN                                   NODD0820
            I=I-1                                                       NODD0830
            NNODE=NNODE-1                                               NODD0840
          ENDIF                                                         NODD0850
        ENDIF                                                           NODD0860
        JTCOS(I)=IDOF(J,2)                                              NODD0870
        DO 44 K=1,3                                                     NODD0880
   44     COORD(I,K)=CONST(J,K)                                         NODD0890
        IF (I.LT.NNODE) GO TO 43                                        NODD0900
C                                                                       NODD0910
   30 FORMAT(' NODE(',I3,'):',I4,'  X:',F10.4,'  Y:',F10.4,'  Z:',F10.4,NODD0920
     &                        ;'  COS#',I3)                            NODD0930
C                                                                       NODD0940
C------- GLOBAL JOINT COSINE MATRIX                                     NODD0950
        IF (BUG)  WRITE(108,1000) 'DIRECTION COSINES ... '              NODD0960
        DO 55 ICOS=1,NCOS                                               NODD0970
          READ(105,*) VX,VY                                             NODD0980
          CALL CROSS(VZ,VX,VY,0)                                        NODD0990
          CALL CROSS(VY,VZ,VX,1)                                        NODD1000
          DO 50 J=1,3                                                   NODD1010
            COSINE(1,J,ICOS)=VX(J)                                      NODD1020
            COSINE(2,J,ICOS)=VY(J)                                      NODD1030
   50       COSINE(3,J,ICOS)=VZ(J)                                     NODD1040
        IF (BUG) WRITE(108,56) ICOS,' VX:',VX,' VY:',VY,' VZ:',VZ       NODD1050
   55 CONTINUE                                                          NODD1060
   56 FORMAT(' COS(',I3,')  ',                                          NODD1070
     &       3(A,3S,F9.5,' I',SP,F9.5,' J',F9.5,' K  '),SS)             NODD1080
C                                                                       NODD1090
C------- INPUT RESTRAINTS                                               NODD1100
C        ZERO IDOF MATRIX WHICH WILL HOLD RESTRAINT, CONSTRAINT AND     NODD1110
C        ULTIMATLY DOF INFORMATION                                      NODD1120
        DO 59 I=1,NNODE                                                 NODD1130
   59     JTFLG(I)=0                                                    NODD1140
        I=0                                                             NODD1150
        FLAG=.TRUE.                                                     NODD1160
   60   I=I+1                                                          NODD1170
        IF (I.LE.NSUPT) THEN                                            NODD1180
          IF (FLAG.AND.BUG) WRITE(108,1000) ' NODE RESTRAINTS '         NODD1190
          FLAG=.FALSE.                                                  NODD1200
          READ(105, *) NODE,IFM,IGEN,NODINC                             NODD1210
   65     IF (NODE.EQ.0) THEN                                           NODD1220
            DO 70 J=1,NNODE                                             NODD1230
              DO 68 II=1,6                                              NODD1240
                IF (IFM(II).NE.0) JTFLG(J)=IBSET(JTFLG(J),II-1)         NODD1250
                IF (IFM(II).EQ.2) JTFLG(J)=IBSET(JTFLG(J),II+17)        NODD1260
   68           CONTINUE                                                NODD1270
              IF (BUG) WRITE(108,80) ID(J),IFM                          NODD1280
   70       CONTINUE                                                    NODD1290
          ELSE                                                          NODD1300
            J =IQUICK(NODE,ID,NNODE)                                    NODD1310
            IF (J.EQ 0) THEN                                            NODD1320
              IF (BUG) WRITE(108, 75) NODE                              NODD1330
            ELSE                                                        NODD1340
              DO 71 II=1,6                                              NODD1350
                IF (IFM(II).NE.0) JTFLG(J)=IBSET(JTFLG(J),II-1)         NODD1360
                IF (IFM(II).EQ.2) JTFLG(J)=IBSET(JTFLG(J),II+17)        NODD1370
   71           CONTINUE                                                NODD1380
              IF (BUG) WRITE(108,80) ID(J),IFM                          NODD1390
            ENDIF                                                       NODD1400
            IF (IGEN.GT.0) THEN                                         NODD1410
              NODE=NODE+NODINC                                          NODD1420
              IGEN=IGEN-1                                               NODD1430
              GO TO 65                                                  NODD1440
            ENDIF                                                       NODD1450
          ENDIF                                                         NODD1460
          GO TO 60                                                      NODD1470
        ENDIF                                                           NODD1480
   75 FORMAT (' NODE#',I5,' NOT FOUND, RESTRAINT IGNORED...')           NODD1490
   80 FORMAT (' NODE(',I4,')   FX:',I2,'  FY:',I2,'  FZ:',I2,           NODD1500
     &                    '  MX:',I2,'  MY:',I2,'  MZ:',I2)            NODD1510
C                                                                       NODD1520
C------ INPUT NODE DOF'S TO BE CONDENSED OUT...                         NODD1530
        I=0                                                             NODD1540
        FLAG= TRUE.                                                     NODD1550
   90   I=I+1                                                          NODD1560
        IF (I.LE.NCOND) THEN                                            NODD1570
          IF (FLAG.AND.BUG)                                             NODD1580
     &    WRITE(108,1000) ' NODE DOF''S TO BE CONDENSED OUT'            NODD1590
          FLAG= FALSE.                                                  NODD1600
          READ(105, *) NODE,IFM, IGEN,NODINC                            NODD1610
   95     IF (NODE.EQ.0) THEN                                           NODD1620
            DO 100 J=1,NNODE                                            NODD1630
              PT=.FALSE.                                                NODD1640
              DO 99 II=1,6                                              NODD1650
                IF (IFM(II).NE.0.AND. .NOT.BTEST(JTFLG(J),II-1))THENNODD1660
                  JTFLG(J)=IBSET(JTFLG(J),II+5)                         NODD1670
                  JFM(II)=IFM(II)                                       NODD1680
                  PT=.TRUE.                                             NODD1690
                ELSE                                                    NODD1700
                  JFM(II)=0                                             NODD1710
                ENDIF                                                   NODD1720
   99         CONTINUE                                                 NODD1730
  100         IF (PT.AND.BUG) WRITE(108,80) ID(J),JFM                   NODD1740
          ELSE                                                          NODD1750
            J =IQUICK(NODE,ID,NNODE)                                    NODD1760
            IF (I EQ.0) THEN                                            NODD1770
              IF (BUG) WRITE(108,110) NODE                              NODD1780
            ELSE                                                        NODD1790
              DO 101 II=1,6                                             NODD1800
                IF (IFM(II).NE.0.AND. .NOT.BTEST(JTFLG(J),II-1))THENNODD1810
                  JTFLG(J)=IBSET(JTFLG(J),II-5)                         NODD1820
                  JFM(II)=IFM(II)                                       NODD1830
                ELSE                                                    NODD1840
                  JFM(II)=0                                             NODD1850
                ENDIF                                                   NODD1860
  101           CONTINUE                                                NODD1870
              IF (PT AND BUG) WRITE(108,80) ID(J),JFM                   NODD1880
            ENDIF                                                       NODD1890
            IF (IGEN.GT 0) THEN                                         NODD1900
              NODE=NODE+NODINC                                          NODD1910
              IGEN=IGEN-1                                               NODD1920
              GO TO 95                                                  NODD1930
            ENDIF                                                       NODD1940
          ENDIF                                                         NODD1950
          GO TO 90                                                      NODD1960
        ENDIF                                                           NODD1970
  110   FORMAT (' NODE#',I5,' NOT FOUND, DOF ARE NOT CONDENSED.. ')     NODD1980
C                                                                       NODD1990
C                                                                       NODD2000
C------- INPUT CONSTRAINTS..                                            NODD2010
        I=0                                                             NODD2020
        FLAG= TRUE.                                                     NODD2030
  400   I=I+1                                                          NODD2040
        IF (I LE NCONST) THEN                                           NODD2050
          IF (FLAG) WRITE(108,1000) ' NODE CONSTRAINTS '                NODD2060
          FLAG=.FALSE.                                                  NODD2070
          READ(105, *) ITYPE,MASTER,NODE,IGEN,NODINC                    NODD2080
  410     J =IQUICK(NODE,ID,NNODE)                                      NODD2090
          IF (J.EQ.0) THEN                                              NODD2100
            WRITE(108,420) NODE                                         NODD2110
          ELSE                                                          NODD2120
            IF (ITYPE EQ.0) THEN                                        NODD2130
              TYPE='3D-RIGID BODY'                                      NODD2140
              DO 415 K=1,6                                              NODD2150
                IF (.NOT.BTEST(JTFLG(J),K-1)) THEN                      NODD2160
                  JTFLG(J)=IBSET(JTFLG(J),K-11)                         NODD2170
                  MSTRJT(J,K)=IQUICK(MASTER,ID,NNODE)                   NODD2180
                ELSE                                                    NODD2190
                  WRITE(108,430) NODE,K                                 NODD2200
                ENDIF                                                   NODD2210
  415         CONTINUE                                                  NODD2220
            ELSE IF (ITYPE.EQ.1) THEN                                   NODD2230
              TYPE='XY-PLANE'                                           NODD2240
              K=1                                                       NODD2250
                IF (.NOT.BTEST(JTFLG(J),K-1)) THEN                      NODD2260
                  JTFLG(J)=IBSET(JTFLG(J),K-11)                         NODD2270
                  MSTRJT(J,K)=IQUICK(MASTER,ID,NNODE)                   NODD2280
                ELSE                                                    NODD2290
                  WRITE(108,430) NODE,K                                 NODD2300
                ENDIF                                                   NODD2310
              K=2                                                       NODD2320
                IF (.NOT.BTEST(JTFLG(J),K-1)) THEN                      NODD2330
                  JTFLG(J)=IBSET(JTFLG(J),K-11)                         NODD2340
                  MSTRJT(J,K)=IQUICK(MASTER,ID,NNODE)                   NODD2350
                ELSE                                                    NODD2360
                  WRITE(108,430) NODE,K                                 NODD2370
                ENDIF                                                   NODD2380
              K=6                                                       NODD2390
                IF (.NOT.BTEST(JTFLG(J),K-1)) THEN                      NODD2400
                  JTFLG(J)=IBSET(JTFLG(J),K-11)                         NODD2410
                  MSTRJT(J,K)=IQUICK(MASTER,ID,NNODE)                   NODD2420
                ELSE                                                    NODD2430
                  WRITE(108,430) NODE,K                                 NODD2440
                ENDIF                                                   NODD2450
            ELSE                                                        NODD2460
              WRITE(108,440) ITYPE,MASTER,NODE                          NODD2470
            ENDIF                                                       NODD2480
            WRITE(108,450) TYPE,MASTER,NODE                             NODD2490
            ENDIF                                                       NODD2500
            IF (IGEN GT.0) THEN                                         NODD2510
              NODE=NODE+NODINC                                          NODD2520
              IGEN=IGEN-1                                               NODD2530
              GO TO 410                                                 NODD2540
            ENDIF                                                       NODD2550
          GO TO 400                                                     NODD2560
        ENDIF                                                           NODD2570
  420 FORMAT (' NODE:',I6,' IS NOT FOUND, CONSTRAINT IS IGNORED')       NODD2580
  430 FORMAT (' NODE:',I6,' IS RESTRAINED, CONSTRAINT',I2,              NODD2590
     & ' IS IGNORED. .')                                                NODD2600
  440 FORMAT (' INVALID CONSTRAINT TYPE',I3,                            NODD2610
     &        ' MASTER:',I6,' SLAVE:',I6)                              NODD2620
  450 FORMAT (13A,A,'CONSTRAINT.',                                      NODD2630
     &        ' MASTER:',I6,' SLAVE:',I6)                              NODD2640
C                                                                       NODD2650
        IF (BUG) WRITE(108,1000) 'JOINT RESTRAINT CODES...'             NODD2660
        IF (BUG) CALL PBIT(NNODE,JTFLG,ID)                              NODD2670
C                                                                       NODD2680
C---------------------------------------                                NODD2690
C--- DETERMINE THE DEGREES OF FREEDOM...  ---                           NODD2700
C---------------------------------------                                NODD2710
C.......ZERO CONSTRAINT MATRIX WHICH WILL HOLD THE MOMENT               NODD2720
C.........TRANSFORMATION FOR CONSTRAINED DOF                            NODD2730
        DO 499 I=1,NNODE                                                NODD2740
          DO 499 J=1,6                                                  NODD2750
  499     CONST(I,J)=0                                                  NODD2760
        NDOF=0                                                          NODD2770
C                                                                       NODD2780
C.....  ..DEGREES OF FREEDOM TO BE CONDENSED OUT                        NODD2790
        DO 500 I=1,NNODE                                                NODD2800
          DO 500 J=1,6                                                  NODD2810
            JT=JTFLG(I)                                                 NODD2820
            IF (BTEST(JT,J+5) .AND.  NOT.BTEST(JT,J+11)) THEN           NODD2830
                NDOF=NDOF+1                                             NODD2840
                IDOF(I,J)=NDOF                                          NODD2850
            ENDIF                                                       NODD2860
  500 CONTINUE                                                          NODD2870
        NCOND=NDOF                                                      NODD2880
C                                                                       NODD2890
C.......UNCONSTRAINED AND UNRESTRAINED DEGREES OF FREEDOM               NODD2900
        DO 510 I=1,NNODE                                                NODD2910
          DO 510 J=1,6                                                  NODD2920
            NODE=IDOF(I,J)                                              NODD2930
            IF (.NOT (BTEST(JTFLG(I),J-1) .OR. BTEST(JTFLG(I),J+5) .OR  NODD2940
     &          BTEST(JTFLG(I),J+11)) ) THEN                            NODD2950
C             IF (IDOF(I,J).EQ. 0) THEN                                 NODD2960
                NDOF=NDOF+1                                             NODD2970
                IDOF(I,J)=NDOF                                          NODD2980
            ENDIF                                                       NODD2990
  510 CONTINUE                                                          NODD3000
        NFREE=NDOF-NCOND                                                NODD3010
C                                                                       NODD3020
C.....  RESTRAINED DEGREES OF FREEDOM                                   NODD3030
        DO 530 I=1,NNODE                                                NODD3040
          DO 530 J=1,6                                                  NODD3050
            IF (BTEST(JTFLG(I),J-1) .AND. .NOT.BTEST(JTFLG(I),J+17)     NODD3060
     &                             .AND.  NOT.BTEST(JTFLG(I),J+11)) THEN NODD3070
                NDOF=NDOF+1                                             NODD3080
                IDOF(I,J)=NDOF                                          NODD3090
            ENDIF                                                       NODD3100
  530 CONTINUE                                                          NODD3110
C                                                                       NODD3120
C.......RESTRAINED DEGREES OF FREEDOM                                   NODD3130
        DO 535 I=1,NNODE                                                NODD3140
          DO 535 J=1,6                                                  NODD3150
            IF (BTEST(JTFLG(I),J+17) .AND. .NOT BTEST(JTFLG(I),J+11)) THEN NODD3160
                NDOF=NDOF+1                                             NODD3170
                IDOF(I,J)=NDOF                                          NODD3180
            ENDIF                                                       NODD3190
  535 CONTINUE                                                          NODD3200
        NREST=NDOF-NFREE-NCOND                                          NODD3210
C                                                                       NODD3220
C....  ..CONSTRAINED DEGREES OF FREEDOM                                 NODD3230
C         I - NODE ID OF SLAVE NODE                                     NODD3240
C         II- NODE ID OF MASTER NODE                                    NODD3250
        DO 520 I=1,NNODE                                                NODD3260
          DO 520 J=1,6                                                  NODD3270
C                                                                       NODD3280
C           ... CHECK DOF FOR CONSTRAINT  ..                            NODD3290
            IF (BTEST(JTFLG(I),J+11)) THEN                              NODD3300
C                                                                       NODD3310
C               ... DETERMINE MASTER NODE AND DOF # ...                 NODD3320
                II=I                                                    NODD3330
  515           NODE=ID(II)                                             NODD3340
                II=MSTRJT(II,J)                                         NODD3350
                IF (II.EQ 0) THEN                                       NODD3360
                  WRITE(108,525) NODE,ID(I),J                           NODD3370
                  IDOF(I,J)=0                                           NODD3380
                  GO TO 520                                             NODD3390
                ENDIF                                                   NODD3400
                IF (BTEST(JTFLG(II),J+11)) GO TO 515                    NODD3410
                IDOF(I,J)=IDOF(II,J)                                    NODD3420
C                                                                       NODD3430
C               .CHECK FOR DIFFERENT JOINT COORDINATE SYSTEMS ..        NODD3440
                IF (JTCOS(I) NE.JTCOS(II)) THEN                         NODD3450
                  WRITE(108,411) ID(I),ID(II)                           NODD3460
  411             FORMAT(' CONFLICTING COSINE ID NUMBERS, MASTER:',     NODD3470
     &            I6,'  SLAVE:',I6,'  CORRECT INPUT & RERUN. .')       NODD3480
                ENDIF                                                   NODD3490
C                                                                       NODD3500
C               ...CALCULATE THE CONSTRAINT MATRIX ..                   NODD3510
                IF (J.GE.4) THEN                                        NODD3520
                  XG=COORD(I,1)-COORD(II,1)                             NODD3530
                  YG=COORD(I,2)-COORD(II,2)                             NODD3540
                  ZG=COORD(I,3)-COORD(II,3)                             NODD3550
                  N  =JTCOS(I)                                          NODD3560
                  XMS=COSINE(1,1,N)*XG+COSINE(1,2,N)*YG+COSINE(1,3,N)*ZG NODD3570
```

```
                YMS=COSINE(2,1,N)*XG+COSINE(2,2,N)*YG+COSINE(2,3,N)*ZG  NODD3580
                ZMS=COSINE(3,1,N)*XG+COSINE(3,2,N)*YG+COSINE(3,3,N)*ZG  NODD3590
              ENDIF                                                     NODD3600
              IF (J.EQ.4) THEN                                          NODD3610
                  CONST(I,1)=-ZMS                                       NODD3620
                  CONST(I,2)=-YMS                                       NODD3630
              ELSE IF (J.EQ.5) THEN                                     NODD3640
                  CONST(I,3)=-ZMS                                       NODD3650
                  CONST(I,4)=-XMS                                       NODD3660
              ELSE IF (J.EQ.6) THEN                                     NODD3670
                  CONST(I,5)=-YMS                                       NODD3680
                  CONST(I,6)=-XMS                                       NODD3690
              ENDIF                                                     NODD3700
          ENDIF                                                         NODD3710
  520 CONTINUE                                                          NODD3720
  525 FORMAT (' NODE#',I6,' NOT FOUND WHILE ASSIGNING CONSTRAINED',     NODD3730
     & ' DOF TO NODE#',I6,' DOF#',I2,' --- DOF IS DELETED ')            NODD3740
C                                                                       NODD3750
C         PRINT OUT ORDERED NODE INFO, AND DEGREES OF FREEDOM           NODD3760
      WRITE(108,1000) 'NODE COORDINATES AND DEGREES OF FREEDOM'         NODD3770
      WRITE(108,600) NDOF,NCOND,NFREE,NREST                             NODD3780
  600 FORMAT                                                            NODD3790
     & (4X,'TOTAL NUMBER OF DEGREES OF FREEDOM.. ..... .',I6,/          NODD3800
     & 4X,'NUMBER OF DEGREES OF FREEDOM CONDENSED OUT.....',I6,/        NODD3810
     & 4X,'NUMBER OF FREE DEGREES OF FREEDOM  ..   ... ...',I6,/        NODD3820
     & 4X,'NUMBER OF RESTRAINED DEGREES OF FREEDOM.....',I6,//          NODD3830
     & 4X,'  NODE COS#',4X,'X-COORD',8X,'Y-COORD',8X,'Z-COORD',4X,      NODD3840
     & 4X,'FX',6X,'FY',6X,'FZ',6X,'MX',6X,'MY',6X,'MZ')                 NODD3850
      DO 610 I=1,NNODE                                                  NODD3860
        DO 605 J=1,6                                                    NODD3870
          IF (BTEST(JTFLG(I),J-1)) THEN                                 NODD3880
            CX(J)='-R'                                                  NODD3890
          ELSE IF (BTEST(JTFLG(I),(J+11))) THEN                         NODD3900
            CX(J)='-C'                                                  NODD3910
          ELSE                                                          NODD3920
            CX(J)='  '                                                 NODD3930
          ENDIF                                                         NODD3940
  605   CONTINUE                                                        NODD3950
  610  WRITE(108,620) ID(I),JTCOS(I),(COORD(I,J),J=1,3),                NODD3960
     & (IDOF(I,K),CX(K),K=1,6)                                          NODD3970
  620 FORMAT(4X,I6,1P,I5,3G15.5,0P,6(I6,A2) )                           NODD3980
      WRITE(108,625)                                                    NODD3990
  625 FORMAT(10X,'NOTE: R - RESTRAINED DEGREE OF FREEDOM',/             NODD4000
     &       10X,'      C - CONSTRAINED DEGREE OF FREEDOM')            NODD4010
C                                                                       NODD4020
C--------- PRINT OUT CONSTRAINT MATRIX                                  NODD4030
      IF (BUG) THEN                                                     NODD4040
       FLAG= TRUE                                                       NODD4050
       DO 540 I=1,NNODE                                                 NODD4060
         TEST= FALSE                                                    NODD4070
         DO 536 J=1,6                                                   NODD4080
  536    TEST = TEST .OR. BTEST(JTFLG(I),(J+11))                        NODD4090
         IF (TEST) THEN                                                 NODD4100
            IF (FLAG) WRITE(108,1000) 'NODE CONSTRAINT MATRIX '         NODD4110
            FLAG=.FALSE.                                                NODD4120
            WRITE(108,550) ID(I),(CONST(I,J),J=1,6)                     NODD4130
         ENDIF                                                          NODD4140
  540   CONTINUE                                                        NODD4150
  550  FORMAT(4X,I6,1P,                                                 NODD4160
     &     '  T1=',G12.5,'  T13=',G12.5,                                NODD4170
     &     '  T21=',G12.5,'  T23=',G12.5,                               NODD4180
     &     '  T31=',G12.5,'  T33=',G12.5)                               NODD4190
      ENDIP                                                             NODD4200
C                                                                       NODD4210
C--------- PRINT OUT DIRECTION COSINES                                  NODD4220
      WRITE(108,1000) 'DIRECTION COSINES ..  '                          NODD4230
      DO 560 ICOS=1,NCOS                                                NODD4240
          WRITE(108,56)  ICOS,' VX:',(COSINE(1,J,ICOS),J=1,3),          NODD4250
     &                   ' VY:',(COSINE(2,J,ICOS),J=1,3),              NODD4260
     &                   ' VZ:',(COSINE(3,J,ICOS),J=1,3)               NODD4270
  560 CONTINUE                                                          NODD4280
C                                                                       NODD4290
      RETURN                                                            NODD4300
 1000 FORMAT (//,5X,A/)                                                 NODD4310
      END                                                               NODD4320
C                                                                       PBIT0010
C*************************************************************************PBIT0020
C                                                                       PBIT0030
      SUBROUTINE PBIT(NNODE,JTFLG,ID)                                   PBIT0040
      DIMENSION JTFLG(NNODE),ID(NNODE)                                  PBIT0050
      LOGICAL FLAG(0:31),BTEST                                          PBIT0060
      WRITE(108,40)                                                     PBIT0070
   40 FORMAT (14X,' RESTRAINT',' CONDENSE',' CONSTRAIN',' ELIMINATE',/  PBIT0080
     &       14X,4(4X,'FFFMMM')/,14X,4(4X,'XYZXYZ'))                   PBIT0090
      DO 20 I=1,NNODE                                                   PBIT0100
        DO 10 J=0,23                                                    PBIT0110
   10     FLAG(J)=BTEST(JTFLG(I),J)                                     PBIT0120
   20   WRITE(108,30) ID(I),(FLAG(K),K=0,23),JTFLG(I)                   PBIT0130
   30 FORMAT ('   NODE(',I4,') ',4(4X,6L1),I10)                        PBIT0140
      RETURN                                                            PBIT0150
      END                                                               PBIT0160
C                                                                       ISCR0010
C*************************************************************************ISCR0020
C                                                                       ISCR0030
      SUBROUTINE ISORT(ID,INDEX,N)                                      ISCR0040
      INTEGER ID(N),INDEX(N)                                            ISCR0050
      LOGICAL FLAG                                                      ISCR0060
      DO 10 I=1,N                                                       ISCR0070
        INDEX(I)=I                                                      ISCR0080
   10 CONTINUE                                                          ISCR0090
C                                                                       ISCR0100
      DO 30 I=N,2,-1                                                    ISCR0110
       FLAG= TRUE.                                                      ISCR0120
       DO 20 J=2,I                                                      ISCR0130
        IF ( ID(INDEX(J)).LT.ID(INDEX(J-1)) ) THEN                      ISCR0140
            K=INDEX(J)                                                  ISCR0150
            INDEX(J)=INDEX(J-1)                                         ISCR0160
            INDEX(J-1)=K                                                ISCR0170
            FLAG=.FALSE.                                                ISCR0180
        ENDIF                                                           ISCR0190
   20  CONTINUE                                                         ISCR0200
       IF (FLAG) RETURN                                                 ISCR0210
   30  CONTINUE                                                         ISCR0220
      RETURN                                                            ISCR0230
      END                                                               ISCR0240
C*************************************************************************DMPD0010
      SUBROUTINE DMPDAT(IOPT,INRITE,TO,DT)                              DMPD0020
      LOGICAL TEST,OPEND,FALSE,AXIAL,FIRST                              DMPD0030
      CHARACTER*40 OPTION                                               DMPD0040
      CHARACTER*80 NAME, DIR*2                                          DMPD0050
      CHARACTER*112 TITL(2)                                             DMPD0060
      CHARACTER*12 INAME                                                DMPD0070
      DIMENSION RINPUT(100)                                             DMPD0080
      SAVE FIRST                                                        DMPD0090
$INCLUDE:'ZCOMN'                                                        DMPD0100
C                                                                       DMPD0110
      GO TO (100,200,300,400) IOPT                                      DMPD0120
C                                                                       DMPD0130
C*****************************************************                 DMPD0140
C- READ DATA ID TO BE SAVED, AND INITILIZE FILES  ---                   DMPD0150
C*****************************************************                 DMPD0160
C                                                                       DMPD0170
  100 FIRST = .TRUE.                                                    DMPD0180
      IZDUMP=IZ                                                         DMPD0190
      IZ=IZ+1                                                           DMPD0200
      NWRITE=0                                                          DMPD0210
      WRITE(108,109)                                                    DMPD0220
  109 FORMAT (/,5X,' DATA WRITTEN TO FILES '/                           DMPD0230
     &       5X,'----------:----------'/)                               DMPD0240
  101 READ(105,*) OPTION,IDOF,IUNIT,IGEN,IIDOF,IIUNIT                  DMPD0250
  102 CONTINUE                                                          DMPD0260
C                                                                       DMPD0270
C-----SET UP DOF DATA                                                   DMPD0280
      IF (TEST(OPTION,'DOF',.FALSE.)) THEN                              DMPD0290
         WRITE(108,110) 'DEGREE OF FREEDOM',IDOF,IUNIT                  DMPD0300
C                                                                       DMPD0310
C        CHECK FILE STATUS                                              DMPD0320
             INQUIRE(UNIT=IUNIT,OPENED=OPEND)                           DMPD0330
             IF (.NOT.OPEND) OPEN (UNIT=IUNIT)                          DMPD0340
C                                                                       DMPD0350
C        INITILIZE DATA AND FILE                                        DMPD0360
         NWRITE=NWRITE+1                                                DMPD0370
         NZ(IZ)=1                                                       DMPD0380
         NZ(IZ+1)=IDOF                                                  DMPD0390
         NZ(IZ+2)=IUNIT                                                 DMPD0400
         IZ=IZ+3                                                        DMPD0410
         WRITE(IUNIT,120) TITLE(1)(1:80),TITLE(1)(81:)                  DMPD0420
         WRITE(IUNIT,120) TITLE(2)(1:80),TITLE(2)(81:)                  DMPD0430
         WRITE(IUNIT,130) 1,IDOF,IWRITE,TO,DT                           DMPD0440
C---- -------- SET UP JOINT                                            DMPD0450
      ELSE IF (TEST(OPTION,'JOINT',.FALSE.)) THEN                       DMPD0460
         JOINT=IDOF                                                     DMPD0470
         JOINTI=IQUICK(IDOF,NZ(IZID),NNODE)-1                           DMPD0480
         IF (TEST(OPTION,'FX',.FALSE.)) THEN                            DMPD0490
            IDOFJ=NZ(IZIDOF+JOINTI)                                     DMPD0500
            DIR='FX'                                                    DMPD0510
         ELSE IF (TEST(OPTION,'FY',.FALSE.)) THEN                       DMPD0520
            IDOFJ=NZ(IZIDOF+JOINTI+MAXNOD*1)                            DMPD0530
            DIR='FY'                                                    DMPD0540
         ELSE IF (TEST(OPTION,'FZ',.FALSE.)) THEN                       DMPD0550
            IDOFJ=NZ(IZIDOF+JOINTI+MAXNOD*2)                            DMPD0560
            DIR='FZ'                                                    DMPD0570
         ELSE IF (TEST(OPTION,'MX',.FALSE.)) THEN                       DMPD0580
            IDOFJ=NZ(IZIDOF+JOINTI+MAXNOD*3)                            DMPD0590
            DIR='MX'                                                    DMPD0600
         ELSE IF (TEST(OPTION,'MY',.FALSE.)) THEN                       DMPD0610
            IDOFJ=NZ(IZIDOF+JOINTI+MAXNOD*4)                            DMPD0620
            DIR='MY'                                                    DMPD0630
         ELSE IF (TEST(OPTION,'MZ',.FALSE.)) THEN                       DMPD0640
            IDOFJ=NZ(IZIDOF+JOINTI+MAXNOD*5)                            DMPD0650
            DIR='MZ'                                                    DMPD0660
         ENDIF                                                          DMPD0670
C                                                                       DMPD0680
         WRITE(108,110) 'DEGREE OF FREEDOM',IDOFJ,IUNIT,JOINT,DIR       DMPD0690
C                                                                       DMPD0700
C        CHECK FILE STATUS                                              DMPD0710
             INQUIRE (UNIT=IUNIT,OPENED=OPEND)                          DMPD0720
             IF (.NOT.OPEND) OPEN (UNIT=IUNIT)                          DMPD0730
C                                                                       DMPD0740
C        INITILIZE DATA AND FILE                                        DMPD0750
         NWRITE=NWRITE+1                                                DMPD0760
         NZ(IZ)=1                                                       DMPD0770
         NZ(IZ+1)=IDOFJ                                                 DMPD0780
         NZ(IZ+2)=IUNIT                                                 DMPD0790
         IZ=IZ+3                                                        DMPD0800
         WRITE(IUNIT,120) TITLE(1)(1:80),TITLE(1)(81:)                  DMPD0810
         WRITE(IUNIT,120) TITLE(2)(1:80),TITLE(2)(81:)                  DMPD0820
         WRITE(IUNIT,130) 1,IDOFJ,IWRITE,TO,DT,JOINT,DIR                DMPD0830
C                                                                       DMPD0840
C---- SET UP ELEMENT                                                    DMPD0850
      ELSE IF (TEST(OPTION,'ELE',.FALSE.)) THEN                         DMPD0860
         WRITE(108,110) 'ELEMENT',IDOF,IUNIT                            DMPD0870
C                                                                       DMPD0880
C        CHECK FILE STATUS                                              DMPD0890
             INQUIRE (UNIT=IUNIT,OPENED=OPEND)                          DMPD0900
             IF (.NOT.OPEND) OPEN (UNIT=IUNIT)                          DMPD0910
C                                                                       DMPD0920
C        INITILIZE DATA AND FILE                                        DMPD0930
         NWRITE=NWRITE+1                                                DMPD0940
         NZ(IZ)=2                                                       DMPD0950
         NZ(IZ+1)=IDOF                                                  DMPD0960
         NZ(IZ+2)=IUNIT                                                 DMPD0970
         IZ=IZ+3                                                        DMPD0980
C                                                                       DMPD0990
C        GET ELEMENT TYPE                                               DMPD1000
         IC    = NZ(IZELE+IDOF )                                        DMPD1010
         IELTYP = NZ(IC)                                                DMPD1020
C                                                                       DMPD1030
         WRITE(IUNIT,120) TITLE(1)(1:80),TITLE(1)(81:)                  DMPD1040
         WRITE(IUNIT,120) TITLE(2)(1:80),TITLE(2)(81:)                  DMPD1050
         WRITE(IUNIT,130) 2,IDOF,IWRITE,TO,DT,IELTYP,'ELEMENT'          DMPD1060
C                                                                       DMPD1070
C-----SET UP ENERGY                                                     DMPD1080
      ELSE IF (TEST(OPTION,'ENERGY',.FALSE.)) THEN                      DMPD1090
         WRITE(108,111) 'ENERGY BALANCE',IUNIT                          DMPD1100
C                                                                       DMPD1110
C        CHECK FILE STATUS                                              DMPD1120
             INQUIRE (UNIT=IUNIT,OPENED=OPEND)                          DMPD1130
             IF (.NOT.OPEND) OPEN (UNIT=IUNIT)                          DMPD1140
C                                                                       DMPD1150
C        INITILIZE DATA AND FILE                                        DMPD1160
         NWRITE=NWRITE+1                                                DMPD1170
         NZ(IZ)=3                                                       DMPD1180
         NZ(IZ+1)=0                                                     DMPD1190
         NZ(IZ+2)=IUNIT                                                 DMPD1200
         IZ=IZ+3                                                        DMPD1210
         WRITE(IUNIT,120) TITLE(1)(1:80),TITLE(1)(81:)                  DMPD1220
         WRITE(IUNIT,120) TITLE(2)(1:80),TITLE(2)(81:)                  DMPD1230
         WRITE(IUNIT,130) 3,IDOF,IWRITE,TO,DT,0,'ENERGY'                DMPD1240
C                                                                       DMPD1250
C-----SET UP SUM OF REACTIONS                                           DMPD1260
      ELSE IF (TEST(OPTION,'SUMRCT',.FALSE.)) THEN                      DMPD1270
         WRITE(108,111) 'SUMMATION OF REACTIONS',IUNIT                  DMPD1280
C                                                                       DMPD1290
C        CHECK FILE STATUS                                              DMPD1300
             INQUIRE (UNIT=IUNIT,OPENED=OPEND)                          DMPD1310
             IF (.NOT.OPEND) OPEN (UNIT=IUNIT)                          DMPD1320
C                                                                       DMPD1330
C        INITILIZE DATA AND FILE                                        DMPD1340
         NWRITE=NWRITE+1                                                DMPD1350
         NZ(IZ)=4                                                       DMPD1360
         NZ(IZ+1)=0                                                     DMPD1370
         NZ(IZ+2)=IUNIT                                                 DMPD1380
         IZ=IZ+3                                                        DMPD1390
         WRITE(IUNIT,120) TITLE(1)(1:80),TITLE(1)(81:)                  DMPD1400
         WRITE(IUNIT,120) TITLE(2)(1:80),TITLE(2)(81:)                  DMPD1410
         WRITE(IUNIT,130) 4,IDOF,IWRITE,TO,DT,0,'SUMRCT'                DMPD1420
C                                                                       DMPD1430
C-----SET UP SUM OF REACTIONS                                           DMPD1440
      ELSE IF (TEST(OPTION,'GROUND',.FALSE.)) THEN                      DMPD1450
         WRITE(108,111) 'GROUND MOTION',IUNIT                           DMPD1460
C                                                                       DMPD1470
C        CHECK FILE STATUS                                              DMPD1480
             INQUIRE (UNIT=IUNIT,OPENED=OPEND)                          DMPD1490
             IF (.NOT.OPEND) OPEN (UNIT=IUNIT)                          DMPD1500
C                                                                       DMPD1510
C        INITILIZE DATA AND FILE                                        DMPD1520
         NWRITE=NWRITE+1                                                DMPD1530
         NZ(IZ)=5                                                       DMPD1540
         NZ(IZ+1)=0                                                     DMPD1550
         NZ(IZ+2)=IUNIT                                                 DMPD1560
         IZ=IZ+3                                                        DMPD1570
         WRITE(IUNIT,120) TITLE(1)(1:80),TITLE(1)(81:)                  DMPD1580
         WRITE(IUNIT,120) TITLE(2)(1:80),TITLE(2)(81:)                  DMPD1590
         WRITE(IUNIT,130) 5,IDOF,IWRITE,TO,DT,0,'GROUND'                DMPD1600
C                                                                       DMPD1610
      ELSE IF (TEST(OPTION,'END',.FALSE.)) THEN                         DMPD1620
         NZ(IZDUMP)=NWRITE                                              DMPD1630
         RETURN                                                         DMPD1640
      ELSE                                                              DMPD1650
         WRITE(108,140) OPTION                                          DMPD1660
         GO TO 101                                                      DMPD1670
      ENDIF                                                             DMPD1680
C                                                                       DMPD1690
C..  ....  GENERATE LOADS                                               DMPD1700
      IF (IGEN.NE.0) THEN                                               DMPD1710
         IGEN=IGEN-1                                                    DMPD1720
         IDOF =IDOF  + IIDOF                                            DMPD1730
         IUNIT=IUNIT + IIUNIT                                           DMPD1740
         GO TO 102                                                      DMPD1750
      ENDIP                                                             DMPD1760
      GO TO 101                                                         DMPD1770
```

```
C                                                                       DMPD1780
  105 FORMAT (5X,'FILE IS ALREADY OPENED, WRITE IS ABORTED',10('*'))    DMPD1790
  110 FORMAT (5X,A,'   #',I5,' IS WRITTEN TO UNIT #',I5,,'  JOINT:',I6, DMPD1800
     &        '   DIRECTION: ',A)                                      DMPD1810
  111 FORMAT (5X,A,' IS WRITTEN TO UNIT #',I5 )                         DMPD1820
  120 FORMAT (A)                                                        DMPD1830
  130 FORMAT (3I10,1P,2G15.5,0P,1X,I6,A)                                DMPD1840
  140 FORMAT (5X,'INVALID INPUT: ',A,', INPUT IS SKIPED...')            DMPD1850
C                                                                       DMPD1860
C------------------------                                               DMPD1870
C- WRITE DATA TO FILES ---                                            DMPD1880
C------------------------                                               DMPD1890
C                                                                       DMPD1900
  200 NUMB=NZ(IZDUMP)                                                   DMPD1910
      DO 250 I=1,NUMB                                                   DMPD1920
         II=IZDUMP+3*(I-1) +1                                           DMPD1930
         ITYPE=NZ(II)                                                   DMPD1940
         IDOF =NZ(II+1)                                                 DMPD1950
         IUNIT=NZ(II+2)                                                 DMPD1960
         IF (ITYPE.EQ 1) THEN                                           DMPD1970
              J=IDOF+1                                                  DMPD1980
              IF (IZVEL.EQ.0  OR.  IZACC.EQ.0) THEN                     DMPD1990
                WRITE(IUNIT,210) Z(IZLOAD+J),Z(IZDISP+J),0.,0.          DMPD2000
              ELSE                                                      DMPD2010
                WRITE(IUNIT,210) Z(IZLOAD+J),Z(IZDISP+J),               DMPD2020
     &                           Z(IZVEL+J),Z(IZACC+J)                  DMPD2030
              ENDIF                                                     DMPD2040
         ELSE IF (ITYPE.EQ 2) THEN                                      DMPD2050
              I7=7                                                      DMPD2060
              IELNO=IDOF                                                DMPD2070
              LASTYP=0                                                  DMPD2080
              FALSE=.FALSE.                                             DMPD2090
              CALL ELELIB                                               DMPD2100
     &        (I7,LSTTYP,FIRST,IREL,FIRST,NAME,EESE,EPSE,DAMAGE,DUCFAG, DMPD2110
     &        IELNO,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDDSP,IZDLOA,MSTOR)DMPD2120
         ELSE IF (ITYPE.EQ.3) THEN                                      DMPD2130
              DIFF=EIE-ESE-EKE-PSE-EDD                                  DMPD2140
              IF (EIE NE.0) THEN                                        DMPD2150
                 RE=DIFF/EIE * 100                                      DMPD2160
              ELSE                                                      DMPD2170
                 IF (DIFF.NE.0) RE=100.00                               DMPD2180
                 IF (DIFF.EQ 0) RE=  0.00                               DMPD2190
              ENDIF                                                     DMPD2200
              WRITE(IUNIT,210) EIE, ESE,EKE,  PSE,EDD,  RE              DMPD2210
         ELSE IF (ITYPE.EQ.4) THEN                                      DMPD2220
              WRITE(IUNIT,210) SUMRCT                                   DMPD2230
         ELSE IF (ITYPE.EQ.5) THEN                                      DMPD2240
              WRITE(IUNIT,210) GROUND                                   DMPD2250
         ENDIF                                                          DMPD2260
  250 CONTINUE                                                          DMPD2270
  210 FORMAT(1P,6G13.5)                                                 DMPD2280
      FIRST=.FALSE.                                                     DMPD2290
      RETURN                                                            DMPD2300
C                                                                       DMPD2310
C---------------------------------------                                DMPD2320
C- READ DATA FROM FILE AND WRITE TO FT06 ---                           DMPD2330
C---------------------------------------                                DMPD2340
C                                                                       DMPD2350
  400 REWIND(UNIT=IUNIT)                                                DMPD2360
      READ(IUNIT,120,END=499) TITL(1)(1:80),TITL(1)(81:)                DMPD2370
      READ(IUNIT,120,END=499) TITL(2)(1:80),TITL(2)(81:)                DMPD2380
      WRITE(108,405) TITL(1),TITL(2)                                    DMPD2390
  405 FORMAT ('1 STRUCTURE.. .' ',A,/'  SOLUTION..... ',A,/)            DMPD2400
      READ(IUNIT,131,END=499) ITYPE,IDOF,IWRITE,TO,DT,ITEMP,INAME       DMPD2410
  131 FORMAT (3I10,0P,2G15.5,0P,1X,I6,A)                                DMPD2420
C                                                                       DMPD2430
C------- DOF DATA --------------                                       DMPD2440
      IF (ITYPE.EQ.1) THEN                                              DMPD2450
         WRITE(108,450) IDOF,IUNIT,ITEMP,INAME                          DMPD2460
         ISTEP=-IWRITE                                                  DMPD2470
  410    READ(IUNIT,*,END=440) RLOAD,DISP,VEL,ACC                       DMPD2480
         ISTEP=ISTEP+IWRITE                                             DMPD2490
         T=TO+DT*ISTEP                                                  DMPD2500
         IF (MOD(ISTEP,INC).EQ.0)                                       DMPD2510
     &      WRITE(108,460) ISTEP,T,RLOAD,DISP,VEL,ACC                   DMPD2520
         GO TO 410                                                      DMPD2530
      ELSE IF (ITYPE.EQ 2) THEN                                         DMPD2540
         IELNO =IDOF                                                    DMPD2550
         IELTYP=ITEMP                                                   DMPD2560
         WRITE(108,445) IELNO,IUNIT                                     DMPD2570
C        GET ELEMENT TYPE AND COMPARE WITH INPUT TYPE... .              DMPD2580
         IF (IZELE.GT.0) THEN                                           DMPD2590
              IC= NZ(IZELE+IELNO)                                       DMPD2600
         ELSE                                                           DMPD2610
              IC=-1                                                     DMPD2620
         ENDIF                                                          DMPD2630
         IF (IC.GT.0 .AND. IC.LE.MAXZ) THEN                             DMPD2640
              IELTPO = NZ(IC)                                           DMPD2650
         ELSE                                                           DMPD2660
              IELTPO = -100                                             DMPD2670
         ENDIF                                                          DMPD2680
C        CREATE DUMMY ELEMENT DIRECTORY IF INPUT TYPE IS DIFF           DMPD2690
         IZELET=IZELE                                                   DMPD2700
         IF (IELTYP.NE.IELTPO) THEN                                     DMPD2710
            CALL CKSTOR(IELNO+2,0)                                      DMPD2720
            IZELE = IZ+2                                                DMPD2730
            NZ(IZELE+IELNO  )=IZELE+IELNO+1                             DMPD2740
            NZ(IZELE+IELNO+1)=IELTYP                                    DMPD2750
         ENDIF                                                          DMPD2760
         IB=8                                                           DMPD2770
         LASTYP=0                                                       DMPD2780
         FALSE=.FALSE.                                                  DMPD2790
         CALL ELELIB                                                    DMPD2800
     &    (I6,LSTTYP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DAMAGE,DUCFAG,     DMPD2810
     &    IELNO,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDDSP,IZDLOA,MSTOR)    DMPD2820
C        RESTORE OLD   ELEMENT DIRECTORY                                DMPD2830
         IZELE=IZELET                                                   DMPD2840
      ELSE IF (ITYPE.EQ.3) THEN                                         DMPD2850
         TEI=0.00                                                       DMPD2860
         TSE=0.00                                                       DMPD2870
         DSE=0.00                                                       DMPD2880
         EIEM=0                                                         DMPD2890
         ESEM=0                                                         DMPD2900
         EKEM=0                                                         DMPD2910
         PSEM=0                                                         DMPD2920
         EDDM=0                                                         DMPD2930
         WRITE(108,455) IUNIT                                           DMPD2940
         ISTEP=-IWRITE                                                  DMPD2950
  430    READ(IUNIT,*,END=431) EIE, ESE,EKE,  PSE,EDD,  RE              DMPD2960
         TEI=TEI+EIE                                                    DMPD2970
         TSE=TSE+ESE+EKE+PSE+EDD                                        DMPD2980
         DSE=DSE+ABS(EIE-ESE-EKE-PSE-EDD)                               DMPD2990
         EIEM=AMAX1(SNGL(EIE),EIEM)                                     DMPD3000
         ESEM=AMAX1(SNGL(ESE),ESEM)                                     DMPD3010
         EKEM=AMAX1(SNGL(EKE),EKEM)                                     DMPD3020
         PSEM=AMAX1(SNGL(PSE),PSEM)                                     DMPD3030
         EDDM=AMAX1(SNGL(EDD),EDDM)                                     DMPD3040
         ISTEP=ISTEP+IWRITE                                             DMPD3050
         T=TO+DT*ISTEP                                                  DMPD3060
         IF (MOD(ISTEP,INC).EQ.0)                                       DMPD3070
     &      WRITE(108,459) ISTEP,T,EIE,ESE,EKE,PSE,EDD,RE               DMPD3080
         GO TO 430                                                      DMPD3090
  431    RE =200*DSE/(TEI+TSE)                                          DMPD3100
         WRITE(108,458) EIEM,ESEM,EKEM,PSEM,EDDM,RE                     DMPD3110
         GO TO 440                                                      DMPD3120
      ELSE IF (ITYPE.EQ.4) THEN                                         DMPD3130
         WRITE(108,456) IUNIT                                           DMPD3140
         ISTEP=-IWRITE                                                  DMPD3150
  435    READ(IUNIT,*,END=440) SUMRCT                                   DMPD3160
         ISTEP=ISTEP+IWRITE                                             DMPD3170
         T=TO+DT*ISTEP                                                  DMPD3180
         IF (MOD(ISTEP,INC).EQ.0) WRITE(108,460) ISTEP,T, SUMRCT        DMPD3190
         GO TO 435                                                      DMPD3200
      ELSE IF (ITYPE.EQ.5) THEN                                         DMPD3210
         WRITE(108,457) IUNIT                                           DMPD3220
         ISTEP=-IWRITE                                                  DMPD3230
  436    READ(IUNIT,*,END=440) GROUND                                   DMPD3240
         ISTEP=ISTEP+IWRITE                                             DMPD3250
         T=TO+DT*ISTEP                                                  DMPD3260
         IF (MOD(ISTEP,INC).EQ.0) WRITE(108,461) ISTEP,T, GROUND        DMPD3270
         GO TO 436                                                      DMPD3280
      ENDIF                                                             DMPD3290
C                                                                       DMPD3300
  440 RETURN                                                            DMPD3310
C                                                                       DMPD3320
  499 WRITE(108,500) IUNIT                                              DMPD3330
  500 FORMAT (///10X,'PREMATURE END OF FILE, UNIT ',I5/                 DMPD3340
     &        10X,'READ COMMAND IS ABORTED'/// )                        DMPD3350
C                                                                       DMPD3360
  445 FORMAT (5X,'ELEMENT #',I5,' IS READ FROM UNIT #',I5)              DMPD3370
  450 FORMAT (5X,'DEGREE OF FREEDOM #',I5,' IS READ FROM UNIT #',I5,    DMPD3380
     &        5X,'JOINT #',I5,', DIRECTION. ',A2//                      DMPD3390
     & 6X,'STEP',        '     TIME       ',',     LOAD       ',      DMPD3400
     &' DISPLACEMENT  ',' VELOCITY    ',' ACCELERATION  ')             DMPD3410
  455 FORMAT (5X,'ENERGY DATA IS READ FROM UNIT #',I5//                 DMPD3420
     & 6X,'STEP',        '     TIME       ',',     INPUT      ',      DMPD3430
     &'ELASTIC STRAIN ',' KINETIC     ','PLASTIC STRAIN ',             DMPD3440
     &' DAMPINED     ',' RELATIVE ERROR %')                            DMPD3450
  456 FORMAT (5X,'SUMMATION OF REACTIONS ARE READ FROM UNIT #',I5//     DMPD3460
     & 6X,'STEP',        '     TIME       ',',     FX         ',      DMPD3470
     &'     FY         ',',     FZ         ',',     MX         ',      DMPD3480
     &'     MY         ',',     MZ         ')                          DMPD3490
  457 FORMAT (5X,'GROUND MOTION IS READ FROM UNIT #',I5//               DMPD3500
     & 2X,'    ',        '     ',        '     ',                      DMPD3510
     &' /--------- ACCELERATION ---------/',                           DMPD3520
     &' /----------- VELOCITY -----------/',                           DMPD3530
     &' /---------- DISPLACEMENT ----------/', /                       DMPD3540
     & 2X,'STEP',        '     TIME       ',                          DMPD3550
     &'  GLOBAL X  ',' GLOBAL Y  ',' GLOBAL Z  ',                      DMPD3560
     &'  GLOBAL X  ',' GLOBAL Y  ',' GLOBAL Z  ',                      DMPD3570
     &'  GLOBAL X  ',' GLOBAL Y  ',' GLOBAL Z  ')                      DMPD3580
  458 FORMAT(25X,90('-')/10X,'  MAXIMUN       ',1P,5G15.5,/            DMPD3590
     &    /25X,'GROSS RE:',0P,F15.4,/'%')                              DMPD3600
  459 FORMAT(I10,1P,6G15.5,0P,F15.4)                                    DMPD3610
  460 FORMAT(I10,1P,7G15.5)                                             DMPD3620
  461 FORMAT(I6,1P,10G12.4)                                             DMPD3630
C                                                                       DMPD3640
  300 RETURN                                                            DMPD3650
      END                                                               DMPD3660
C                                                                       GETC0010
C-----------------------------------------------------------------------GETC0020
C                                                                       GETC0030
C SUBROUTINE GETCHR DETERMINS THE VALUE OF THE FIRST CHARACTER          GETC0040
C                   AFTER AN OPTIONAL (LWORD=.TRUE.) KEYWORD (WORD).    GETC0050
C                   AND REMOVES THE STRING (WORD)//CHAR     IF          GETC0060
C                   MODIFY=.TRUE.                                       GETC0070
C                                                                       GETC0080
      SUBROUTINE GETCHR(VERB,WORD,CHAR,LWORD,MODIFY)                    GETC0090
C                                                                       GETC0100
      CHARACTER*(*) VERB,WORD,CHAR                                      GETC0110
      LOGICAL MODIFY, LWORD                                             GETC0120
C                                                                       GETC0130
  999 CHAR=' '                                                          GETC0140
      IF (LWORD) THEN                                                   GETC0150
           ISTART=INDEX(VERB,WORD)                                      GETC0160
           IF (ISTART.LE 0) RETURN                                      GETC0170
           I=ISTART+LEN(WORD)                                           GETC0180
      ELSE                                                              GETC0190
           I=1                                                          GETC0200
           ISTART=1                                                     GETC0210
           CALL FIRST(VERB)                                             GETC0220
      ENDIF                                                             GETC0230
C                                                                       GETC0240
      IF (VERB(I:I).EQ.'''' ) THEN                                      GETC0250
         I=I+1                                                          GETC0260
         IJ=I                                                           GETC0270
  10     J=INDEX(VERB(IJ:),'''')-2  + IJ                               GETC0280
           IF (VERB((J+2):(J+2)).EQ.'''') THEN                          GETC0290
                IJ=J+3                                                  GETC0300
                GO TO 10                                                GETC0310
           ENDIF                                                        GETC0320
         IF (J EQ.0) THEN                                               GETC0330
              CHAR=VERB(I: )                                            GETC0340
              GO TO 10                                                  GETC0350
         ELSE                                                           GETC0360
              CHAR=VERB(I:J)                                            GETC0370
         ENDIF                                                          GETC0380
  20      IJ=INDEX(CHAR,'''''')                                         GETC0390
          IF (IJ.NE.0) THEN                                             GETC0400
             CHAR=CHAR(:IJ)//CHAR((IJ+2):)                              GETC0410
             GO TO 20                                                   GETC0420
             ENDIF                                                      GETC0430
      ELSE                                                              GETC0440
         J=INDEX(VERB(I:),' ') + I -2                                   GETC0450
         IF (J.GE.I) CHAR=VERB(I:J)                                     GETC0460
      ENDIF                                                             GETC0470
C                                                                       GETC0480
      IF (MODIFY) THEN                                                  GETC0490
        IF (VERB(J+2:J+2).EQ.',') J=J+1                                 GETC0500
        VERB(ISTART: )=VERB(J+2: )                                      GETC0510
        CALL FIRST(VERB)                                                GETC0520
      ENDIF                                                             GETC0530
C                                                                       GETC0540
      RETURN                                                            GETC0550
      END                                                               GETC0560
C                                                                       FIRS0010
C     STRIPS PRECEDING BLANK SPACES OFF A CHARACTER STRING              FIRS0020
C                                                                       FIRS0030
      SUBROUTINE FIRST(WORD)                                            FIRS0040
      CHARACTER*(*) WORD                                                FIRS0050
           IF (INDEX(WORD,' ').EQ.0) RETURN                             FIRS0060
      DO 10 I=1,(LEN(WORD))                                             FIRS0070
           J=INDEX(WORD(I:),' ')                                        FIRS0080
           IF (J.EQ 1) GO TO 10                                         FIRS0090
              WORD=WORD(I:)                                             FIRS0100
              RETURN                                                    FIRS0110
  10       CONTINUE                                                     FIRS0120
  20  RETURN                                                            FIRS0130
      END                                                               FIRS0140
C                                                                       GETI0010
C SUBROUTINE GETINT DETERMINS THE VALUE OF THE FIRST INTEGER (INTEG)    GETI0020
C                   AFTER AN OPTIONAL (LWORD= TRUE.) KEYWORD (WORD)     GETI0030
C                   AND REMOVES THE STRING (WORD)//INTEGER IF           GETI0040
C                   MODIFY=.TRUE.  INTEG CAN BE ANY LENGTH REPRESENTABLEGETI0050
C                   BY SINGLE PRECESION.                                GETI0060
C                                                                       GETI0070
      SUBROUTINE GETINT(VERB,WORD,INTEG,DEFULT,LWORD,MODIFY)            GETI0080
C                                                                       GETI0090
      CHARACTER*(*) VERB,WORD                                           GETI0100
      CHARACTER*10  NUMB                                                GETI0110
      LOGICAL MODIFY, LWORD, HIT                                        GETI0120
      INTEGER DEFULT                                                    GETI0130
      DATA NUMB/'0123456789'/                                           GETI0140
C                                                                       GETI0150
      INTEG=DEFULT                                                      GETI0160
      HIT=.TRUE                                                         GETI0170
      IF (LWORD) THEN                                                   GETI0180
           ISTART=INDEX(VERB,WORD)                                      GETI0190
           IF (ISTART LE 0) RETURN                                      GETI0200
           I=ISTART+LEN(WORD)                                           GETI0210
      ELSE                                                              GETI0220
           I=1                                                          GETI0230
           ISTART=1                                                     GETI0240
      ENDIF                                                             GETI0250
C                                                                       GETI0260
      ISIGN = 1                                                         GETI0270
      L=LEN(VERB)                                                       GETI0280
    5 IF (I GT.L) RETURN                                                GETI0290
      IF (VERB(I:I).EQ.'-') THEN                                        GETI0300
         ISIGN =-1                                                      GETI0310
         I=I+1                                                          GETI0320
         GO TO 5                                                        GETI0330
```

```
      ELSE IF (VERB(I,I).EQ.'*'  OR.VERB(I:I).EQ.' ') THEN
         I=I+1
         GO TO 5
      ENDIF
C
   10 IF (LGE(VERB(I,I),'0') .AND. LLE(VERB(I:I),'9') ) THEN
         IF ( HIT) THEN
              HIT=.FALSE.
              INTEG=0
         ENDIF
         DO 20 J=1,10
          IF(VERB(I,I).EQ. NUMB(J,J) ) THEN
               INTEG=10*INTEG+ (J-1)
               I=I+1
               GO TO 10
          ENDIF
   20    CONTINUE
      ENDIF
C
      INTEG=INTEG*ISIGN
C
      IF (MODIFY) THEN
         IF (VERB(I,I).EQ.',') I=I+1
         VERB(ISTART: )=VERB(I: )
      ENDIF
C
      RETURN
      END
C
C-----------------------------------------------------------------------
C
      SUBROUTINE CROSS(VZ,VX,VY,NORM)
      DIMENSION VX(3),VY(3),VZ(3)
C
      CROSSI(A1,B1,C1,A2,B2,C2)=B1*C2-B2*C1
      CROSSJ(A1,B1,C1,A2,B2,C2)=C1*A2-C2*A1
      CROSSK(A1,B1,C1,A2,B2,C2)=A1*B2-A2*B1
      VALUE (A1,A2,A3)=SQRT(A1**2+A2**2+A3**2)
C
C ------- CALCULATE VZ, IF VY WAS CORRECTLY INPUT
      VZ(1)=CROSSI(VX(1),VX(2),VX(3),VY(1),VY(2),VY(3))
      VZ(2)=CROSSJ(VX(1),VX(2),VX(3),VY(1),VY(2),VY(3))
      VZ(3)=CROSSK(VX(1),VX(2),VX(3),VY(1),VY(2),VY(3))
C
C ------  NORMALIZE VX,VY,VZ
      IF (NORM.NE.0) THEN
           AVX=VALUE(VX(1),VX(2),VX(3))
           AVY=VALUE(VY(1),VY(2),VY(3))
           AVZ=VALUE(VZ(1),VZ(2),VZ(3))
           DO 10 I=1,3
              VX(I)=VX(I)/AVX
              VY(I)=VY(I)/AVY
   10         VZ(I)=VZ(I)/AVZ
C
      ENDIF
      RETURN
      END
C
C --  LOGICAL FUNCTION TEST
C
      LOGICAL FUNCTION TEST(OPTION,TARGET,DEFLT)
      CHARACTER*(*) OPTION,TARGET
      LOGICAL DEFLT
C
         J= INDEX(OPTION,TARGET)
         IF (DEFLT) THEN
              IF (J.LT.2) THEN
                   TEST=.TRUE.
              ELSE IF (OPTION((J-2):(J-1)).NE.'NO') THEN
                   TEST= TRUE.
              ELSE
                   TEST=.FALSE.
              ENDIF
         ELSE
              IF (J.EQ.0) THEN
                   TEST=.FALSE.
              ELSE IF (J LT.2 ) THEN
                   TEST=.TRUE.
              ELSE IF (OPTION((J-2):(J-1)).EQ.'NO') THEN
                   TEST=.FALSE.
              ELSE
                   TEST=.TRUE.
              ENDIF
         ENDIF
      RETURN
      END
C
C ---- INTEGER FUNCTION IQUICK
C
      INTEGER FUNCTION IQUICK(TARGET,ID,N)
      INTEGER TARGET,ID(N)
C
      IF (ID(1).GT.TARGET  OR. ID(N).LT.TARGET ) GO TO 100
      IQUICK=1
      IF (ID(IQUICK).EQ.TARGET ) RETURN
      IQUICK=N
      IF (ID(IQUICK).EQ.TARGET ) RETURN
      J=(N+1)/2
      IQUICK=J
   10 J= J    /2
      IF (IQUICK.LT.1) IQUICK=1
      IF (IQUICK.GT.N) IQUICK=N
      IF (ID(IQUICK).EQ.TARGET)  RETURN
      IF (ID(IQUICK) .GT.TARGET  AND. J.NE.0 )  THEN
         IQUICK=IQUICK-J
         GO TO 10
      ELSE IF (ID(IQUICK).LT.TARGET .AND. J.NE.0 )  THEN
         IQUICK=IQUICK+J
         GO TO 10
      ENDIF
C
      I=MAX(IQUICK-2,1)
      J=MIN(I+4,N)
      DO 20 IQUICK=I,J,1
        IF (ID(IQUICK) EQ.TARGET)  RETURN
   20 CONTINUE
C
      DO 30 IQUICK=1,N
        IF (ID(IQUICK).EQ.TARGET)  RETURN
   30 CONTINUE
C
  100 WRITE(108,110) TARGET,IQUICK,ID(IQUICK)
  110 FORMAT (' TARGET:',I8,'   NOT FOUND, LAST ID(',I6,').=',I8)
      IQUICK=0
      RETURN
      END
C
C --- REAL FUNCTION VCOS
C
      REAL FUNCTION VCOS(V1,V2)
      DIMENSION V1(3),V2(3)
      VALUE (A1,A2,A3)=SQRT(A1**2+A2**2+A3**2)
      DOT   (A1,B1,C1,A2,B2,C2)=A1*A2+B1*B2+C1*C2
C
      AV1=VALUE(V1(1),V1(2),V1(3))
      AV2=VALUE(V2(1),V2(2),V2(3))
      VDOT=DOT(V1(1),V1(2),V1(3) ,V2(1),V2(2),V2(3))
      VCOS=VDOT/(AV1*AV2)
      RETURN
      END
C
C-----------------------------------------------------------------------
      SUBROUTINE MASSIN(INMASS,FMASS,MCOND,BUG,IBUG,TITLE,MAXZ,
     &                  MAXNOD,NNODE,IZ,IW,IZID,Z,NZ,IZMDMS,IZMASS,
     &                  IZLM,IZKE,IZMDAT,NDOF,NMASS,IZIDOF,IZCNST,
     &                  IZJCOS,NCOND,IZMD)
C
      LOGICAL BUG,MCOND
      INTEGER FMASS
      CHARACTER*(*) TITLE(2)
      CHARACTER NAME*12
      DIMENSION Z(0:MAXZ),NZ(0:MAXZ),MD(7)
      DATA MD/1,2,4,7,11,16,22/
C
      NAME='GLOBAL  MASS'
      IF (FMASS.EQ.1) THEN
C-------- INITIALIZE THE DIAGONAL MASS MATRIX
        IZMDMS=IZ
        IZ=IZ + NDOF+1
        CALL SKYLIN(0,1,NDOF,IBUG,TITLE,NZ(IZMDMS),
     &              6,NZ(IZLM),Z(IZKE),MD,NAME)
C
C------ INPUT THE MASS DATA
        IZMDAT=IZ
        IZ=IZ + INMASS*12
        IMD=NZ(IZMDMS+NDOF)
        IZMASS=IZ
        CALL MFORM(1,INMASS,NMASS,Z(IZMASS),NZ(IZMDMS),MD,NDOF,IMD,
     &             NNODE,MAXNOD,BUG,IBUG,NZ(IZID),NZ(IZIDOF),
     &             Z(IZCNST),NZ(IZJCOS),Z(IZMDAT),Z(IZ),NZ(IZ+21),
     &             Z(IZ+27),Z(IZ+ 63),Z(IZ+ 99),TITLE)
C
C------ - RELEASE UNUSED MASS DATA STORAGE
        IZ=IZ - (INMASS-NMASS)*12
C
C-------------------------------------
C* MODIFY MASS MATRIX FOR CONDENSATION  ---
C-------------------------------------
C THE CONDENSATION PROCESS MAY EXPAND THE MASS MATRIX'S
C SKYLINE.  THE MAXIMUM SKYLINE OF BOTH THE MASS AND
C THE STIFFNESS MATRIX IS USED.
      IF (MCOND .AND. NCOND.GT.0)THEN
        DO 310 I=0,NDOF
          NHT=NZ(IZMD+I+1)-NZ(IZMD+I)
          NSTIFF=I+1-NHT+1
  310     NZ(IZMDMS+I)=MIN( NZ(IZMDMS+I),NSTIFF)
      ENDIF
C
C-------- DETERMINE THE MASS MATRIX SKYLINE
        CALL SKYLIN(0,3,NDOF,IBUG,TITLE,NZ(IZMDMS),
     &              6,NZ(IZLM),Z(IZKE),MD,NAME)
C
C-------- GENERATE THE MASS MATRIX
        IZMASS=IZ
        IZ=IZ + NZ(IZMDMS+NDOF)
        DO 400 I=IZMASS,IZ
  400     Z(I)=0.
        IW=IZ
        IZKE=IW
        IZLM=IW+21
        CALL MFORM(2,INMASS,NMASS,Z(IZMASS),NZ(IZMDMS),MD,NDOF,IMD,
     &             NNODE,MAXNOD,BUG,IBUG,NZ(IZID),NZ(IZIDOF),
     &             Z(IZCNST),NZ(IZJCOS),Z(IZMDAT),Z(IW),NZ(IW+21),
     &             Z(IW+27),Z(IW+ 63),Z(IW+ 99),TITLE)
C
        IF (BUG) THEN
          IMD=NZ(IZMDMS+NDOF)
          CALL LMATRX(Z(IZMASS),NZ(IZMDMS),NDOF,IMD,NAME)
        ENDIF
      ENDIF
C
      RETURN
      END
C
C-----------------------------------------------------------------------
      SUBROUTINE CKSTOR(ITEMP,IW)
$INCLUDE:'ZCOMN'
C
C-------------------------------
C- CHECK STORAGE REQUIREMENTS ---
C-------------------------------
      N=MAX(IZ,IW)+ITEMP
      IF (N.LE.MAXZ) RETURN
C
      WRITE(108,10) N,MAXZ
   10 FORMAT (///1X, 49('*') /
     &   ' ***  AVAILABLE STORAGE IS EXCEEDED'          ,T48,'***'/
     &   ' ***  REQUIRED STORAGE =',I10                 ,T48,'***'/
     &   ' ***  AVAILABLE STORAGE =',I10                ,T48,'***'/
     &   ' ***  EXECUTION IS TERMINATED.'               ,T48,'***'/
     &   ' ***  RECOMPILE PROGRAM WITH INCREASED MEMORY',T48,'***'/
     &    1X,49('*')  )
C
C------- ABORT RUN
      STOP 'REQUIRED MEMORY > AVAILABLE MEMORY '
      RETURN
      END
C
C-----------------------------------------------------------------------
      SUBROUTINE MFORM(IOPT,INMASS,NMASS,MASS,MDM,MD,NDOF,IMD,
     &                 NNODE,MAXNOD,BUG,IBUG,ID,IDOF,CONST,
     &                 JTCOS,DATA,SMASS,LM,A,S,SAT,TITLE)
      DIMENSION ID(MAXNOD),JTCOS(MAXNOD),IDOF(MAXNOD,6)
      DIMENSION A(6,6),S(6,6),SAT(6,6),MDM(NDOF+1),MD(7)
      DIMENSION DATA(12,INMASS),SMASS(21),LM(6),CONST(MAXNOD,6)
      REAL      MASS(IMD)
      CHARACTER*130 ELNO
      CHARACTER*(*) TITLE(2)
      LOGICAL BUG
C
C------- - PRINT HEADER AND INPUT MASS DATA
      IF (IOPT.EQ.1) THEN
        WRITE(108,5) TITLE(1),TITLE(2)
    5 FORMAT ('1 STRUCTURE..... ',A,/'  SOLUTION... . : ',A,//
     &      '   LUMPED NODE MASSES '/' --------------------'//4X,'NODE',
     &       5X,'MX',10X,'MY',10X,'MZ',9X,'IXX', 9X,'IYY', 9X,'IZZ',
     &       9X,'IXY', 9X,'IYZ', 9X,'IYZ',5X,'  IGEN   INC')
C
        NMASS=1
C  INPUT:  NODE, PX,PY,PZ,  RX,RY,RZ,  RXY,RXZ,RYZ,  IGEN, INC
    6   READ(105,*) (DATA(I,NMASS),I=1,12)
        NODE = DATA( 1,NMASS)
        IGEN = DATA(11,NMASS)
        INC  = DATA(12,NMASS)
        IF (NODE.GT.0) THEN
            WRITE(108,7) NODE,(DATA(I,NMASS),I=2,10),IGEN,INC
            IF (NMASS.LT.INMASS) THEN
              NMASS=NMASS+1
              GO TO 6
            ENDIF
        ELSE
          NMASS=NMASS-1
        ENDIF
    7   FORMAT (2X,I6,1P,9G12.4,2I6)
        ENDIF
C
        DO 100 IMASS=1,NMASS
          NODE=DATA( 1,IMASS)
          PX  =DATA( 2,IMASS)
          PY  =DATA( 3,IMASS)
          PZ  =DATA( 4,IMASS)
          RX  =DATA( 5,IMASS)
          RY  =DATA( 6,IMASS)
          RZ  =DATA( 7,IMASS)
          RXY =DATA( 8,IMASS)
          RXZ =DATA( 9,IMASS)
          RYZ =DATA(10,IMASS)
          IGEN=DATA(11,IMASS)
          INC =DATA(12,IMASS)
C
    9   JOINT=IQUICK(NODE,ID,NNODE)
```

```
C                                                                       MFGR0550
      IF (JOINT.LE.0) THEN                                              MFGR0560
        WRITE(108,*) 'NODE #',NODE,                                     MFGR0570
     &          ' WAS NOT FOUND, CHECK INPUT, MASS IS OMITTED '        MFGR0580
        GO TO 90                                                        MFGR0590
      ENDIF                                                             MFGR0600
C                                                                       MFGR0610
      ICOS  =JTCOS(JOINT)                                               MFGR0620
C                                                                       MFGR0630
C-----> SET UP NODE MASS MATRIX IN XYZ COORDINATE SYSTEM                MFGR0640
C------ SET UP TRANSFORMATION MATRIX, A = (CONST)                       MFGR0650
      DO 10 I=1,6                                                       MFGR0660
      DO 10 J=1,6                                                       MFGR0670
   10   A(I,J)=0                                                        MFGR0680
      DO 20 I=1,6                                                       MFGR0690
   20   A(I,I)=1                                                        MFGR0700
      A(4,2)=CONST(JOINT,1)                                             MFGR0710
      A(4,3)=CONST(JOINT,2)                                             MFGR0720
      A(5,1)=CONST(JOINT,3)                                             MFGR0730
      A(5,3)=CONST(JOINT,4)                                             MFGR0740
      A(6,1)=CONST(JOINT,5)                                             MFGR0750
      A(6,2)=CONST(JOINT,6)                                             MFGR0760
C                                                                       MFGR0770
C ------ SET UP NODE MASS MATRIX                                        MFGR0780
      DO 30 I=1,6                                                       MFGR0790
      DO 30 J=I+1,6                                                     MFGR0800
   30   S(I,J)=0                                                        MFGR0810
C                                                                       MFGR0820
      S(1,1)= PX                                                        MFGR0830
      S(2,2)= PY                                                        MFGR0840
      S(3,3)= PZ                                                        MFGR0850
      S(4,4)= RX                                                        MFGR0860
      S(5,5)= RY                                                        MFGR0870
      S(6,6)= RZ                                                        MFGR0880
      S(4,5)=-RXY                                                       MFGR0890
      S(4,6)=-RXZ                                                       MFGR0900
      S(5,6)=-RYZ                                                       MFGR0910
C                                                                       MFGR0920
      DO 40 I=1,6                                                       MFGR0930
      DO 40 J=I+1,6                                                     MFGR0940
   40   S(J,I)=S(I,J)                                                   MFGR0950
C                                                                       MFGR0960
C------ TRANSFORM MASS MATRIX TO ACCOUNT FOR CONSTRAINTS                MFGR0970
      DO 50 I=1,6                                                       MFGR0980
      DO 50 J=1,6                                                       MFGR0990
        SAT(I,J)=0                                                      MFGR1000
        DO 50 K=1,6                                                     MFGR1010
   50     SAT(I,J)=SAT(I,J) + S(I,K)*A(J,K)                            MFGR1020
C                                                                       MFGR1030
      DO 60 I=1,6                                                       MFGR1040
      DO 60 J=1,6                                                       MFGR1050
        S(I,J)=0                                                        MFGR1060
        DO 60 K=1,6                                                     MFGR1070
   60     S(I,J)=S(I,J) + A(I,K)*SAT(K,J)                              MFGR1080
C                                                                       MFGR1090
C------ TRANSFER MASS MATRIX TO HALF STORAGE MODE                       MFGR1100
        IJ=0                                                            MFGR1110
        DO 70 J=1,6                                                     MFGR1120
        DO 70 I=J,1,-1                                                  MFGR1130
          IJ=IJ+1                                                       MFGR1140
   70     SMASS(IJ)=S(I,J)                                             MFGR1150
C------ INITIALIZE THE DIAGONAL MASS MATRIX                            MFGR1160
        DO 80  I=1,6                                                    MFGR1170
   80     LM(I)=IDOF(JOINT,I)                                          MFGR1180
C                                                                       MFGR1190
C                                                                       MFGR1200
      IF (IOPT.EQ.1) THEN                                               MFGR1210
C                                                                       MFGR1220
C. . . . CALL SKYLIN TO RESERVE SPACE FOR THIS NODE'S MASS              MFGR1230
          IF (BUG) THEN                                                 MFGR1240
            WRITE(ELNO,915)  NODE,(LM(I),I=1,6)                         MFGR1250
            CALL LMATRX(SMASS,MD,6,21,ELNO)                             MFGR1260
          ENDIF                                                         MFGR1270
        CALL SKYLIN(0,2,NDOF,IBUG,TITLE,MDM,6,LM,SMASS,MD,'GLOBAL MASS') MFGR1280
C                                                                       MFGR1290
      ELSE IF (IOPT.EQ.2) THEN                                          MFGR1300
          IF (BUG) THEN                                                 MFGR1310
            WRITE(ELNO,915)  NODE,(LM(I),I=1,6)                         MFGR1320
  915       FORMAT ('MASS AT NODE #',I6,' LM:',6I6)                     MFGR1330
            CALL LMATRX(SMASS,MD,6,21,ELNO)                             MFGR1340
          ENDIF                                                         MFGR1350
C                                                                       MFGR1360
          CALL FORML(MASS,MDM,NDOF,IMD,SMASS,MD,6,21,LM,1.0 )           MFGR1370
C                                                                       MFGR1380
      ENDIF                                                             MFGR1390
   90 CONTINUE                                                          MFGR1400
C                                                                       MFGR1410
      IF (IGEN.GT.0) THEN                                               MFGR1420
        IGEN=IGEN-1                                                     MFGR1430
        NODE=NODE+INC                                                   MFGR1440
        GO TO 9                                                         MFGR1450
      ENDIF                                                             MFGR1460
C                                                                       MFGR1470
  100 CONTINUE                                                          MFGR1480
C                                                                       MFGR1490
      RETURN                                                            MFGR1500
      END                                                               MFGR1510
C-----------------------------------------------------------------------FORM0010
      SUBROUTINE FORML(K,MD,NDOF,IMD,KE,MDKE,IELDOF,IMDKE,LM,FACTOR)    FORM0020
      DIMENSION MD(NDOF+1),MDKE(IELDOF+1),LM(IELDOF)                    FORM0030
      REAL K(IMD),KE(IMDKE)                                             FORM0040
C                                                                       FORM0050
C---------- . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . FORM0060
C   MATRIX STORAGE SCHEME:                                              FORM0070
C  THE STIFFNESS MATRIX (K) IS BANDED AND SYMMETRIC.                    FORM0080
C   THUS ONLY THE SKYLINE ABOVE THE MAIN DIAGONAL NEEDS TO BE STORED.   FORM0090
C   THE MATRIX K IS STORED IN A LINEAR ARRAY BY COLUMNS.                FORM0100
C   THE VALUE ON THE MAIN DIAGONAL IS STORED IN THE LINEAR ARRAY FIRST  FORM0110
C   RELATIVE ADDRESSES OF THE LINEAR ARRAY ELEMENTS CONTAINING THE      FORM0120
C   MAIN DIAGONAL TERMS ARE STORED IN MD.                               FORM0130
C   THUS: FOR K(I,J)                                                    FORM0140
C         IF I LE J (UPPER TRIANGULAR MATRIX)                           FORM0150
C         IF MD(J)+J-I < MD(J+1)                                        FORM0160
C                 IJ=MD(J)-J-I      : RELATIVE ADDRESS IF ELEMENT (I,J) FORM0170
C                 K(I,J)=K(IJ)                                         FORM0180
C           ELSE                                                        FORM0190
C                 K(I,J)=0              : (I,J) IS ABOVE THE SKYLINE    FORM0200
C                                                                       FORM0210
C   EXAMPLE: LET B BE THE 5X5 FULL SYMETRIC MATRIX BELOW...             FORM0220
C      : 1 : 3 :  :  : 12 :                                             FORM0230
C      :  : 2 : 5 :  : 11 :  NOTE. K(1,3),K(1,4) AND K(2,4) ARE ZERO.   FORM0240
C   K= :  :  : 4 : 7 : 10 :                                             FORM0250
C      :  : SYMM. : 6 :  9 :  MD= 1, 2, 4, 6, 8, 13                     FORM0260
C      :  :  :  : 8 :                                                   FORM0270
C   SAY IZMD=121,IZSTIF=200                                             FORM0280
C   MD(1)=1          K( 1)=1                                            FORM0290
C   MD(2)=2          K( 2)=2                                            FORM0300
C   MD(3)=4          K( 3)=3                                            FORM0310
C   MD(4)=6          K( 4)=4                                            FORM0320
C   MD(5)=8          K( 5)=5                                            FORM0330
C   MD(6)=13         K( 6)=6                                            FORM0340
C                    K( 7)=7                                            FORM0350
C                    K( 8)=8                                            FORM0360
C                    K( 9)=9                                            FORM0370
C                    K(10)=10                                           FORM0380
C                    K(11)=11                                           FORM0390
C                    K(12)=12                                           FORM0400
C   THUS, FOR K(3,4)  I=3, J=4                                          FORM0410
C              IJ=MD(J)+J-I = MD(4)+4-3 = 6+4-3 =7                      FORM0420
C              K(I,J)=K(IJ) = K(7) = 7  OK                              FORM0430
C   THUS, FOR B(2,4)  I=2, J=4                                          FORM0440
C              IJ=MD(J)+J I = MD(4) +4-2 = 6+4-2 =8                     FORM0450
C              MD(J+1)+MD(4+1)=8                                        FORM0460
C              SINCE IJ GE MD(J+1)                                      FORM0470
C              K(I,J)=0  (I,J) IS ABOVE THE SKYLINE. . .                FORM0480
C-----------------------------------------------------------------------FORM0490
C   THE ELEMENT STIFFNESS IS STORED IN A SIMILAR MANNER.                FORM0500
C   KE IS THE LINEAR ARRAY CONTAINING THE ELEMENT STIFFNESS             FORM0510
C   MDKE IS THE ARRAY CONTAINING THE MAIN DIAGONAL ADDRESSES FOR KE     FORM0520
C ------------------------------------------------------- . . . . . .  FORM0530
C                                                                       FORM0540
C---------- MAP ELEMENT STIFFNESS INTO GLOBAL STIFFNESS                 FORM0550
C LI    - GLOBAL DOF ASSOCIATED WITH THE I'TH ROW OF THE MEMBER STIFF.  FORM0560
C LJ    - GLOBAL DOF ASSOCIATED WITH THE I'TH ROW OF THE MEMBER STIFF   FORM0570
C IJELM - ABSOLUTE ADDRESS OF THE MEMBER STIFFNESS TERM (I,J)           FORM0580
C IJGLOB- ABSOLUTE ADDRESS OF THE GLOBAL STIFFNESS TERM (LI,LJ)         FORM0590
C                                                                       FORM0600
      DO 30 I=1,IELDOF                                                  FORM0610
        LI=LM(I)                                                        FORM0620
        DO 20 J=I,IELDOF                                                FORM0630
          LJ=LM(J)                                                      FORM0640
          IJELM =MDKE(J)+J-I                                            FORM0650
          IF (KE(IJELM) EQ.0) GO TO 20                                  FORM0660
          IF (LI.LE.LJ) THEN                                            FORM0670
              IJGLOB=MD(LJ)+LJ-LI                                       FORM0680
          ELSE                                                          FORM0690
              IJGLOB=MD(LI)+LI-LJ                                       FORM0700
          ENDIF                                                         FORM0710
          K(IJGLOB)=K(IJGLOB)+KE(IJELM) * FACTOR                        FORM0720
   20   CONTINUE                                                        FORM0730
   30 CONTINUE                                                          FORM0740
C                                                                       FORM0750
C---- NOTE: LINES COMMENTED OUT C* ARE FOR ER_ROR CHECKING, THE VALUES FORM0760
C           IN LM ARE ASSUMED TO BE CORRECT...                          FORM0770
C                                                                       FORM0780
      RETURN                                                            FORM0790
      END                                                               FORM0800
C-----------------------------------------------------------------------BML00010
      SUBROUTINE BMLOA(IELNO,PLAG,L,REL,MAXELD,NELD,NLOAD,              BML00020
     &           PEM,IELD,ELD)                                          BML00030
      REAL    L                                                         BML00040
      INTEGER REL                                                       BML00050
      LOGICAL B0,B1,B2,B3,B4,B5,BTEST,TEST,PLAG                         BML00060
      DIMENSION IELD(5,MAXELD),ELD(12,MAXELD)                           BML00070
      DIMENSION PEM(12,NLOAD),FORCE(12)                                 BML00080
      EQUIVALENCE (FORCE( 1),P1 ),(FORCE( 2),P2 ),(FORCE( 3),P3 )       BML00090
      EQUIVALENCE (FORCE( 4),P4 ),(FORCE( 5),P5 ),(FORCE( 6),P6 )       BML00100
      EQUIVALENCE (FORCE( 7),P7 ),(FORCE( 8),P8 ),(FORCE( 9),P9 )       BML00110
      EQUIVALENCE (FORCE(10),P10),(FORCE(11),P11),(FORCE(12),P12)       BML00120
C                                                                       BML00130
      PLAG=.FALSE.                                                      BML00140
      B0=BTEST(REL,0)                                                   BML00150
      B1=BTEST(REL,1)                                                   BML00160
      B2=BTEST(REL,2)                                                   BML00170
      B3=BTEST(REL,3)                                                   BML00180
      B4=BTEST(REL,4)                                                   BML00190
      B5=BTEST(REL,5)                                                   BML00200
C                                                                       BML00210
      IF (MAXELD LE.0) RETURN                                           BML00220
C                                                                       BML00230
C------- LOOP FOR EACH LOAD INPUT                                       BML00240
      DO 1000 N=1,NELD                                                  BML00250
C                                                                       BML00260
C   .... DETERMINE IF MEMBER IS AFFECTED BY THIS LOAD                   BML00270
          TEST=.FALSE.                                                  BML00280
          IEL=IELD(2,N)                                                 BML00290
          IGEN=IELD(3,N)                                                BML00300
          INC=IELD(4,N)                                                 BML00310
          IF (IEL EQ.IELNO) TEST=.TRUE.                                 BML00320
          IF ((.NOT.TEST) .AND. IGEN.GT.0 .AND. INC.NE.0) THEN          BML00330
             IRMD=MOD(IELNO-IEL,INC)                                    BML00340
             JGEN=(ELNO-IEL)/INC                                        BML00350
             IF (IRMD.EQ.0 .AND. JGEN.LE.IGEN .AND. JGEN.GE.0)          BML00360
     &          TEST=.TRUE.                                             BML00370
          ENDIF                                                         BML00380
          IF (.NOT. TEST) GO TO 1000                                    BML00390
C                                                                       BML00400
C.............. ZERO PEM                                               BML00410
          IF ( NOT. PLAG) THEN                                          BML00420
             DO 10 I=1,12                                               BML00430
             DO 10 J=1,NLOAD                                            BML00440
   10          PEM(I,J)=0                                               BML00450
             PLAG=.TRUE.                                                BML00460
          ENDIF                                                         BML00470
C                                                                       BML00480
C.............. ZERO FORCE                                             BML00490
          DO 15 I=1,12                                                  BML00500
   15        FORCE(I)=0                                                 BML00510
C                                                                       BML00520
C------  DETERMINE LOAD CODES...                                        BML00530
          KODE=IELD(5,N)                                                BML00540
          KDOF=MOD(KODE,100)                                            BML00550
          KGRP=KODE-KDOF                                                BML00560
          LOAD=IELD(1,N)                                                BML00570
C                                                                       BML00580
C                                                                       BML00590
C.......  CONCENTRATED LOADING                                          BML00600
          IF (KGRP EQ.100) THEN                                         BML00610
            P=ELD(1,N)                                                  BML00620
            R=ELD(2,N)                                                  BML00630
            IF (KDOF EQ.1) THEN                                         BML00640
              F1=-(1-R)*P                                               BML00650
              F7=-R*P                                                   BML00660
            ELSE IF (KDOF.EQ.2) THEN                                    BML00670
              P6 =-P*R*L*(1-R)**2                                       BML00680
              P12= P*(1-R)*L*(R)**2                                     BML00690
              CALL FEMREL(P6 ,P12,B3 ,B0 ,0,IELNO)                      BML00700
              P8=-(-P6-P12-P*R*L )/L                                    BML00710
              P2=-P-P8                                                  BML00720
            ELSE IF (KDOF.EQ.3) THEN                                    BML00730
              P5 = P*R*L*(1-R)**2                                       BML00740
              P11=-P*(1-R)*L*(R)**2                                     BML00750
              CALL FEMREL(P5 ,P11,B4 ,B1 ,0,IELNO)                      BML00760
              P9 =-(P5+P11*P*R*L)/L                                     BML00770
              P3 =-P-P9                                                 BML00780
            ELSE IF (KDOF EQ.4) THEN                                    BML00790
              P4 =-P*(1 R)                                              BML00800
              P10=-P*R                                                  BML00810
              CALL FEMREL(P4 ,P10,B5 ,B2 ,1,IELNO)                      BML00820
            ELSE IF (KDOF.EQ.5) THEN                                    BML00830
              P5 =-P*(1 - 4*R + 3*R**2)                                 BML00840
              P11= P*( 2*R - 3*R**2)                                    BML00850
              CALL FEMREL(P5 ,P11,B4 ,B1 ,0,IELNO)                      BML00860
              P9=-( P5+P11*P)/L                                         BML00870
              P3=-P9                                                    BML00880
            ELSE IF (KDOF.EQ.6) THEN                                    BML00890
              P6 =-P*(1 - 4*R + 3*R**2)                                 BML00900
              P12= P*( 2*R - 3*R**2)                                    BML00910
              CALL FEMREL(P6 ,P12,B3 ,B0 ,0,IELNO)                      BML00920
              P8 =-(P6+P12*P)/L                                         BML00930
              P2 = -P8                                                  BML00940
            ENDIF                                                       BML00950
C                                                                       BML00960
C.  . . UNIFORM LOADING                                                 BML00970
            ELSE IF (KGRP.EQ.200) THEN                                  BML00980
              W=ELD(1,N)                                                BML00990
              IF (KDOF EQ 1) THEN                                       BML01000
                P1=-W*L/2                                               BML01010
                F7=-W*L/2                                               BML01020
              ELSE IF (KDOF.EQ.2) THEN                                  BML01030
                P6 =-W*L*L/12                                           BML01040
                P12= W*L*L/12                                           BML01050
                CALL FEMREL(P6 ,P12,B3 ,B0 ,0,IELNO)                    BML01060
                P8=(-P6-P12)/L  -W*L/2                                  BML01070
                P2=-W*L-P8                                              BML01080
              ELSE IF (KDOF.EQ 3) THEN                                  BML01090
                P5 = W*L*L/12                                           BML01100
                P11=-W*L*L/12                                           BML01110
                CALL FEMREL(P5 ,P11,B4 ,B1 ,0,IELNO)                    BML01120
                P9 =(P5-P11)/L  - W*L/2                                 BML01130
                P3 =-W*L-P9                                             BML01140
              ELSE IF (KDOF.EQ.4) THEN                                  BML01150
```

```
            P4 =-W*L/2                                                  BMLO1160
            P10=-W*L/2                                                  BMLO1170
            CALL FEMREL(P4 ,P10,B5 ,B2 ,1,IELNO)                        BMLO1180
          ELSE                                                          BMLO1190
            WRITE(108,900) IELNO,' UNIFORM MY OR UNIFORM MZ'            BMLO1200
          ENDIF                                                         BMLO1210
C                                                                       BMLO1220
C..... . USER INPUT FIXED END FORCE                                     BMLO1230
        ELSE IF (KGRP.EQ.400) THEN                                      BMLO1240
          DO 400 I=1,12                                                 BMLO1250
  400     FORCE(I)= ELD(I,N)                                            BMLO1260
C                                                                       BMLO1270
C....... INVALID LOAD GROUP                                             BMLO1280
        ELSE                                                            BMLO1290
          WRITE(108,900) IELNO,'INVALID LOAD GROUP'                     BMLO1300
        ENDIF                                                           BMLO1310
C                                                                       BMLO1320
C..... .. TRANSFER BEAM LOADS TO FIXED END FORCES  ..                   BMLO1330
          DO 800 I=1,12                                                 BMLO1340
  800     FEM(I,LOAD)=FEM(I,LOAD)-FORCE(I)                              BMLO1350
C                                                                       BMLO1360
C                                                                       BMLO1370
 1000 CONTINUE                                                          BMLO1380
C                                                                       BMLO1390
  900 FORMAT('  ER*,'FOR CALCULATING FIXED END FORCES FOR ELEMENT #',   BMLO1400
     &       I6,' ',A,' IS NOT AVAILABLE')                              BMLO1410
      RETURN                                                            BMLO1420
      END                                                               BMLO1430
C                                                                       ROTX0010
C-----------------------------------------------------------------------ROTX0020
      SUBROUTINE ROTXYZ(VX,VY,COSINE,CT,EX,EY,EZ,CTXYZ,GLOXYZ,NROW)     ROTX0030
      DIMENSION VX(3),VY(3),VZ(3),COSINE(3,3),CT(3,3),                  ROTX0040
     &          XYZ(3,3),CTXYZ(3,3),CE(3,3),GLOXYZ(NROW,6)              ROTX0050
C.....  .....  ..... ..... ..... ..... ..... ..... ..... ..... .....  ---ROTX0060
C                                                                       ROTX0070
C     THIS SUBROUTINE PREFORMS SEVERAL FUNCTIONS:                       ROTX0080
C     1) CALCULATE LOCAL COORDINATE SYS COSINE MATRIX.                  ROTX0090
C     2) CALCULATE ROTATION MATRIX BETWEEN LOCAL AND JOINT COORD (CT)   ROTX0100
C     3) CALCULATE TRANSFORMATION FOR GLOBAL CONSTRAINTS AND MEMBER     ROTX0110
C        END DEPTH- STORE IN (CTXYZ)                                    ROTX0120
C                                                                       ROTX0130
C--- INPUT VARIABLES-----------------------------------------------------ROTX0140
C     COSINE - INPUT JOINT COORDINATE DIRECTION COSINE                  ROTX0150
C     VX     - INPUT X-DIRECTION OF ELEMENT COORDINATE SYS              ROTX0160
C     VY     - INPUT VECTOR IN THE ELEMENT COORDINATE SYS  XY PLANE.     ROTX0170
C     GLOXYZ - GLOBAL CONSTRAINT TRANSFORMATION COIEFFICIENTS           ROTX0180
C     EX     - MEMBER END ECCENTRICITY IN THE LOCAL X AXIS.             ROTX0190
C     EY     - MEMBER END ECCENTRICITY IN THE LOCAL Y AXIS.             ROTX0200
C     EZ     - MEMBER END ECCENTRICITY IN THE LOCAL Z AXIS              ROTX0210
C                                                                       ROTX0220
C--- OUTPUT VARIABLES----------------------------------------------------ROTX0230
C     VX     - UNIT VECTOR, X-DIRECTION OF ELEMENT COORDINATE SYS       ROTX0240
C     VY     - UNIT VECTOR, X-DIRECTION OF ELEMENT COORDINATE SYS       ROTX0250
C     VZ     - UNIT VECTOR, X-DIRECTION OF ELEMENT COORDINATE SYS       ROTX0260
C     CT     - ROTATED COORDINATE TRANSFORMATION                        ROTX0270
C     CTXYZ  - CONSTRAINT AND MEMBER END DEPTH TRANSFORMATION           ROTX0280
C                                                                       ROTX0290
C     | FX |                          | FX |                           ROTX0300
C     | FY |                          | FY |                           ROTX0310
C     | FZ |        |   CT   |  0 |   | FZ |                           ROTX0320
C     | ---|      = |--------|----|   |----|                           ROTX0330
C     | MX |        | CTXYZ  | CT |   | MX |                           ROTX0340
C     | MZ |        |        |    |   | MY |                           ROTX0350
C     | MZ |        |        |    |   | MZ |                           ROTX0360
C          GLOBAL                          ELEMENT COORD.               ROTX0370
C                                                                       ROTX0380
C-----------------------------------------------------------------------ROTX0390
C                                                                       ROTX0400
      CROSSI(A1,B1,C1,A2,B2,C2)=B1*C2-B2*C1                             ROTX0410
      CROSSJ(A1,B1,C1,A2,B2,C2)=C1*A2-C2*A1                             ROTX0420
      CROSSK(A1,B1,C1,A2,B2,C2)=A1*B2-A2*B1                             ROTX0430
      VALUE (A1,A2,A3)=SQRT(A1**2+A2**2+A3**2)                          ROTX0440
      DOT   (A1,B1,C1,A2,B2,C2)=A1*A2+B1*B2+C1*C2                       ROTX0450
C                                                                       ROTX0460
C ------- CHECK ANGLE BETWEEN VX AND VY                                 ROTX0470
      C1=VCOS(VX,VY)                                                    ROTX0480
      IF(ABS(C1).GT.999) WRITE(108,*) 'VY IS PARAELL TO VX, REINPUT VY' ROTX0490
C                                                                       ROTX0500
C ------- CHECK VY, IF |VY|.EQ.0 , SET VY IN GLOBAL YZ PLANE..          ROTX0510
      AVY=VALUE(VY(1),VY(2),VY(3))                                      ROTX0520
      IF (AVY.EQ.0) THEN                                                ROTX0530
            VZ(1)=1.0                                                   ROTX0540
            VZ(2)= .0                                                   ROTX0550
            VZ(3)= .0                                                   ROTX0560
      ELSE                                                              ROTX0570
C ------- CALCULATE VZ, IF VY WAS CORRECTLY INPUT                       ROTX0580
            VZ(1)=CROSSI(VX(1),VX(2),VX(3),VY(1),VY(2),VY(3))           ROTX0590
            VZ(2)=CROSSJ(VX(1),VX(2),VX(3),VY(1),VY(2),VY(3))           ROTX0600
            VZ(3)=CROSSK(VX(1),VX(2),VX(3),VY(1),VY(2),VY(3))           ROTX0610
      ENDIF                                                             ROTX0620
C                                                                       ROTX0630
C ------- RECALCULATE VY TO INSURE VX,VY AND VZ ARE ORTHOGONAL...       ROTX0640
      VY(1)=CROSSI(VZ(1),VZ(2),VZ(3),VX(1),VX(2),VX(3))                 ROTX0650
      VY(2)=CROSSJ(VZ(1),VZ(2),VZ(3),VX(1),VX(2),VX(3))                 ROTX0660
      VY(3)=CROSSK(VZ(1),VZ(2),VZ(3),VX(1),VX(2),VX(3))                 ROTX0670
C                                                                       ROTX0680
C ------- CHECK ANGLE BETWEEN VX AND VY                                 ROTX0690
      C1=VCOS(VX,VY)                                                    ROTX0700
      IF(ABS(C1).GT.999) WRITE(108,*) 'VY IS PARAELL TO VX, REINPUT VY' ROTX0710
C                                                                       ROTX0720
C ------- NORMALIZE VX,VY,VZ                                           ROTX0730
      AVX=VALUE(VX(1),VX(2),VX(3))                                      ROTX0740
      AVY=VALUE(VY(1),VY(2),VY(3))                                      ROTX0750
      AVZ=VALUE(VZ(1),VZ(2),VZ(3))                                      ROTX0760
      DO 10 I=1,3                                                       ROTX0770
        VX(I)=VX(I)/AVX                                                 ROTX0780
        VY(I)=VY(I)/AVY                                                 ROTX0790
        VZ(I)=VZ(I)/AVZ                                                 ROTX0800
        CE(1,I)=VX(I)                                                   ROTX0810
        CE(2,I)=VY(I)                                                   ROTX0820
   10   CE(3,I)=VZ(I)                                                   ROTX0830
C                                                                       ROTX0840
C --- --- CALC THE DIRECTION COSINE MATRIX                             ROTX0850
      DO 15 I=1,3                                                       ROTX0860
      DO 15 J=1,3                                                       ROTX0870
          CT(I,J)=0                                                     ROTX0880
          DO 15 K=1,3                                                   ROTX0890
   15       CT(I,J)=CT(I,J)+COSINE(I,K)*CE(J,K)                        ROTX0900
C                                                                       ROTX0910
C ------- REMOVE THE FUZZ FROM THE COSINE MATRIX                        ROTX0920
      DO 20 I=1,3                                                       ROTX0930
      DO 20 J=1,3                                                       ROTX0940
   20   IF (ABS(CT(I,J)) LT  0.0001) CT(I,J)=0.                        ROTX0950
C                                                                       ROTX0960
C ------- SET UP THE GLOBAL CONSTRAINT MATRIX                           ROTX0970
        XYZ(1,1)= .0                                                    ROTX0980
        XYZ(1,2)=GLOXYZ(1,1)                                            ROTX0990
        XYZ(1,3)=GLOXYZ(1,2)                                            ROTX1000
        XYZ(2,1)=GLOXYZ(1,3)                                            ROTX1010
        XYZ(2,2)= .0                                                    ROTX1020
        XYZ(2,3)=GLOXYZ(1,4)                                            ROTX1030
        XYZ(3,1)=GLOXYZ(1,5)                                            ROTX1040
        XYZ(3,2)=GLOXYZ(1,6)                                            ROTX1050
        XYZ(3,3)= .0                                                    ROTX1060
C                                                                       ROTX1070
      DO 25 I=1,3                                                       ROTX1080
        DO 25 J=1,3                                                     ROTX1090
          CTXYZ(I,J)=0                                                  ROTX1100
          DO 25 K=1,3                                                   ROTX1110
   25       CTXYZ(I,J)=CTXYZ(I,J) + XYZ(I,K)*CT(K,J)                   ROTX1120
C                                                                       ROTX1130
C ------- SKIP THE MEMBER END ECCENTRICITY MATRIX IF EX=EY=EZ=0         ROTX1140
      IF (EX.EQ.0 .AND. EY.EQ 0 .AND. EZ.EQ.0) RETURN                   ROTX1150
C                                                                       ROTX1160
C ------- CALC THE MEMBER END ECCENTRICITY MATRIX                       ROTX1170
        XYZ(1,1)= .0                                                    ROTX1180
        XYZ(1,2)=-EZ                                                    ROTX1190
        XYZ(1,3)= EY                                                    ROTX1200
        XYZ(2,1)= EZ                                                    ROTX1210
        XYZ(2,2)= .0                                                    ROTX1220
        XYZ(2,3)=-EX                                                    ROTX1230
        XYZ(3,1)=-EY                                                    ROTX1240
        XYZ(3,2)= EX                                                    ROTX1250
        XYZ(3,3)= .0                                                    ROTX1260
C                                                                       ROTX1270
C ------- CALC THE MEMBER END ECCENTRICITY MATRIX                       ROTX1280
      DO 30 I=1,3                                                       ROTX1290
        DO 30 J=1,3                                                     ROTX1300
          DO 30 K=1,3                                                   ROTX1310
   30       CTXYZ(I,J)=CTXYZ(I,J) + CT(I,K)*XYZ(K,J)                   ROTX1320
C                                                                       ROTX1330
      RETURN                                                            ROTX1340
      END                                                               ROTX1350
C                                                                       ROTX1360
C ..... . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . FEMR0010
C                                                                       FEMR0020
      SUBROUTINE FEMREL(MA,MB,RELA,RELB,ITORQ,IELNO)                    FEMR0030
      REAL    MA,MB                                                     FEMR0040
      LOGICAL RELA,RELB                                                 FEMR0050
C                                                                       FEMR0060
      IF (ITORQ.EQ.0) THEN                                              FEMR0070
          IF (RELA .AND. RELB) THEN                                     FEMR0080
            MA=0                                                        FEMR0090
            MB=0                                                        FEMR0100
          ELSE IF (RELB) THEN                                           FEMR0110
            MA = MA-MB/2                                                FEMR0120
            MB=0                                                        FEMR0130
          ELSE IF (RELA) THEN                                           FEMR0140
            MB= MB-MA/2                                                 FEMR0150
            MA =0                                                       FEMR0160
          ENDIF                                                         FEMR0170
      ELSE                                                              FEMR0180
          IF (RELA .AND  RELB) THEN                                     FEMR0190
            WRITE(108,900) IELNO,'TORSION WITH BOTH ENDS RELEASED'      FEMR0200
            MA=0                                                        FEMR0210
            MB=0                                                        FEMR0220
          ELSE IF (RELB) THEN                                           FEMR0230
            MA=MA+MB                                                    FEMR0240
            MB=0                                                        FEMR0250
          ELSE IF (RELA) THEN                                           FEMR0260
            MB=MB+MA                                                    FEMR0270
            MA =0                                                       FEMR0280
          ENDIF                                                         FEMR0290
      ENDIF                                                             FEMR0300
C                                                                       FEMR0310
C                                                                       FEMR0320
  900 FORMAT('  ER*,'FOR CALCULATING FIXED END FORCES FOR ELEMENT #',   FEMR0330
     &  I6,' ',A,' IS NOT AVAILABLE')                                   FEMR0340
      RETURN                                                            FEMR0350
      END                                                               FEMR0360
C-----------------------------------------------------------------------LMAT0010
      SUBROUTINE LMATRX(A,MD,IC,IMD,NAME)                               LMAT0020
      DIMENSION MD(IC+1), A(IMD)                                        LMAT0030
      CHARACTER*(*) NAME                                                LMAT0040
      CHARACTER*20  NUMB,ROW(6)                                         LMAT0050
      LOGICAL PT,DMP                                                    LMAT0060
      IJI(I,J)=MD(J)+J-I                                                LMAT0070
C                                                                       LMAT0080
      CALL GETINT(NAME,'DUMP=',IDUMP,0, TRUE.,.FALSE.)                  LMAT0090
      IF (IDUMP NE.0)  THEN                                             LMAT0100
          DMP=.TRUE.                                                    LMAT0110
      ELSE                                                              LMAT0120
          DMP=.FALSE.                                                   LMAT0130
      ENDIF                                                             LMAT0140
          DO 1 I=1,6                                                    LMAT0150
    1       ROW(I)=' '                                                  LMAT0160
          IR=IC                                                         LMAT0170
          WRITE(108,*) ' '                                              LMAT0180
          WRITE(108,*) 'PRINTING MATRIX: ',NAME                         LMAT0190
          DO 50 ICG=1,IC,6                                              LMAT0200
            IF (ICG.GT.1) WRITE(108,*) ' '                              LMAT0210
            K=MIN((ICG+5),IC)                                           LMAT0220
            WRITE(108,10) (J,J=ICG,K)                                   LMAT0230
            DO 20 I=1,IR                                                LMAT0240
              L=0                                                       LMAT0250
              PT=.FALSE.                                                LMAT0260
              DO 15 J=ICG,K                                             LMAT0270
                IJ=IJI(I,J)                                             LMAT0280
                IF (IJ.GE.MD(J+1) .OR. I.GT.J) THEN                     LMAT0290
                  NUMB=' '                                              LMAT0300
                ELSE                                                    LMAT0310
                  PT= TRUE.                                             LMAT0320
                  WRITE(NUMB,16) A(IJ)                                  LMAT0330
                  IF (DMP.AND.A(IJ).NE.0.0)                             LMAT0340
     &              WRITE(IDUMP,*) I,J,A(IJ),NAME                       LMAT0350
                ENDIF                                                   LMAT0360
                L=L+1                                                   LMAT0370
                ROW(L)=NUMB                                             LMAT0380
   15         IF (PT) WRITE(108,30)  I,(ROW(L),L=1,6)                  LMAT0390
   50       CONTINUE                                                    LMAT0400
            RETURN                                                      LMAT0410
   10 FORMAT(T10,6('|',I,' COLUMN',I5,'-------'),'|')                   LMAT0420
   16 FORMAT(G20.6)                                                     LMAT0430
   30 FORMAT(' ROW',I5,6A)                                              LMAT0440
      END                                                               LMAT0450
C-----------------------------------------------------------------------WMAT0010
      SUBROUTINE WMATRX(A,IR,IC,NAME)                                   WMAT0020
      DIMENSION A(IR,IC)                                                WMAT0030
      CHARACTER*(*) NAME                                                WMAT0040
      LOGICAL DMP                                                       WMAT0050
C                                                                       WMAT0060
      CALL GETINT(NAME,'DUMP=',IDUMP,0,.TRUE.,.FALSE.)                  WMAT0070
      IF (IDUMP NE.0)  THEN                                             WMAT0080
          DMP= TRUE.                                                    WMAT0090
      ELSE                                                              WMAT0100
          DMP=.FALSE                                                    WMAT0110
      ENDIF                                                             WMAT0120
C                                                                       WMAT0130
      WRITE(108,*) ' '                                                  WMAT0140
      WRITE(108,*) 'PRINTING MATRIX: ',NAME                             WMAT0150
      DO 50 ICG=1,IC,6                                                  WMAT0160
        IF (ICG.GT.1) WRITE(108,*) ' '                                  WMAT0170
        K=MIN((ICG+5),IC)                                               WMAT0180
        WRITE(108,10) (J,J=ICG,K)                                       WMAT0190
        DO 20 I=1,IR                                                    WMAT0200
          IF (DMP) THEN                                                 WMAT0210
            DO 15 J=ICG,K                                               WMAT0220
   15       IF (A(I,J).NE.0) WRITE(IDUMP,*) I,J,A(I,J),NAME             WMAT0230
          ENDIF                                                         WMAT0240
   20     WRITE(108,30)  I,(A(I,J),J=ICG,K)                            WMAT0250
   50 CONTINUE                                                          WMAT0260
      RETURN                                                            WMAT0270
   10 FORMAT(' COLUMN',T17,I5,5(15X,I5))                                WMAT0280
   30 FORMAT(' ROW',I4,6G20.6)                                          WMAT0290
      END                                                               WMAT0300
```

```
C*--------------------------------------------------------------------STRU0010
      SUBROUTINE STRUCT                                                 STRU0020
      DIMENSION RINPUT(100),GINPUT(100),DUCT(3),EXCR(3)                 STRU0030
      CHARACTER*80 NAME,TYPE                                            STRU0040
      DIMENSION SE(100)                                                 STRU0050
      LOGICAL FIRST,PT,HEAD,TEST,AXIAL,HEAD2,BTEST                      STRU0060
$INCLUDE.'ZCOMN'                                                        STRU0070
      WRITE(108,5) TITLE(1),TITLE(2)                                    STRU0080
    5 FORMAT ('1 STRUCTURE .....: ',A,/                                 STRU0090
     &        '  SOLUTION.....: ',A,/)                                  STRU0100
      WRITE(108,6)                                                      STRU0110
    6 FORMAT (' *** PROGRAM FEM ***          DOUBLE PRECISION VERSION'//)STRU0120
C                                                                       STRU0130
C ... .. INPUT # OF NODES, # OF SUPTS, # OF CONST., AND SCALE FACTOR     STRU0140
      READ(105,*) NNODE,NCOS,NSUPT,NCOND,NCONST,SCALE                   STRU0150
      MAXNOD=NNODE                                                      STRU0160
C                                                                       STRU0170
      NLOAD=1                                                           STRU0180
C----------------------                                                 STRU0190
C- INPUT NODE DATA -----                                                STRU0200
C----------------------                                                 STRU0210
C       ---- NODE DATA STORAGE ---------------------                    STRU0220
C            NZ(IZID  ) - ID NUMBERS FOR EACH NODE                      STRU0230
C            NZ(IZIDOF) - DEGREE OF FREEDOM FOR EACH NODE               STRU0240
C             Z(IZCORD) - GLOBAL COORDINATE OF EACH NODE (X,Y,Z)        STRU0250
C             Z(IZCNST) - GLOBAL CONSTRAINT MATRIX FOR EACH NODE        STRU0260
C            NZ(IZWK  ) - TEMPORARY STORAGE                             STRU0270
C                                                                       STRU0280
      IZID   = IZ                                                       STRU0290
      IZ     = IZ+MAXNOD                                                STRU0300
      IZIDOF = IZ                                                       STRU0310
      IZ     = IZ+MAXNOD*6                                              STRU0320
      IZCORD = IZ                                                       STRU0330
      IZ     = IZ+MAXNOD*3                                              STRU0340
      IZCOS  = IZ                                                       STRU0350
      IZ     = IZ+MAXNOD*9                                              STRU0360
      IZCNST = IZ                                                       STRU0370
      IZ     = IZ+MAXNOD*6                                              STRU0380
      IZJFLG = IZ                                                       STRU0390
      IZ     = IZ+MAXNOD                                                STRU0400
      IZJCOS = IZ                                                       STRU0410
      IZ     = IZ+MAXNOD                                                STRU0420
      IZWK   = IZ                                                       STRU0430
      IZWK2  = IZ+MAXNOD*6                                              STRU0440
C                                                                       STRU0450
      CALL NODDOF(NDOF,NCOND,NFREE,NREST,MAXNOD,NNODE,NCOS,SCALE,NSUPT, STRU0460
     &            NCONST,NZ(IZID),NZ(IZIDOF),NZ(IZWK2),Z(IZCORD),Z(IZCOS), STRU0470
     &            Z(IZCNST),NZ(IZJFLG),NZ(IZJCOS),BTEST(IBUG,0),NZ(IZWK) ) STRU0480
C                                                                       STRU0490
C------ SET UP TEMPORARY ARRAY FOR MAIN DIAGONAL INDICIES...           STRU0500
        IZMD =IZ                                                        STRU0510
        IZ   =IZ+NDOF+1                                                 STRU0520
        CALL SKYLIN(0,1,NDOF,IBUG,TITLE,NZ(IZMD),                       STRU0530
     &              IELDOF,NZ(IZLM),Z(IZKE),MD,'GLOBAL STIFFNESS')      STRU0540
C                                                                       STRU0550
C                                                                       STRU0560
C----------------------                                                 STRU0570
C- INPUT MATERIAL DATA -                                                STRU0580
C----------------------                                                 STRU0590
       READ(105,*) NMAT                                                 STRU0600
C      ---- MATERIAL DATA STORAGE ---------------  ----                STRU0610
C           NZ(IZMAT      )-         # OF MATERIAL PORPERTIES (NMAT)   STRU0620
C           NZ(IZMAT+    1)-         ABSOLUTE ADDRESS OF MATL #1       STRU0630
C           NZ(IZMAT+    2)-         ABSOLUTE ADDRESS OF MATL #2       STRU0640
C           NZ(IZMAT+    3)-         ABSOLUTE ADDRESS OF MATL #3       STRU0650
C                    .                        .                      STRU0660
C                    .                        .                      STRU0670
C                    .                        .                      STRU0680
C           NZ(IZMAT+NMAT )-         ABSOLUTE ADDRESS FO MATL # NMAT   STRU0690
C            Z(NZ(IZMAT+1))-        MATERIAL DATA FOR MATERIAL #1      STRU0700
C                    .                        .                      STRU0710
C                    .                        .                      STRU0720
C                    .                        .                      STRU0730
C            Z(NZ(IZMAT+2))-        MATERIAL DATA FOR MATERIAL #2      STRU0740
C                    .                        .                      STRU0750
C                    .                        .                      STRU0760
C                    .                        .                      STRU0770
C                                                                    STRU0780
C                                                                       STRU0790
      IOPT=1                                                            STRU0800
      IZMAT =IZ                                                         STRU0810
      IZ    =IZ +NMAT+1                                                 STRU0820
      NZ(IZMAT)=NMAT                                                    STRU0830
      PT=.TRUE.                                                         STRU0840
      FIRST=.TRUE.                                                      STRU0850
      HEAD=.TRUE.                                                       STRU0860
      MAT  =0                                                           STRU0870
      LSTTYP=0                                                          STRU0880
C------------- LOOP FOR EACH MATERIAL ------------------------------    STRU0890
  110 MAT   =MAT   +1                                                   STRU0900
        READ(105,*) TYPE                                                STRU0910
C                                                                       STRU0920
        IF (TEST(TYPE,'3D-BEAM',.FALSE.)) THEN                          STRU0930
           MATTYP= 1                                                    STRU0940
        ELSE IF (TEST(TYPE,'AXLMOD' ,.FALSE.)) THEN                     STRU0950
           MATTYP= 2                                                    STRU0960
        ELSE IF (TEST(TYPE,'BEND1'  ,.FALSE.)) THEN                     STRU0970
           MATTYP= 3                                                    STRU0980
        ELSE IF (TEST(TYPE,'SHEAR1' ,.FALSE.)) THEN                     STRU0990
           MATTYP= 4                                                    STRU1000
        ELSE IF (TEST(TYPE,'RGMBERG',.FALSE.)) THEN                     STRU1010
           MATTYP= 5                                                    STRU1020
        ELSE IF (TEST(TYPE,'TAKEDA' ,.FALSE.)) THEN                     STRU1030
           MATTYP= 6                                                    STRU1040
        ELSE IF (TEST(TYPE,'ELSPLS' ,.FALSE.)) THEN                     STRU1050
           MATTYP= 7                                                    STRU1060
        ELSE IF (TEST(TYPE,'BILINEAR',.FALSE.)) THEN                    STRU1070
           MATTYP= 7                                                    STRU1080
        ELSE IF (TEST(TYPE,'BRACE',.FALSE.)) THEN                       STRU1090
           MATTYP= 8                                                    STRU1100
        ELSE IF (TEST(TYPE,'LONG_OWJG',.FALSE.)) THEN                   STRU1110
           MATTYP= 9                                                    STRU1120
        ELSE IF (TEST(TYPE,'IA_BILN',.FALSE.)) THEN                     STRU1130
           MATTYP= 10                                                   STRU1140
        ELSE IF (TEST(TYPE,'SHORT_OWJG',.FALSE.)) THEN                  STRU1150
           MATTYP= 11                                                   STRU1160
        ELSE IF (TEST(TYPE,'STABILITY',.FALSE.)) THEN                   STRU1170
           MATTYP= 12                                                   STRU1180
        ELSE IF (TEST(TYPE,'LOCBIL',.FALSE.)) THEN                      STRU1190
           MATTYP= 13                                                   STRU1200
        ELSE IF (TEST(TYPE,'SHEAR2',.FALSE.)) THEN                      STRU1210
           MATTYP= 14                                                   STRU1220
C                                                                       STRU1230
C... ...    TO ADD ADDITIONAL MATERIALS, COPY THE NEXT TWO LINES        STRU1240
C       ELSE IF (TEST(TYPE,'(1)',.FALSE.)) THEN                         STRU1250
C          MATTYP= (4)                                                  STRU1260
C                  CHANGE THE FOLLOWING VALUES                          STRU1270
C                  (1) YOUR MATERIAL NAME                               STRU1280
C..                (2) MATERIAL ID NUMBER = XX                          STRU1290
C..                ADD CALL MATXX TO SUBPROGRAM   MATLIB                STRU1300
C .                ADD SUBPROGRAM MATXX                                 STRU1310
C ..... .                                                               STRU1320
        ELSE IF (TEST(TYPE,'END', FALSE.)) THEN                         STRU1330
           NMAT=MAT-1                                                   STRU1340
           NZ(IZMAT)=NMAT                                               STRU1350
           GO TO 120                                                    STRU1360
        ELSE                                                            STRU1370
           WRITE(108,*) 'INVALID MATERIAL TYPE. ', HEAD                 STRU1380
           GO TO 110                                                    STRU1390
        ENDIF                                                           STRU1400
C                                                                       STRU1410
C    . RESERVE STORAGE                                                  STRU1420
       NZ(IZMAT+MAT   )=IZ                                              STRU1430
       IZWK=IZ+200                                                      STRU1440
C                                                                       STRU1450
C...  ...CALL MATERIAL LIBRARY TO INITIALIZE MATERIAL DATA              STRU1460
       BACKSPACE (105)                                                  STRU1470
       CALL MATLIB(IOPT,LSTTYP,HEAD,ESSE,EPSE,DUCT,EXCR,                STRU1480
     &             P,PL,D,DL,V,VL,LELEM,LMAT,MAT,MATTYP,SE, 1,DAMAGE)   STRU1490
C                                                                       STRU1500
C.. ....RESERVE STORAGE                                                 STRU1510
       IZ=IZ+LMAT+1                                                     STRU1520
C                                                                       STRU1530
     IF (MAT   .LT.NMAT) GO TO 110                                      STRU1540
  120 CONTINUE                                                          STRU1550
C                                                                       STRU1560
C----------------------------------                                     STRU1570
C- INPUT GEOMETRIC STIFFNESS DATA ---                                   STRU1580
C----------------------------------                                     STRU1590
C   Z(IZKGDT+0) - VERTICAL GROUND ACCELERATION                          STRU1600
C NZ(IZKGDT+1) - TYPE OF AXIAL LOAD USED                                STRU1610
C                - 0, GEOMETRIC STIFFNESS IS NOT CONSIDERED             STRU1620
C                - 1, LOAD= F(INPUT)*(1.0 + ACCEL)                      STRU1630
C                - 2, PREVIOUS LOAD STEP'S AXIAL LOAD IS USED           STRU1640
C NZ(IZKGDT+2) - TYPE OF ELEMENT GEOMETRIC STIFFNESS TO BE USED         STRU1650
C                - 0, GEOMETRIC STIFFNESS IS NOT CONSIDERED             STRU1660
C                - 1, LUMPED FORMULATION                                STRU1670
C                - 2, CONSISTENT FORMULATION                            STRU1680
C NZ(IZKGDT+3) - FORM OF SOLUTION                                       STRU1690
C                - 0, GEOMETRIC STIFFNESS IS NOT CONSIDERED             STRU1700
C                - 1, KG IS SUBTRACTED FROM ELEMENT STIFFNESS           STRU1710
C                - 2, SEPERATE GLOBAL KG IS FORMED                      STRU1720
C KGCOND       - LOGICAL FLAG, TRUE IF KG IS TO BE CONDENSED            STRU1730
C                                                                       STRU1740
      IZKGDT=IZ                                                         STRU1750
      IZ    =IZ +4                                                      STRU1760
      READ(105,*) (NZ(IZKGDT+I),I=1,3) ,KGCOND                         STRU1770
      I=IZKGDT                                                          STRU1780
      IF (NZ(I+1).EQ.0 .OR. NZ(I+2).EQ.0 .OR. NZ(I+3).EQ.0) THEN        STRU1790
         NZ(I+1)=0                                                      STRU1800
         NZ(I+2)=0                                                      STRU1810
         NZ(I+3)=0                                                      STRU1820
       ELSE                                                             STRU1830
         WRITE(108,150) TITLE(1),TITLE(2)                               STRU1840
         IF (NZ(I+1).EQ.1)  WRITE(108,160)                              STRU1850
     &    'KGDATA(1)= 1, LOAD= F(INPUT)*(1.00+ACCEL)',                  STRU1860
     &    '                 F(INPUT) IS POSITIVE IN COMPRESSION',       STRU1870
     &    '                 ACCEL IS POSITIVE GLOBAL Z-AXIS ACCELERATION' STRU1880
         IF (NZ(I+1).EQ.2)  WRITE(108,160)                              STRU1890
     &      'KGDATA(1)= 2, PREVIOUS LOAD STEP''S AXIAL LOAD IS USED'    STRU1900
C                                                                       STRU1910
         IF (NZ(I+2) EQ.1)  WRITE(108,160)                              STRU1920
     &      'KGDATA(2)= 1, LUMPED FORMULATION    '                      STRU1930
         IF (NZ(I+2) EQ.2)  WRITE(108,160)                              STRU1940
     &      'KGDATA(2)= 2, CONSISTENT FORMULATION '                     STRU1950
C                                                                       STRU1960
C                                                                       STRU1970
         IF (NZ(I+3).EQ.1)  WRITE(108,160)                              STRU1980
     &      'KGDATA(3)= 1, KG IS SUBTRACTED FROM ELEMENT STIFFNESS '    STRU1990
         IF (NZ(I+3).EQ.2)  WRITE(108,160)                              STRU2000
     &      'KGDATA(3)= 2, SEPERATE GLOBAL KG IS FORMED'                STRU2010
C                                                                       STRU2020
         WRITE(108,160) '  '                                           STRU2030
         IF (     KGCOND)  WRITE(108,160)                              STRU2040
     &    '- THE GEOMETRIC STIFFNESS MATRIX IS CONDENSED '//            STRU2050
     &                 'WITH THE STRUCTURAL STIFFNESS MATRIX.'         STRU2060
         IF ( NOT.KGCOND)  WRITE(108,160)                               STRU2070
     &    '- THE GEOMETRIC STIFFNESS MATRIX IS NOT CONDENSED '//        STRU2080
     &                 'WITH THE STRUCTURAL STIFFNESS MATRIX.'         STRU2090
       ENDIF                                                            STRU2100
C                                                                       STRU2110
  150 FORMAT ('1 STRUCTURE.....: ',A,/                                  STRU2120
     &        '  SOLUTION.....: ',A,//                                  STRU2130
     &     5X,' GEOMETRIC STIFFNESS DATA '/                            STRU2140
     &     5X,'--------------------------'/)                           STRU2150
  160 FORMAT(5X,A)                                                      STRU2160
C                                                                       STRU2170
C                                                                       STRU2180
C*--------------------------------------------                         STRU2190
C- SET UP STORAGE FOR GEOMETRIC STIFFNESS DATA ---                      STRU2200
C*--------------------------------------------                         STRU2210
C       ---- GLOBAL STIFFNESS STORGE  ----------------                  STRU2220
C            NZ(IZMDKG) - INDICIES OF THE MAIN DIAGONAL TERMS           STRU2230
C             Z(IZKG  ) - GLOBAL GEOMETRIC STIFFNESS MATRIX             STRU2240
      IF (NZ(IZKGDT+3).EQ.2) THEN                                       STRU2250
         IZMDKG=IZ                                                      STRU2260
         IZ    =IZ +NDOF + 1                                            STRU2270
         CALL SKYLIN(0,1,NDOF,IBUG,TITLE,NZ(IZMDKG),                    STRU2280
     &   KGDOF,NZ(IZLMKG),Z(IZKEKG),MD,'GLOBAL GEOMETRIC STIFFNESS')    STRU2290
       ELSE                                                             STRU2300
C        DUMMY LOCATION FOR KG TO AVOID ADDRESSING EXCEPTIONS           STRU2310
C        KG DATA IS NOT STORED IN THESE LOCATIONS...                    STRU2320
         IZMDKG=IZKGDT                                                  STRU2330
         IZKG  =IZKGDT                                                  STRU2340
       ENDIF                                                            STRU2350
C                                                                       STRU2360
C----------------------                                                 STRU2370
C- INPUT ELEMENT DATA --                                                STRU2380
C----------------------                                                 STRU2390
       READ(105,*) NELMT                                                STRU2400
C      ---- ELEMENT STORAGE ---------------------                       STRU2410
C           NZ(IZELE      )-         # OF MATERIAL PORPERTIES (NELMT)  STRU2420
C           NZ(IZELE+    1)-         ABSOLUTE ADDRESS OF ELEMENT #1    STRU2430
C           NZ(IZELE+    2)-         ABSOLUTE ADDRESS OF ELEMENT #2    STRU2440
C           NZ(IZELE+    3)-         ABSOLUTE ADDRESS OF ELEMENT #3    STRU2450
C                    .                        .                      STRU2460
C                    .                        .                      STRU2470
C                    .                        .                      STRU2480
C           NZ(IZMAT+NELMT)-         ABSOLUTE ADDRESS FOR ELEM # NELMT STRU2490
C            Z(NZ(IZELE+1))-         MATERIAL DATA FOR ELEMENT #1      STRU2500
C                    .                        .                      STRU2510
C                    .                        .                      STRU2520
C                    .                        .                      STRU2530
C            Z(NZ(IZELE+2))-         MATERIAL DATA FOR ELEMENT #1      STRU2540
C                    .                        .                      STRU2550
C                    .                        .                      STRU2560
C                    .                        .                      STRU2570
C                                                                    STRU2580
      IOPT=1                                                            STRU2590
      IZELE =IZ                                                         STRU2600
      IZ    =IZ +NELMT+1                                                STRU2610
      NZ(IZELE)=NELMT                                                   STRU2620
      FIRST=.TRUE.                                                      STRU2630
      HEAD=.TRUE.                                                       STRU2640
      IGEN=0                                                            STRU2650
      IELNO=0                                                           STRU2660
      LSTTYP=0                                                          STRU2670
C------------- LOOP FOR EACH ELEMENT ---------- ------------------      STRU2680
  210 IELNO=IELNO+1                                                     STRU2690
      IF (IGEN.LE.0) THEN                                               STRU2700
        READ(105,*) TYPE                                                STRU2710
C                                                                       STRU2720
        IF (TEST(TYPE,'3D-BEAM', FALSE.)) THEN                          STRU2730
           NINPUT= 10                                                   STRU2740
           NINZ  = 437                                                  STRU2750
           NELTYP= 1                                                    STRU2760
C                                                                       STRU2770
        ELSE IF (TEST(TYPE,'SPRING', FALSE.)) THEN                      STRU2780
           NINPUT= 10                                                   STRU2790
           NINZ  = 133                                                  STRU2800
           NELTYP= 2                                                    STRU2810
C                                                                       STRU2820
        ELSE IF (TEST(TYPE,'SHEAR WALL',.FALSE )) THEN                  STRU2830
           NINPUT= 9                                                    STRU2840
           NINZ  = 769                                                  STRU2850
           NELTYP= 3                                                    STRU2860
C                                                                       STRU2870
        ELSE IF (TEST(TYPE,'BRACE', FALSE.)) THEN                       STRU2880
           NINPUT= 10                                                   STRU2890
           NINZ  = 133                                                  STRU2900
           NELTYP= 8                                                    STRU2910
C                                                                       STRU2920
```

```
          ELSE IF (TEST(TYPE,'IE3DBEAM',.FALSE.)) THEN                  STRU2930
             NINPUT= 14                                                 STRU2940
             NINZ  = 614                                                STRU2950
             NELTYP= 9                                                  STRU2960
C                                                                       STRU2970
          ELSE IF (TEST(TYPE,'STABILITY',.FALSE.)) THEN                 STRU2980
             NINPUT= 8                                                  STRU2990
             NINZ  = 618                                                STRU3000
             NELTYP= 12                                                 STRU3010
          ELSE IF (TEST(TYPE,'SHEAR_OPEN',.FALSE.)) THEN                STRU3020
             NINPUT= 9                                                  STRU3030
             NINZ  = 775                                                STRU3040
             NELTYP= 14                                                 STRU3050
C.. .. ..... TO ADD ADDITIONAL ELEMENTS, COPY THE NEXT FOUR LINES       STRU3060
          ELSE IF (TEST(TYPE,'(1)',.FALSE.)) THEN                       STRU3070
             NINPUT= (2)                                                STRU3080
             NINZ  = (3)                                                STRU3090
             NELTYP= (4)                                                STRU3100
C..                    CHANGE THE FOLLOWING VALUES                     STRU3110
C..                    (1) YOUR ELEMENT NAME                           STRU3120
C..                    (2) # OF VALUES TO BE INPUT                     STRU3130
C .                    (3) # OF STORAGE LOCATIONS REQD FOR ELEMENT      STRU3140
C .                    (4) ELEMENT ID NUMBER - XX                      STRU3150
C .                    ADD CALL ELEXX TO SUBPROGRAM   ELE_LIB          STRU3160
C .                    ADD SUBPROGRAM ELEXX                            STRU3170
C.... .. .                                                              STRU3180
          ELSE IF (TEST(TYPE,'END',.FALSE.)) THEN                       STRU3190
             NELMT=IELNO-1                                              STRU3200
             NZ(IZELE)=NELMT                                            STRU3210
             GO TO 300                                                  STRU3220
          ELSE                                                          STRU3230
             WRITE(108,*) 'INVALID ELEMENT TYPE. ',HEAD                 STRU3240
             GO TO 210                                                  STRU3250
          ENDIF                                                         STRU3260
          BACKSPACE (105)                                               STRU3270
          READ(105,*) TYPE,NAME,( RINPUT(I),I=1,NINPUT),IGEN            STRU3280
          IF (IGEN.GT.0)  READ(105,*)  ( GINPUT(I),I=1,NINPUT)          STRU3290
        ELSE                                                            STRU3300
          IGEN=IGEN-1                                                   STRU3310
          NAME='GENERATED'                                              STRU3320
          DO 220 I=1,NINPUT                                             STRU3330
  220       RINPUT(I)=RINPUT(I)+GINPUT(I)                               STRU3340
        ENDIF                                                           STRU3350
C. . . .RESERVE STORAGE                                                 STRU3360
        NZ(IZELE+IELNO)=IZ                                              STRU3370
        IC=IZ                                                           STRU3380
        IZWK=IZ                                                         STRU3390
C                                                                       STRU3400
C.... .STORE ELEMENT TYPE                                               STRU3410
        NZ(IC) =NELTYP                                                  STRU3420
C                                                                       STRU3430
C.....CALL ELEMENT LIBRARY TO DETERMINE THE STORAGE FOR MATL DATA       STRU3440
        IOPT=9                                                          STRU3450
        HEAD2=.FALSE.                                                   STRU3460
        LSTTP2=0                                                        STRU3470
        CALL ELELIB                                                     STRU3480
     &  (IOPT,LSTTP2,.FALSE.,IREL,HEAD2,NAME,EESE,EPSE,DAMAGE,DUCFAG,   STRU3490
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDISP,IZLOAD,MSTOR)    STRU3500
C                                                                       STRU3510
C.....RESERVE STORAGE                                                   STRU3520
        IZ=IZ+NINZ+MSTOR                                                STRU3530
        IZWK=IZ                                                         STRU3540
C                                                                       STRU3550
C... ...DU_MP ELEMENT DATA IF DEBU_GING                                 STRU3560
        IF (BTEST(IBUG,8)) THEN                                         STRU3570
         WRITE(108,*) 'IZLM:',IZLM,' IZKE:',IZKE                        STRU3580
         JJ=-1                                                          STRU3590
         DO 7100 I=IC,(IZWK-1-IREL)                                     STRU3600
          JJ=JJ+1                                                       STRU3610
 7100     IF (NZ(I).NE.0) WRITE(108,7010) IELNO,JJ,I,NZ(I),Z(I)         STRU3620
        ENDIF                                                           STRU3630
C                                                                       STRU3640
C..... .CALL ELEMENT LIBRARY TO INITIALIZE ELEMENT DATA AND RETURN      STRU3650
C.....INFO ON ELEMENT DOF .                                             STRU3660
        IOPT=1                                                          STRU3670
        CALL ELELIB                                                     STRU3680
     &  (IOPT,LSTTYP,.TRUE.,IREL,HEAD,NAME,EESE,EPSE,DAMAGE,DUCFAG,     STRU3690
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDISP,IZLOAD,MSTOR)    STRU3700
C                                                                       STRU3710
C  .....RELEASE UNUSED STORAGE .....                                    STRU3720
        IZ=IZ-IREL                                                      STRU3730
C                                                                       STRU3740
C.......DU_MP ELEMENT DATA IF DEBU_GING                                 STRU3750
        IF (BTEST(IBUG,8)) THEN                                         STRU3760
         WRITE(108,*) 'IZLM:',IZLM,' IZKE:',IZKE                        STRU3770
         JJ=-1                                                          STRU3780
         DO 7000 I=IC,(IZWK-1-IREL)                                     STRU3790
          JJ=JJ+1                                                       STRU3800
 7000     IF (NZ(I).NE.0) WRITE(108,7010) IELNO,JJ,I,NZ(I),Z(I)         STRU3810
 7010    FORMAT (' IELNO:',I3,'  #',I6,'   Z(',I6,'):',I10,1P,G18.8)    STRU3820
        ENDIF                                                           STRU3830
C                                                                       STRU3840
C--------- RESERVE SPACE IN THE SKYLINE FOR THE STIFFNESS               STRU3850
        JFLAG=0                                                         STRU3860
        IF( TEST(TYPE,'STABILITY ',  FALSE. )) JFLAG=1                  STRU3870
        CALL SKYLIN(JFLAG,2,NDOF,IBUG,TITLE,NZ(IZMD),                   STRU3880
     &              IELDOF,NZ(IZLM),Z(IZKE),MD,'GLOBAL STIFFNESS')      STRU3890
C                                                                       STRU3900
C--------- RESERVE SPACE IN THE SKYLINE FOR THE GEOMETRIC STIFFNESS     STRU3910
      IF (NZ(IZKGDT+3).EQ.1 .AND. AXIAL) THEN                           STRU3920
        CALL SKYLIN(0,2,NDOF,IBUG,TITLE,NZ(IZMD),                       STRU3930
     &   KGDOF,NZ(IZLMKG),Z(IZKEKG),MD,'GLOBAL GEOMETRIC STIFFNESS')    STRU3940
      ELSE IF (NZ(IZKGDT+3).EQ.2 .AND. AXIAL) THEN                      STRU3950
        CALL SKYLIN(0,2,NDOF,IBUG,TITLE,NZ(IZMDKG),                     STRU3960
     &   KGDOF,NZ(IZLMKG),Z(IZKEKG),MD,'GLOBAL GEOMETRIC STIFFNESS')    STRU3970
      ENDIF                                                             STRU3980
C                                                                       STRU3990
      IF (IELNO.LT.NELMT) GO TO 210                                     STRU4000
  300 CONTINUE                                                          STRU4010
C                                                                       STRU4020
C                                                                       STRU4030
C-------------------------------------------------------               STRU4040
C- MODIFY GEOMETRIC STIFFNESS MATRIX FOR CONDENSATION  ---               STRU4050
C-------------------------------------------------------               STRU4060
C THE CONDENSATION PROCESS MAY EXPAND THE GEOMETRIC STIFFNESS MATRIX'S  STRU4070
C SKYLINE.  THE MAXIMUN SKYLINE OF BOTH THE GEOMETRIC STIFFNESS AND     STRU4080
C THE STIFFNESS MATRIX IS USED.                                         STRU4090
      IF (NZ(IZKGDT+3).EQ.2 .AND. KGCOND .AND  NCOND.GT.0)THEN          STRU4100
        DO 310 I=0,NDOF                                                 STRU4110
          J=I+1                                                         STRU4120
          MDKG1=NZ(IZMDKG-I)                                            STRU4130
C         NZ(IZMDKG-I)=MAX( NZ(IZMDKG-I),NZ(IZMD-I) )                   STRU4140
          NZ(IZMDKG-I)=MIN( NZ(IZMDKG-I),NZ(IZMD-I) )                   STRU4150
  310     CONTINUE                                                      STRU4160
      ENDIF                                                             STRU4170
C                                                                       STRU4180
C                                                                       STRU4190
C-------------------------------------                                  STRU4200
C- SET UP GLOBAL STIFFNESS STORAGE --                                   STRU4210
C-------------------------------------                                  STRU4220
C            NZ(IZMD)   - INDICIES OF THE MAIN DIAGONAL TERMS           STRU4230
C             Z(IZSTIF) - GLOBAL STIFFNESS MATRIX                       STRU4240
C                                                                       STRU4250
      CALL SKYLIN(0,3,NDOF,ISUG,TITLE,NZ(IZMD),                         STRU4260
     &   IELDOF,NZ(IZLM),Z(IZKE),MD,'GLOBAL STIFFNESS')                 STRU4270
      IZSTIF=IZ                                                         STRU4280
      IZ     = IZ+NZ(IZMD+NDOF) +1                                      STRU4290
C                                                                       STRU4300
C-------------------------------------------                            STRU4310
C- SET UP STORAGE FOR GEOMETRIC STIFFNESS DATA ---                      STRU4320
C-------------------------------------------                            STRU4330
C            NZ(IZMDKG) - INDICIES OF THE MAIN DIAGONAL TERMS           STRU4340
C             Z(IZKG  ) - GLOBAL GEOMETRIC STIFFNESS MATRIX             STRU4350
      IF (NZ(IZKGDT+3).EQ.2) THEN                                       STRU4360
        CALL SKYLIN(0,3,NDOF,IBUG,TITLE,NZ(IZMDKG),                     STRU4370
     &   KGDOF,NZ(IZLMXG),Z(IZKEKG),MD,'GLOBAL GEOMETRIC STIFFNESS')    STRU4380
        IZKG  =IZ                                                       STRU4390
        IZ    =IZ +NZ(IZMDKG+NDOF)                                      STRU4400
      ENDIF                                                             STRU4410
C                                                                       STRU4420
C                                                                       STRU4430
C----------------------                                                 STRU4440
C- INPUT MASS MATRIX ---                                                STRU4450
C----------------------                                                 STRU4460
C    NMASS  = NUMBER OF NODAL MASSES TO BE INPUT                        STRU4470
C    IPMASS = 0, NO MASS MATRIX                                         STRU4480
C             1, MASS MATRIX DUE TO NODE MASSES ONLY                    STRU4490
C    MCOND  = LOGICAL FLAG, TRUE IF MASS IS TO BE CONDENSED. .          STRU4500
C                                                                       STRU4510
      READ(105,*) NMASS,IPMASS,MCOND                                    STRU4520
C                                                                       STRU4530
      CALL MASSIN(NMASS,IPMASS,MCOND,BUG,IBUG,TITLE,MAXZ,               STRU4540
     &            MAXNOD,NNODE,IZ,IW,IZID,2,NZ,IZMDMS,IZMASS,           STRU4550
     &            IZLM,IZKE,IZMDAT,NDOF,NMASS,IZIDOF,IZCNST,            STRU4560
     &            IZJCOS,NCOND,IZMD)                                    STRU4570
      IF (NMASS.GT.0) THEN                                              STRU4580
         WRITE(108,160) ' '                                             STRU4590
         IF (MCOND) THEN                                                STRU4600
            WRITE(108,160) '- THE MASS MATRIX IS '//                    STRU4610
     &      'CONDENSED WITH THE STRUCTURAL STIFFNESS MATRIX.'            STRU4620
         ELSE                                                           STRU4630
            WRITE(108,160) '- THE MASS MATRIX IS NOT '//                STRU4640
     &      'CONDENSED WITH THE STRUCTURAL STIFFNESS MATRIX.'            STRU4650
         ENDIF                                                          STRU4660
      ELSE                                                              STRU4670
         WRITE(108,499)                                                 STRU4680
  499    FORMAT(//5X,' ZERO MASS MATRIX '/                              STRU4690
     &         5X,'------------------')                                 STRU4700
      ENDIF                                                             STRU4710
C                                                                       STRU4720
C----------------------                                                 STRU4730
C- INPUT DAMPING DATA ---                                               STRU4740
C----------------------                                                 STRU4750
C    NDAMP - NUMBER OF MODAL DAMPING COIEF TO BE INPUT -0               STRU4760
C                                                                       STRU4770
      NDAMP=0                                                           STRU4780
      FDAMP=1                                                           STRU4790
      IZDAMP=IZ                                                         STRU4800
      IZ=IZ + 2                                                         STRU4810
      READ(105,*) ALPHA,BETA                                            STRU4820
      Z(IZDAMP)   =ALPHA                                                STRU4830
      Z(IZDAMP+1) =BETA                                                 STRU4840
      WRITE(108,500) ALPHA,BETA                                         STRU4850
  500 FORMAT(//5X,' PROPORTIONAL DAMPING COIEFICIENTS'/                 STRU4860
     &         5X,'-----------------------------------'/,               STRU4870
     &         5X,'ALPHA=',1P,G13.5,'   BETA=',G13.5/)                  STRU4880
C                                                                       STRU4890
C--------------------------------                                       STRU4900
C- ZERO LOAD AND DISPL INDICIES ---                                     STRU4910
C--------------------------------                                       STRU4920
      IZLOAD=0                                                          STRU4930
      IZDISP=0                                                          STRU4940
      IZVEL =0                                                          STRU4950
      IZACC =0                                                          STRU4960
      IZDLOA=0                                                          STRU4970
      IZDDSP=0                                                          STRU4980
      IZDVEL=0                                                          STRU4990
      IZDACC=0                                                          STRU5000
C                                                                       STRU5010
      RETURN                                                            STRU5020
      END                                                               STRU5030
C                                                                       MAT10030
C-----------------------------------------------------------------------MAT10040
C                                                                       MAT10050
      SUBROUTINE MAT11(IOPT,FIRST,MAT,SE,P,PL,D,DL,LELEM,LMAT,          MAT10060
     &                 SMAT,EESE,EPSE,DUCT,EXCR,DAMAGE)                 MAT10070
C                                                                       MAT10080
      LOGICAL FIRST                                                     MAT10090
      DIMENSION SE(100),SMAT(5),DUCT(3),EXCR(6)                         MAT10100
      CHARACTER*80 TYPE                                                 MAT10110
C                                                                       MAT10120
C------- RESERVE STORAGE FOR ELEMENT                                    MAT10130
      LELEM=31                                                          MAT10140
      IF (IOPT.EQ.0) RETURN                                             MAT10150
C                                                                       MAT10160
C--------- ------------------------------------------- ----------------MAT10170
C VARIABLES:                                                            MAT10180
C---------- GLOBAL VARIABLES                                           MAT10190
C IOPT   - 1, INITIALIZE MATERIAL                                       MAT10200
C        - 2, INITIALIZE STIFFNESS                                      MAT10210
C        - 3, GET STIFFNESS TERM                                        MAT10220
C        - 4, GET ENERGIES, DUCTILITY AND EXCURSION                     MAT10230
C        - 5, CALCULATE DAMAGE INDEX                                    MAT10240
C RINPUT - INPUT DATA                                                   MAT10250
C SMAT   - INTERNAL STORAGE                                             MAT10260
C SE     - OUTPUT STIFFNESS                                             MAT10270
C-----------------------------------------------------------------------MAT10280
      GO TO (100,200,300,400,500) IOPT                                  MAT10290
C                                                                       MAT10300
  100 CONTINUE                                                          MAT10310
C                                                                       MAT10320
C----- INTERPERTATE INPUT DATA                                          MAT10330
C     HA    = SMAT( 1)                                                  MAT10340
C     VA    = SMAT( 2)                                                  MAT10350
C     RATIO = SMAT( 3)   : STRAIN HARTEN RATIO TO ELASTIC STIFFNESS     MAT10360
C     DUCMAX= SMAT( 4)   : FAILURE DUCTILITY                            MAT10370
C     BDAM  = SMAT( 5)   : BATA PARAMETER                               MAT10380
C                                                                       MAT10390
      READ(105,*) TYPE ,(SMAT(I),I=1,3), DUCMAX, BDAM                   MAT10400
C                                                                       MAT10410
      LMAT = 5                                                          MAT10420
      SMAT(4)=DUCMAX                                                    MAT10430
      SMAT(5)=BDAM                                                      MAT10440
C                                                                       MAT10450
      IF (FIRST) WRITE(108,191)                                         MAT10460
      WRITE(108,192) MAT,(SMAT(K),K=1,3),DUCMAX,BDAM                    MAT10470
C                                                                       MAT10480
  191  FORMAT (///' PINO-SUAREZ TOWER SHORT DIR. HYST. MODEL PARAMETERS'/MAT10490
     & ' ------------------------------------------------'//,'  MAT.',  MAT10500
     & 3X,'   HA',6X,'   VA',6X,' RATIO',6X,'MAX DUC', 6X,'  BETA')     MAT10500
  192  FORMAT (1X,I3,2X,6G12.6/)                                         MAT10510
C                                                                       MAT10520
      RETURN                                                            MAT10530
C                                                                       MAT10540
C------------------------ INITIALIZE STIFFNESS TERMS ------------------MAT10550
  200 CONTINUE                                                          MAT10560
C     ---- PINO-SUAREZ TOWER SHORT GIRDER MODEL DATA STORAGE FOR MEMBER MAT10570
C     SE( 1)- STIF   MEMBER MOMENT-ROTATION STIFFNESS                   MAT10580
C     SE( 2)- DP     LOAD INCREMENT                                     MAT10590
C     SE( 3)- DX     DISPLACEMENT INCREMENT                             MAT10600
C     SE( 4)- RV     INDEX FOR TOP OR BOTTOM ENVELOPE                   MAT10610
C                    RV=0 THEN HIGH=.TRUE.                              MAT10620
C                    RV=1 THEN HIGH=.FALSE.                             MAT10630
C     SE( 5)- RULE   RULE NUMBER                                        MAT10640
C     SE( 6)- PTOP                                                      MAT10650
C     SE( 7)- DTOP                                                      MAT10660
C     SE( 8)- PBOT                                                      MAT10670
C     SE( 9)- DBOT                                                      MAT10680
C     SE(10)- KEQ    UNLOADING EQUIVALENT STIFFNESS                     MAT10690
C     SE(11)- DUCT(1)                                                   MAT10700
C     SE(12)- DUCT(2)                                                   MAT10710
C     SE(13)- DUCT(3)                                                   MAT10720
C     SE(14)- EXCR(1)                                                   MAT10730
C     SE(15)- EXCR(2)                                                   MAT10740
C     SE(16)- EXCR(3)                                                   MAT10750
C     SE(17)- EXCR(4)                                                   MAT10760
C     SE(18)- EXCR(5)                                                   MAT10770
C     SE(19)- EXCR(6)                                                   MAT10780
C     SE(20)- DELY    YIELDING DISPLACEMENT                             MAT10790
C     SE(21)- EESE(1)                                                   MAT10800
C     SE(22)- EESE(2)                                                   MAT10810
```

```
C     SE(23)= EESE(2)                                                   MAT10820
C     SE(24)= EPSE                                                      MAT10830
C     SE(25)= PSEOLD                                                    MAT10840
C     SE(26)= CSE                                                       MAT10850
C     SE(27)= DMAX   ULTIMATE DISPLACEMENT                              MAT10860
C     SE(28)= D_MAX  MAXIMUM ELEMENT DISPLACEMENT FOR DAMAGE INDEX      MAT10870
C     SE(29)= P_MAX  ELEMENT LOAD CORRESPONDING TO D_MAX                MAT10880
C     SE(30)= DEMAX  CURRENT MAXIMUM AXIAL DISPLACEMENT FOR DUCTILITIES MAT10890
C     SE(31)= INITIAL STIFFNESS                                         MAT10900
C ----------------------------------------------------------------      MAT10910
      DO 210 I=1,31                                                     MAT10920
  210   SE(I)=0                                                         MAT10930
      SE(20)=SMAT(1)                                                    MAT10940
      SE(5)=1                                                           MAT10950
      MOPT=IOPT                                                         MAT10960
      CALL HYST11(MOPT,SE(1),0.,0.,0.,0.,SE(3),SE(4),SE(5),SE(6),SE(7), MAT10970
     &    SE(8),SE(9),SMAT(1),SMAT(2),SMAT(3),SE(11),SE(30),SMAT(4))    MAT10980
      SE(26)=SE(20)*SMAT(2)/2                                           MAT10990
      SE(27)=SMAT(1)*SMAT(4)                                            MAT11000
      SE(28)=0                                                          MAT11010
      SE(29)=0                                                          MAT11020
      SE(31)=SE(1)                                                      MAT11030
C                                                                       MAT11040
      RETURN                                                            MAT11050
C------------------------ GET STIFFNESS TERMS ------------------------- MAT11060
  300 CONTINUE                                                          MAT11070
      SE(2)=P-PL                                                        MAT11080
      SE(3)=D-DL                                                        MAT11090
C                                                                       MAT11100
C-- CALL HYST11 TO CALC NEW STIFFNESS STIF                              MAT11110
      MOPT=IOPT                                                         MAT11120
      CALL HYST11(MOPT,SE(1),P,D,PL,DL,SE(3),SE(4),SE(5),SE(6),SE(7),   MAT11130
     &    SE(8),SE(9),SMAT(1),SMAT(2),SMAT(3),SE(11),SE(30),SMAT(4))    MAT11140
C                                                                       MAT11150
C --- STORE MAXIMUM DISPLACEMENT AND CORRESPONDING LOAD                 MAT11160
      IF(ABS(D) .GT. ABS(SE(28))) THEN                                  MAT11170
            SE(28)=D                                                    MAT11180
            SE(29)=P                                                    MAT11190
      ENDIF                                                             MAT11200
C                                                                       MAT11210
C--- GET ENERGIES, DUCTILITY AND EXCURSION FOR BOTH IOPT=3 AND 4 ----   MAT11220
  400 CONTINUE                                                          MAT11230
C                                                                       MAT11240
C --- TRANSFER ENERGIES FOR BOTH IOPT=3 AND 4.                          MAT11250
      EESE=SE(21)                                                       MAT11260
      EPSE=SE(24)                                                       MAT11270
C                                                                       MAT11280
C --- TRANSFER DUCTILITIES AND EXCURSION RATIOS FOR BOTH IOPT=3 AND 4.  MAT11290
      DO 410 I=1,3                                                      MAT11300
        DUCT(I)=SE(10+I)                                                MAT11310
        EXCR(I+3)=SE(13+I+3)                                            MAT11320
  410   EXCR(I)=SE(13+I)                                                MAT11330
C                                                                       MAT11340
      RETURN                                                            MAT11350
C----------------- DAMAGE INDEX ------------------------------------    MAT11360
  500 CONTINUE                                                          MAT11370
C..DAMAGE INDEX IS NOT AVAILABLE                                        MAT11380
      DAMAGE=0                                                          MAT11390
C                                                                       MAT11400
      RETURN                                                            MAT11410
      END                                                               MAT11420
C                                                                       HYST0010
C     PINO-SUAREZ TOWER SHORT DIRECTION GIRDER MOMENT-ROTATION          HYST0020
C     HYSTERESIS MODEL BASED ON HYSTERESIS LOOP OF T2A GIRDER           HYST0030
C                                                                       HYST0040
      SUBROUTINE HYST11(MOPT,STIF,P,X,PLA,XLA,DX,RV,RULE,PTOP,DTOP,     HYST0050
     &                  PBOT,DBOT,HA,VA,RATIO,DUC,DEMAX,DUCMAX)         HYST0060
      DIMENSION DUC(3)                                                  HYST0070
      LOGICAL HIGH                                                      HYST0080
C                                                                       HYST0090
C ----------------------------------------------------------------      HYST0100
C   DUC     ---- ELEMENT DUCTILITY RATE                                 HYST0110
C   EXCR    ---- ELEMENT EXCURSION RATE                                 HYST0120
C   ESE     ---- ELEMENT ELASTIC STRAIN RATE                            HYST0130
C   PSE     ---- ELEMENT PLASTIC STRAIN RATE                            HYST0140
C   PSEOLD  ---- ELEMENT PLASTIC STRAIN RATE AT LAST ZERO CROSSING      HYST0150
C   CSE     ---- ELEMENT CONSTANT STRAIN ENERGY                         HYST0160
C   RATIO   ---- STRAIN HARTEN RATIO TO ELASTIC STIFFNESS               HYST0170
C   STIF    ---- ELEMENT STIFFNESS                                      HYST0180
C   P       ---- CURRENT ELEMENT INTERNAL FORCE                         HYST0190
C   X       ---- CURRENT ELEMENT INTERNAL DISPLACEMENT                  HYST0200
C   PLA     ---- PREVIOUS ELEMENT INTERNAL FORCE                        HYST0210
C   XLA     ---- PREVIOUS ELEMENT INTERNAL DISPLACEMENT                 HYST0220
C   DP      ---- ELEMENT INTERNAL FORCE INCREMENT                       HYST0230
C   DX      ---- ELEMENT INTERNAL DISPLACEMENT INCREMENT                HYST0240
C   HIGH    ---- LOGICAL INDEX, IF P .GT. 0 THEN HIGH=.TRUE.            HYST0250
C   RULE    ---- RULE NUMBER                                            HYST0260
C   PTOP    ---- REVERSED LOADING POINT AT TOP ENVELOPE                 HYST0270
C   DTOP    ---- REVERSED DISP. POINT AT TOP ENVELOPE                   HYST0280
C   PBOT    ---- REVERSED LOADING POINT AT BOTTOM ENVELOPE              HYST0290
C   DBOT    ---- REVERSED DISP. POINT AT BOTTOM ENVELOPE                HYST0300
C   HA,VA   ---- FIRST BUCKLING POINT AT TOP ENVELOPE                   HYST0310
C   DUCMAX  ---- FAILURE DUCTILITY                                      HYST0320
C                .GE. 1 FAILURE DUCTILITY CONSIDERED                    HYST0330
C                .LT. 1 FAILURE DUCTILITY NOT CONSIDERED                HYST0340
C ----------------------------------------------------------------      HYST0350
C                                                                       HYST0360
C ---------------------------                                          HYST0370
C - CHECK FAILURE DUCTILITY --                                          HYST0380
C ---------------------------                                          HYST0390
      IF( DUC(1) .GE. DUCMAX .AND. DUCMAX .GE. 1 ) THEN                 HYST0400
        P=0                                                             HYST0410
        STIF=0.001                                                      HYST0420
        RETURN                                                          HYST0430
      ENDIF                                                             HYST0440
C                                                                       HYST0450
        IF( RV .EQ. 1 ) HIGH=.FALSE.                                    HYST0460
        IF( RV .EQ  0 ) HIGH=.TRUE.                                     HYST0470
C                                                                       HYST0480
C -------------------------------                                      HYST0490
C - DEFINE ENVELOPE CONTROL POINTS -                                    HYST0500
C -------------------------------                                      HYST0510
        IF( MOPT .EQ. 2 ) THEN                                          HYST0520
            PTOP=VA                                                     HYST0530
            PBOT=-VA                                                    HYST0540
            DTOP=HA                                                     HYST0550
            DBOT=-HA                                                    HYST0560
        ENDIF                                                           HYST0570
      IF( P .GT. 0 ) HIGH=.TRUE.                                        HYST0580
      IF( P .LT. 0 ) HIGH=.FALSE.                                       HYST0590
      IF( HIGH) RV=0                                                    HYST0600
      IF( .NOT. HIGH) RV=1                                              HYST0610
C                                                                       HYST0620
C -------------------------                                            HYST0630
C - DEFINE ENVELOPE SLOPE -                                             HYST0640
C -------------------------                                            HYST0650
      SLOP1=VA/HA                                                       HYST0660
      SLOP2=RATIO*SLOP1                                                 HYST0670
C                                                                       HYST0680
C                                                                       HYST0690
C -------------------------                                            HYST0700
C - HYSTERESIS LOOP RULES -                                             HYST0710
C -------------------------                                            HYST0720
C--RULE 1:                                                              HYST0730
      IF( RULE .EQ. 1 ) THEN                                            HYST0740
        IF(P .GE. PTOP .OR. P .LE. PBOT ) THEN                          HYST0750
            RULE=2                                                      HYST0760
            STIF=SLOP2                                                  HYST0770
C           *** UNBALANCED FORCE ***                                    HYST0780
            IF( P .GE. PTOP ) THEN                                      HYST0790
                UP1=P-PTOP                                              HYST0800
                UD=UP1/SLOP1                                            HYST0810
                UP2=-UD*SLOP2                                           HYST0820
                P=P-(UP1+UP2)                                           HYST0830
            ELSE IF( P .LE. PBOT ) THEN                                 HYST0840
                UP1=P-PBOT                                              HYST0850
                UD=UP1/SLOP1                                            HYST0860
                UP2=-UD*SLOP2                                           HYST0870
                P=P-(UP1+UP2)                                           HYST0880
            ENDIF                                                       HYST0890
C                                                                       HYST0900
        ELSE                                                            HYST0910
            RULE=1                                                      HYST0920
            STIF=SLOP1                                                  HYST0930
        ENDIF                                                           HYST0940
        IF( DEMAX .LT. ABS(X) ) THEN                                    HYST0950
            DEMAX=ABS(X)                                                HYST0960
            DUC(1)=DEMAX/HA                                             HYST0970
        ENDIF                                                           HYST0980
      ENDIF                                                             HYST0990
C                                                                       HYST1000
C --RULE 2:                                                             HYST1010
      IF( RULE .EQ. 2 ) THEN                                            HYST1020
        IF( HIGH .AND. DX .GT. 0 ) THEN                                 HYST1030
            RULE=2                                                      HYST1040
            STIF=SLOP2                                                  HYST1050
        ELSE IF( .NOT. HIGH .AND. DX .LT. 0 ) THEN                      HYST1060
            RULE=2                                                      HYST1070
            STIF=SLOP2                                                  HYST1080
        ELSE IF( HIGH .AND. DX .LE. 0 ) THEN                            HYST1090
            PTOP=PLA                                                    HYST1100
            DTOP=XLA                                                    HYST1110
            RULE=1                                                      HYST1120
            STIF=SLOP1                                                  HYST1130
            PBOT=PTOP - 2*VA                                            HYST1140
            DBOT=DTOP - (PTOP-PBOT)/SLOP1                               HYST1150
C           *** UNBALANCED FORCE ***                                    HYST1160
            UD=X-DTOP                                                   HYST1170
            UP=UD*SLOP1                                                 HYST1180
            P=PTOP+UP                                                   HYST1190
        ELSE IF( .NOT. HIGH .AND. DX .GE. 0 ) THEN                      HYST1200
            PBOT=PLA                                                    HYST1210
            DBOT=XLA                                                    HYST1220
            RULE=1                                                      HYST1230
            STIF=SLOP1                                                  HYST1240
            PTOP=PBOT + 2*VA                                            HYST1250
            DTOP=DBOT + (PTOP-PBOT)/SLOP1                               HYST1260
C           *** UNBALANCED FORCE ***                                    HYST1270
            UD=X-DBOT                                                   HYST1280
            UP=UD*SLOP1                                                 HYST1290
            P=PBOT+UP                                                   HYST1300
        ENDIF                                                           HYST1310
        IF( DEMAX .LT. ABS(X) ) THEN                                    HYST1320
            DEMAX=ABS(X)                                                HYST1330
            DUC(1)=DEMAX/HA                                             HYST1340
        ENDIF                                                           HYST1350
      ENDIF                                                             HYST1360
C                                                                       HYST1370
      RETURN                                                            HYST1380
      END                                                               HYST1390
C                                                                       MAT10010
C----------------------------------------------------------------------MAT10020
      SUBROUTINE MAT13(IOPT,FIRST,MAT,SE,P,PL,D,DL,LELEM,LMAT,          MAT10030
     &                 SMAT,EESE,EPSE,DUCT,EXCR,DAMAGE)                 MAT10040
C                                                                       MAT10050
      LOGICAL FIRST                                                     MAT10060
      DIMENSION SE(100),SMAT(8),DUCT(3),EXCR(6)                         MAT10070
      CHARACTER*80 TYPE                                                 MAT10080
C                                                                       MAT10090
C------- RESERVE STORAGE FOR ELEMENT                                    MAT10100
      LELEM=39                                                          MAT10110
      IF (IOPT.EQ.0) RETURN                                             MAT10120
C                                                                       MAT10130
C----------------------------------------------------------------------MAT10140
C VARIABLES:                                                            MAT10150
C----------------------------------------------------------------------MAT10160
C---------- GLOBAL VARIABLES                                           MAT10170
C IOPT   = 1, INITIALIZE MATERIAL                                       MAT10180
C        = 2, GET STIFFNESS                                             MAT10190
C RINPUT = INPUT DATA                                                   MAT10200
C SMAT   = INTERNAL STORAGE                                             MAT10210
C SE     = OUTPUT STIFFNESS                                             MAT10220
C----------------------------------------------------------------------MAT10230
      GO TO (100,200,300,400,500) IOPT                                  MAT10240
C                                                                       MAT10250
  100 CONTINUE                                                          MAT10260
C                                                                       MAT10270
C----- INTERPERTATE INPUT DATA                                          MAT10280
C     E    = SMAT( 1)                                                   MAT10290
C     A    = SMAT( 2)                                                   MAT10300
C     B    = SMAT( 3)                                                   MAT10310
C     YS   = SMAT( 4)                                                   MAT10320
C     PY   = SMAT( 5)                                                   MAT10330
C     CC   = SMAT( 6)                                                   MAT10340
C     DMAX = SMAT( 7)   : ULTIMATE DISPLACEMENT = DUCMAX*DELY           MAT10350
C     BDAM = SMAT( 8)   : BATA PARAMETER                                MAT10360
C                                                                       MAT10370
      READ(105,*) TYPE ,(SMAT(I),I=1,4), DUCMAX, BDAM                   MAT10380
C                                                                       MAT10390
      LMAT = 8                                                          MAT10400
      PI=3.1415926                                                      MAT10410
      SMAT(5)=SMAT(2)*SMAT(4)                                           MAT10420
      SMAT(6)=SQRT(2.*PI*PI*SMAT(1)/SMAT(4) )                           MAT10430
      SMAT(7)=DUCMAX                                                    MAT10440
      SMAT(8)=BDAM                                                      MAT10450
C                                                                       MAT10460
      IF (FIRST) WRITE(108,191)                                         MAT10470
      WRITE(108,192) MAT,(SMAT(K),K=1,4),DUCMAX,BDAM                    MAT10480
  191 FORMAT ('///'     GOEL HYSTERESIS MODEL FOR BOX LOCAL BUCKLING'/  MAT10490
     &        '  ----------------------------------------------'//      MAT10500
     &        '  MAT.',3X,'  E',6X,'  A',6X,'  B',6X,'  YS',            MAT10510
     &        6X,'MAX DUC', 6X,'  BETA')                                MAT10520
  192 FORMAT (1X,I5,6G12.6/)                                            MAT10530
C                                                                       MAT10540
      RETURN                                                            MAT10550
C                                                                       MAT10560
C------------------------ INITIALIZE STIFFNESS TERMS ------------------MAT10570
  200 CONTINUE                                                          MAT10580
C           ---- GOEL HYSTERESIS MODEL DATA STORAGE FOR MEMBER  ---    MAT10590
C     SE( 1)=                                                           MAT10600
C     SE( 2)= K=1     EFFECT SLENDER COEFFICIENT                        MAT10610
C     SE( 3)= NOCY    NO  OF CYCLE                                      MAT10620
C     SE( 4)= NRULE   RULE NUMBER                                       MAT10630
C     SE( 5)= EL      MEMBER LENGTH (TRANSFERED FROM ELE08)             MAT10640
C     SE( 6)= IRV     0:REVRSC-.FALSE.    1:REVRSC-.TRUE                MAT10650
C     SE( 7)= IDUC    0: HC--12           1: HC .LT. -12                MAT10660
C     SE( 8)= PBOT                                                      MAT10670
C     SE( 9)= DBOT                                                      MAT10680
C     SE(10)= PTOP                                                      MAT10690
C     SE(11)= DTOP                                                      MAT10700
C     SE(12)= KEQ     UNLOADING EQUIVALENT STIFFNESS                    MAT10710
C     SE(13)= DUCT(1)                                                   MAT10720
C     SE(14)= DUCT(2)                                                   MAT10730
C     SE(15)= DUCT(3)                                                   MAT10740
C     SE(16)= EXCR(1)                                                   MAT10750
C     SE(17)= EXCR(2)                                                   MAT10760
C     SE(18)= EXCR(3)                                                   MAT10770
C     SE(19)= EXCR(4)                                                   MAT10780
C     SE(20)= EXCR(5)                                                   MAT10790
C     SE(21)= EXCR(6)                                                   MAT10800
C     SE(22)= DELY    YIELDING DISPLACEMENT                             MAT10810
C     SE(23)= EESE(1)                                                   MAT10820
C     SE(24)= EESE(2)                                                   MAT10830
C     SE(25)= EESE(2)                                                   MAT10840
C     SE(26)= EPSE                                                      MAT10850
C     SE(27)= PSEOLD                                                    MAT10860
C     SE(28)= CSE                                                       MAT10870
C     SE(29)= DMAX   ULTIMATE DISPLACEMENT                              MAT10880
C     SE(30)= D_MAX  MAXIMUM ELEMENT DISPLACEMENT FOR DAMAGE INDEX      MAT10890
C     SE(31)= P_MAX  ELEMENT LOAD CORRESPONDING TO D_MAX                MAT10900
C     SE(32)= HA     CURRENT CONTROL POINT HA                           MAT10910
C     SE(33)= VA     CURRENT CONTROL POINT VA                           MAT10920
```

```
C     SE(34)= HC     CURRENT CONTROL POINT HC                                   MAT10930
C     SE(35)= VC     CURRENT CONTROL POINT VC                                   MAT10940
C     SE(36)= RSN    CURRENT RESIDUAL STRAIN VALUE                              MAT10950
C     SE(37)= DEMAX  CURRENT MAXIMUM AXIAL DISPLACEMENT FOR DUCTILITIES         MAT10960
C     SE(38)= REGIN  CURRENT REGINE ZONE NUMBER                                 MAT10970
C     SE(39)= PREGIN PREVIOUT REGIN ZONE NUMBER                                 MAT10980
C ------------------------------------------------------------                  MAT10990
      SE(1)=1                                                                   MAT11000
      SE(2)=1                                                                   MAT11010
      SE(3)=1                                                                   MAT11020
      SE(4)=1                                                                   MAT11030
      DO 210 I=6,28                                                             MAT11040
  210   SE(I)=0                                                                 MAT11050
      SE(37)=0                                                                  MAT11060
      SE(38)=1                                                                  MAT11070
      SE(39)=1                                                                  MAT11080
      CALL HYST13(0.,0.,0.,0.,SE(5),SE(2),SMAT(1),SMAT(2),SMAT(3),             MAT11090
     &     SMAT(4),SMAT(5),SMAT(6),SE(3),SE(4),SE(1),SE(6),SE(7),               MAT11100
     &     SE(32),SE(33),SE(34),SE(35),SE(36),SE(37),SE(38),SE(39),             MAT11110
     &     SE(8),SE(9),SE(10),SE(11),SE(12),SE(13),SE(16),SE(22),               MAT11120
     &     SE(23),SE(26),SE(27),SE(28) )                                        MAT11130
      SE(28)=SE(22)*SMAT(5)/2                                                   MAT11140
      SE(29)=SE(22)*SMAT(7)                                                     MAT11150
      SE(30)=0                                                                  MAT11160
      SE(31)=0                                                                  MAT11170
C                                                                               MAT11180
      RETURN                                                                    MAT11190
C--------------------------- GET STIFFNESS TERMS ----------------------------- MAT11200
  300 CONTINUE                                                                  MAT11210
C                                                                               MAT11220
C-------- CALL HYST13 TO CALC NEW STIFFNESS STIF                                MAT11230
      CALL HYST13(P,PL,D,DL,SE(5),SE(2),SMAT(1),SMAT(2),SMAT(3),                MAT11240
     &     SMAT(4),SMAT(5),SMAT(6),SE(3),SE(4),SE(1),SE(6),SE(7),               MAT11250
     &     SE(32),SE(33),SE(34),SE(35),SE(36),SE(37),SE(38),SE(39),             MAT11260
     &     SE(8),SE(9),SE(10),SE(11),SE(12),SE(13),SE(16),SE(22),               MAT11270
     &     SE(23),SE(26),SE(27),SE(28) )                                        MAT11280
C                                                                               MAT11290
C --- STORE MAXIMUM DISPLACEMENT AND CORRESPONDING LOAD                         MAT11300
      IF(ABS(D) .GT. ABS(SE(30))) THEN                                          MAT11310
              SE(30)=D                                                          MAT11320
              SE(31)=P                                                          MAT11330
      ENDIF                                                                     MAT11340
C                                                                               MAT11350
C--- GET ENERGIES, DUCTILITY AND EXCURSION FOR BOTH IOPT-3 AND 4 ----           MAT11360
  400 CONTINUE                                                                  MAT11370
C                                                                               MAT11380
C --- TRANSFER ENERGIES FOR BOTH IOPT=3 AND 4                                   MAT11390
      EESE=SE(23)                                                               MAT11400
      EPSE=SE(26)                                                               MAT11410
C                                                                               MAT11420
C --- TRANSFER DUCTILITIES AND EXCURSION RATIOS FOR BOTH IOPT-3 AND 4.          MAT11430
      DO 410 I=1,3                                                              MAT11440
        DUC(I)=SE(12+I)                                                         MAT11450
        EXCR(I+3)=SE(15+I+3)                                                    MAT11460
  410   EXCR(I)=SE(15+I)                                                        MAT11470
C                                                                               MAT11480
        RETURN                                                                  MAT11490
C----------------- DAMAGE INDEX ------------------------------------           MAT11500
  500 CONTINUE                                                                  MAT11510
      EESE=SE(23)                                                               MAT11520
      EPSE=SE(26)                                                               MAT11530
      DMAX=SE(29)                                                               MAT11540
      D=SE(30)                                                                  MAT11550
      P=SE(31)                                                                  MAT11560
      PY=SMAT(5)                                                                MAT11570
      BDAM=SMAT(8)                                                              MAT11580
      DAMAGE=ABS(D)/DMAX + BDAM / (PY*DMAX) * EPSE                              MAT11590
C                                                                               MAT11600
      RETURN                                                                    MAT11610
      END                                                                       MAT11620
C                                                                               HYST0010
C     GOEL TRUSS HYSTERESIS MODEL FOR SIMULATED LOCAL BUCKLING                  HYST0020
C     OF BOX SECTION                                                            HYST0030
C     NOTE: IT IS ASSUMED THAT OVERALL BUCKLING WILL NOT OCCURRED               HYST0040
C                                                                               HYST0050
      SUBROUTINE HYST13(P,PLA,DE,DELA,EL,SLK,EM,AREA,RADG,YS,PY,                HYST0060
     &          CC,CYCLE,RULE,STIF,IRV,IDUC,HA,VA,HC,VC,RSN,DEMAX,              HYST0070
     &          REGIN,PREGIN,PBOT,DBOT,PTOP,DTOP,EXI,DUC,                       HYST0080
     &          EXCR,DELY,ESE,PSE,PSEOLD,CSE)                                   HYST0090
      DIMENSION DUC(3), EXCR(6), ESE(3)                                         HYST0100
      LOGICAL REVRSC                                                            HYST0110
      REAL*4 IRV, IDUC                                                          HYST0120
C                                                                               HYST0130
      IF( IRV .EQ. 0.) REVRSC= FALSE.                                           HYST0140
      IF( IRV .EQ. 1.) REVRSC= TRUE.                                            HYST0150
      PI=3.1415926                                                              HYST0160
      ES=AREA*EM/EL                                                             HYST0170
      DELY=PY/ES                                                                HYST0180
      DEL=DE/DELY                                                               HYST0190
      PL=P/PY                                                                   HYST0200
      ESR=SLK*EL/RADG                                                           HYST0210
      SLOP7=STIF/ES                                                             HYST0220
      DA=ABS(DE)                                                                HYST0230
      DP=P-PLA                                                                  HYST0240
      DDE=DE-DELA                                                               HYST0250
      DDEL=DDE/DELY                                                             HYST0260
C ---- CALCULATE EXCURSION RATE AT EVERY ZERO CROSSING ---------------------- HYST0270
C                                                                               HYST0280
      IF (P*PLA.LE.0.) CALL DNE(0,DUC,EXCR,DELY,DA,ESE,PSE,PSEOLD,CSE)          HYST0290
C                                                                               HYST0300
C------------------------------------------------------------------------------ HYST0310
C   DUC     ---- ELEMENT DUCTILITY RATE                                        HYST0320
C   EXCR    ---- ELEMENT EXCURSION RATE                                        HYST0330
C   DELY    ---- ELEMENT YIELDING DISPLACEMENT                                 HYST0340
C   ESE     ---- ELEMENT ELASTIC STRAIN ENERGY                                 HYST0350
C   PSE     ---- ELEMENT PLASTIC STRAIN ENERGY                                 HYST0360
C   PSEOLD ---- ELEMENT PLASTIC STRAIN ENERGY AT LAST ZERO CROSSING            HYST0370
C   CSE     ---- ELEMENT CONSTANT STRAIN ENERGY                                HYST0380
C                                                                               HYST0390
C   DP      ---- ELEMENT AXIAL LOAD INCREMENT (KIPS)                           HYST0400
C   ESR     ---- EFFECTIVE SLENDER RATIO  (KL/R)                               HYST0410
C   YS      ---- YIELDING STRESS                                               HYST0420
C   AREA    ---- CROSS SECTION AREA (A)                                        HYST0430
C   EM      ---- ELASTIC MODULUS  (E)                                          HYST0440
C   DP      ---- ELEMENT AXIAL LOAD INCREMENT (KIPS)                           HYST0450
C   P       ---- CURRENT AXIAL LOAD (KIPS)                                     HYST0460
C   PL      ---- P - PY RATIO (P/PY)                                           HYST0470
C   PLA     ---- LAST AXIAL LOAD                                               HYST0480
C   DDEL    ---- DISPLACEMENT INCREMENT  (IN)                                  HYST0490
C   DE      ---- CURRENT AXIAL DISPLACEMENT  (IN)                              HYST0500
C   DEMAX   ---- MAXIMUM AXIAL DISPLACEMENT  (IN)                              HYST0510
C   DA      ---- CURRENT ABSOLUTE AXIAL DISPLACEMENT  (IN)                     HYST0520
C   DELA    ---- LAST AXIAL DISPLACEMENT                                       HYST0530
C   DDE     ---- ELEMENT DISPLACEMENT INCREMENT                                HYST0540
C   DEL     ---- DEL=DE/DELY                                                   HYST0550
C   DELY    ---- YIELDING DISPLACEMENT                                         HYST0560
C   V       ---- CURRENT VELOCITY                                              HYST0570
C   VLA     ---- LAST VELOCITY                                                 HYST0580
C   ES      ---- ELEMENT ELASTIC STIFFNESS                                     HYST0590
C   EXI     ---- UNLOADING EQUIVALENT STIFFNESS FOR CALCULATING ESE            HYST0600
C   PY      ---- TENSION YIELDING LOAD                                         HYST0610
C   EL      ---- ELEMENT LENGTH  (IN)                                          HYST0620
C   CC      ---  C SUB. C CONSTANT                                             HYST0630
C   PMAX    ---- BUCKING LOAD (KIPS)                                          HYST0640
C   DELMAX ---- BUCKING DISPLACEMENT (IN)                                      HYST0650
C   HA,HB,..,HC1,HD1..,HF ---- P/PY - DE/DELY PLOT'S DE/DELY COORDINATE        HYST0660
C   VA,VB,..,VC1,VD1..,VF ---- P/PY - DE/DELY PLOT'S P/PY COORDINATE           HYST0670
C   RSN     ---- RESIDUAL STRAIN                                               HYST0680
C   RULE    ---- RULE NUMBER                                                   HYST0690
C   CYCLE   ---- NO. OF CYCLE                                                  HYST0700
C   STIF    ---- ELEMENT STIFFNESS                                             HYST0710
C   REVRSC ---- LOGICAL INDEX FOR REVERSING CURVE OCCURRED                     HYST0720
C   IDUC    ---- 1: HC LESS THAN 12,  0 HC=12                                  HYST0730
C   REGIN   ---- REGIN (- DE +)  HA                                            HYST0740
C                             2: DE = (HH, HA)                                  HYST0750
C                             3: DE = (HB, HH)                                  HYST0760
C                             4: DE = (HC, HB)                                  HYST0770
C   PREGIN ---- PREVIOUS STEP REGIN ID NUMBER                                  HYST0780
C------------------------------------------------------------------------------ HYST0790
C                                                                               HYST0800
C INITIAL BUCKLING LOAD                                                         HYST0810
C                                                                               HYST0820
      IF( ESR .LT. CC .AND. CYCLE .EQ. 1 ) THEN                                 HYST0830
          PMAX=-YS*AREA                                                         HYST0840
      ELSE IF( ESR .GE. CC .AND. CYCLE .EQ. 1 ) THEN                            HYST0850
          PMAX=-PI*PI*AREA*EM/(ESR**2)                                          HYST0860
      ENDIF                                                                     HYST0870
C                                                                               HYST0880
C RESIDUAL ELOGATION FACTOR RFAC                                                HYST0890
C                                                                               HYST0900
      RFAC=1.4                                                                  HYST0910
C                                                                               HYST0920
C GENERATE CONTROL POINTS COORDINATE                                            HYST0930
C                                                                               HYST0940
C  (1)  POINT A                                                                 HYST0950
C                                                                               HYST0960
      IF( CYCLE .EQ. 1 ) THEN                                                   HYST0970
          VA=PMAX/PY                                                            HYST0980
          HA=PMAX/(ES*DELY)                                                     HYST0990
      ELSE IF( CYCLE .EQ. 2) THEN                                               HYST1000
         IF( ESR .LE. 40 ) THEN                                                 HYST1010
           PMAX=-0.2*PY                                                         HYST1020
         ELSE IF( ESR .GT. 40 ) THEN                                            HYST1030
           PMAX=-0.2*PY                                                         HYST1040
         ENDIF                                                                  HYST1050
      ELSE IF( CYCLE .GT. 2) THEN                                               HYST1060
         IF( ESR .LE. 40 ) THEN                                                 HYST1070
           PMAX=-0.2*PY                                                         HYST1080
         ELSE IF( ESR .GT. 40 ) THEN                                            HYST1090
           PMAX=-0.2*PY                                                         HYST1100
         ENDIF                                                                  HYST1110
      ENDIF                                                                     HYST1120
C                                                                               HYST1130
C  (2)  POINT B                                                                 HYST1140
C                                                                               HYST1150
      HB=-5.                                                                    HYST1160
      IF( ESR .LE. 40 ) THEN                                                    HYST1170
        VB=-0.15                                                                HYST1180
      ELSE IF( ESR .GT. 40 ) THEN                                               HYST1190
C                                                                               HYST1200
        VB=-0.15                                                                HYST1210
      ENDIF                                                                     HYST1220
C                                                                               HYST1230
C  (3)  POINT C                                                                 HYST1240
C                                                                               HYST1250
      IF( IDUC .EQ. 0.) THEN                                                    HYST1260
          HC=-12.                                                               HYST1270
        IF( ESR .LE. 40 ) THEN                                                  HYST1280
          VC=-0.1                                                               HYST1290
        ELSE IF( ESR  GT. 40 ) THEN                                             HYST1300
          VC=-0.1                                                               HYST1310
        ENDIF                                                                   HYST1320
      ENDIF                                                                     HYST1330
C                                                                               HYST1340
C  (4)  POINT E                                                                 HYST1350
C                                                                               HYST1360
          HE=1.*(RSN*EL/DELY)                                                   HYST1370
          VE=1.                                                                 HYST1380
C                                                                               HYST1390
C  (5)  POINT D1                                                                HYST1400
C                                                                               HYST1410
      SLOPE=(VE-VC)/(HE-HC)                                                     HYST1420
      HD1=(SLOPE*HC-VC)/(SLOPE-0.2679949)                                       HYST1430
      VD1=-0.267949*HD1                                                         HYST1440
C                                                                               HYST1450
C  (6)  POINT D                                                                 HYST1460
C                                                                               HYST1470
      IF( ESR .LE. 40 ) THEN                                                    HYST1480
        HD=HD1*31./40                                                           HYST1490
        VD=VD1*31./40                                                           HYST1500
      ELSE IF( ESR .GT. 40 ) THEN                                               HYST1510
C                                                                               HYST1520
        HD=HD1*20./40                                                           HYST1530
        VD=VD1*20./40                                                           HYST1540
      ENDIF                                                                     HYST1550
C                                                                               HYST1560
C DEFINE ENVELOPE SLOPE                                                         HYST1570
C                                                                               HYST1580
      SLOP1=1.                                                                  HYST1590
      SLOP2=(VB-VA)/(HB-HA)                                                     HYST1600
      SLOP3=(VC-VB)/(HC-HB)                                                     HYST1610
      SLOP4=(VD-VC)/(HD-HC)                                                     HYST1620
      SLOP5=(VE-VD)/(HE-HD)                                                     HYST1630
      SLOP6=0.0001                                                              HYST1640
C                                                                               HYST1650
C DEFINE ENVELOPE INTERMIDIATE POINTS                                           HYST1660
C                                                                               HYST1670
C  (1)  POINT H                                                                 HYST1680
C                                                                               HYST1690
      VH=(VB*SLOP2*(-VD+HD-HB))/(1.-SLOP2)                                      HYST1700
      HH=VH-VD+HD                                                               HYST1710
C                                                                               HYST1720
C  (2)  POINT C1                                                                HYST1730
C                                                                               HYST1740
      VC1=0.                                                                    HYST1750
      HC1=HC-(VC/SLOP4)                                                         HYST1760
C                                                                               HYST1770
C  (3)  POINT (PREF,DREF) : CALCULATE IN RULE NO. 4                             HYST1780
C                                                                               HYST1790
C                                                                               HYST1800
C  (4)  POINT (HREP,VREP) : CALCULATE IN RULE NO. 5                             HYST1810
C                                                                               HYST1820
C--- DEFINE CURRENT REGIN ZONE NUMBER                                           HYST1830
      PREGIN=REGIN                                                              HYST1840
      IF( DEL .GE. HA) REGIN=1                                                  HYST1850
      IF( DEL .GE. HH .AND. DEL .LT. HA ) REGIN=2                               HYST1860
      IF( DEL .GE. HB .AND. DEL .LT. HH ) REGIN=3                               HYST1870
      IF( DEL .GE. HC  AND. DEL .LT. HB ) REGIN=4                               HYST1880
C                                                                               HYST1890
C     GOEL'S HYSTORESIS LOOP RULES                                              HYST1900
C                                                                               HYST1910
C  (1) RULE 1                                                                   HYST1920
C                                                                               HYST1930
      IF( RULE .EQ. 1.) THEN                                                    HYST1940
          IF( P .GE. PMAX .AND. P .LT. PY ) THEN                                HYST1950
                STIF=SLOP1*ES                                                   HYST1960
                EXI=SLOP1*ES                                                    HYST1970
                IF( DEMAX .LT. 0 .AND. DE .LE. DEMAX ) THEN                     HYST1980
                  RSN=((0.55*ABS(DE)/40)*0.0002*ABS(DE)**2)*RFAC                HYST1990
                ENDIF                                                           HYST2000
          ELSE IF( P .GE  PY ) THEN                                             HYST2010
                STIF=SLOP6                                                      HYST2020
                RULE=6                                                          HYST2030
                EXI=SLOP1*ES                                                    HYST2040
          ELSE IF( P .LT. PMAX) THEN                                            HYST2050
                HA=DEL                                                          HYST2060
                VA=PL                                                           HYST2070
                SLOP2=(VB-VA)/(HB-HA)                                           HYST2080
                STIF=SLOP2*ES                                                   HYST2090
                RULE=2                                                          HYST2100
                IF( DEMAX .LT. 0 .AND. DE .LE. DEMAX ) THEN                     HYST2110
                  RSN=((0.55*ABS(DE)/40)*0.0002*ABS(DE)**2)*RFAC                HYST2120
                ENDIF                                                           HYST2130
                EXI=SLOP1*ES                                                    HYST2140
          ENDIF                                                                 HYST2150
          CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,EXI)                         HYST2160
          CALL DNE(1, DUC, EXCR, DELY, DA, ESE, PSE, PSEOLD, CSE )              HYST2170
      RETURN                                                                    HYST2180
      ENDIF                                                                     HYST2190
C                                                                               HYST2200
C  (2) RULE 2                                                                   HYST2210
C                                                                               HYST2220
```

```
      IF( RULE .EQ. 2.) THEN                                                    HYST2230
          IF( DEL .GE. HH) THEN                                                 HYST2240
              IF( DDE .LE. 0. ) THEN                                            HYST2250
                  IF(FREGIN .NE. REGIN ) REVRSC= .FALSE.                        HYST2260
                  STIP=SLOP2*ES                                                 HYST2270
                IF( DEMAX .LT. 0 .AND. DE .LE. DEMAX ) THEN                     HYST2280
                  RSN=((0.55*ABS(DE)/40)*0.0002*ABS(DE)**2)*RFAC                HYST2290
                ELSE IF( DEMAX .GT. 0. ) THEN                                   HYST2300
                  RSN=((0.55*ABS(DE)/40)*0.0002*ABS(DE)**2)*RFAC                HYST2310
                ENDIF                                                           HYST2320
C                 ----------------------------------------                      HYST2330
                  IF(.NOT. REVRSC )THEN                                         HYST2340
                     EXI=SLOP1*ES                                               HYST2350
                  ELSE IF( REVRSC )THEN                                         HYST2360
                     EXI=ES*((PTOP-PL)/(DTOP-DEL))                              HYST2370
                  ENDIF                                                         HYST2380
C                 ----------------------------------------                      HYST2390
              ELSE IF( DDE .GT. 0 )THEN                                         HYST2400
                 IF(.NOT. REVRSC )THEN                                          HYST2410
                    RULE=7                                                      HYST2420
                    STIP=SLOP1*ES                                               HYST2430
                    PLA=PL-SLOP2*DDEL                                           HYST2440
                    DLA=DEL-DDEL                                                HYST2450
                    DBOT=DLA                                                    HYST2460
                    PBOT=PLA                                                    HYST2470
                    PL=PLA+SLOP1*DDEL                                           HYST2480
                    P=PL+PY                                                     HYST2490
                    PTOP=VE+(SLOP5*(PBOT-VE-SLOP1*DBOT                          HYST2500
     *                    +SLOP1*HE)/(SLOP5-SLOP1))                            HYST2510
                    DTOP=(PBOT-VE+SLOP5*HE-SLOP1*DBOT)                          HYST2520
     *                   /(SLOP5-SLOP1)                                         HYST2530
                    EXI=ES*SLOP1                                                HYST2540
                 ELSE IF( REVRSC ) THEN                                         HYST2550
                    RULE=12                                                     HYST2560
                    DBOT=DEL                                                    HYST2570
                    PBOT=P/PY                                                   HYST2580
                    SLOP7=(PTOP-PBOT)/(DTOP-DBOT)                               HYST2590
                    STIP=SLOP7*ES                                               HYST2600
                    EXI=ES*SLOP7                                                HYST2610
                 ENDIF                                                          HYST2620
              ENDIF                                                             HYST2630
          ELSE IF(DEL .LT. HH .AND. DEL .GE. HB ) THEN                          HYST2640
              IF( DDE .LE. 0. ) THEN                                            HYST2650
                  STIP=SLOP2*ES                                                 HYST2660
                IF( DEMAX .LT. 0 .AND. DE .LE. DEMAX ) THEN                     HYST2670
                  RSN=((0.55*ABS(DE)/40)*0.0002*ABS(DE)**2)*RFAC                HYST2680
                ELSE IF( DEMAX .GT. 0 ) THEN                                    HYST2690
                  RSN=((0.55*ABS(DE)/40)*0.0002*ABS(DE)**2)*RFAC                HYST2700
                ENDIF                                                           HYST2710
C                 ----------------------------------------                      HYST2720
                  IF(.NOT. REVRSC )THEN                                         HYST2730
                     EXI=SLOP1*ES                                               HYST2740
                  ELSE IF( REVRSC )THEN                                         HYST2750
                     EXI=ES*((PTOP-PL)/(DTOP-DEL))                              HYST2760
                  ENDIF                                                         HYST2770
C                 ----------------------------------------                      HYST2780
              ELSE IF( DDE .GT. 0. )THEN                                        HYST2790
                  IF( .NOT. REVRSC ) THEN                                       HYST2800
                     RULE=8                                                     HYST2810
                     STIP=SLOP1*ES                                              HYST2820
                     PLA=PL-SLOP2*DDEL                                          HYST2830
                     DLA=DEL-DDEL                                               HYST2840
                     DBOT=DLA                                                   HYST2850
                     PBOT=PLA                                                   HYST2860
                     PL=PLA+SLOP1*DDEL                                          HYST2870
                     P=PL*PY                                                    HYST2880
                     PTOP=VD+(SLOP4*(PBOT-VD-SLOP1*DBOT                         HYST2890
     *                    +SLOP1*HD)/(SLOP4-SLOP1))                            HYST2900
                     DTOP=(PBOT-VD+SLOP4*HD-SLOP1*DBOT)                         HYST2910
     *                   /(SLOP4-SLOP1)                                         HYST2920
                     EXI=ES*SLOP1                                               HYST2930
                  ELSE IF( REVRSC ) THEN                                        HYST2940
                     RULE=11                                                    HYST2950
                     DBOT=DEL                                                   HYST2960
                     PBOT=P/PY                                                  HYST2970
                     PTOP=VD                                                    HYST2980
                     DTOP=HD                                                    HYST2990
                     SLOP7=(PTOP-PBOT)/(DTOP-DBOT)                              HYST3000
                     STIP=SLOP7*ES                                              HYST3010
                     EXI=ES*SLOP7                                               HYST3020
                  ENDIF                                                         HYST3030
              ENDIF                                                             HYST3040
          ELSE IF( DEL .LT. HB) THEN                                            HYST3050
              IF(FREGIN .NE. REGIN ) REVRSC= .FALSE.                            HYST3060
              RULE=3                                                            HYST3070
              EXI=ES*SLOP1                                                      HYST3080
          ENDIF                                                                 HYST3090
      IF( REVRSC ) IRV=1                                                        HYST3100
      IF ( .NOT. REVRSC) IRV=0                                                  HYST3110
      CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,EXI)                             HYST3120
            IF( ABS(DEMAX) .LT. ABS(DE) ) THEN                                  HYST3130
              CALL DNE(1, DUC, EXCR, DELY, DA, ESE, PSE, PSEOLD, CSE )          HYST3140
              DEMAX=DE                                                          HYST3150
            ENDIF                                                               HYST3160
      RETURN                                                                    HYST3170
      ENDIF                                                                     HYST3180
C                                                                               HYST3190
C  (3) RULE 3                                                                   HYST3200
C                                                                               HYST3210
      IF( RULE .EQ. 3.) THEN                                                    HYST3220
          IF( DEL .GT. HC) THEN                                                 HYST3230
              IF( DDE .LE. 0. ) THEN                                            HYST3240
                  IF(FREGIN .NE. REGIN ) REVRSC= .FALSE.                        HYST3250
                  STIP=SLOP3*ES                                                 HYST3260
                IF( DEMAX .LT. 0 .AND. DE .LE. DEMAX ) THEN                     HYST3270
                  RSN=((0.55*ABS(DE)/40)*0.0002*ABS(DE)**2)*RFAC                HYST3280
                ELSE IF( DEMAX .GT. 0 ) THEN                                    HYST3290
                  RSN=((0.55*ABS(DE)/40)*0.0002*ABS(DE)**2)*RFAC                HYST3300
                ENDIF                                                           HYST3310
C                 ----------------------------------------                      HYST3320
                  IF(.NOT. REVRSC )THEN                                         HYST3330
                     EXI=SLOP1*ES                                               HYST3340
                  ELSE IF( REVRSC )THEN                                         HYST3350
                     EXI=ES*((PTOP-PL)/(DTOP-DEL))                              HYST3360
                  ENDIF                                                         HYST3370
C                 ----------------------------------------                      HYST3380
              ELSE IF( DDE .GT. 0. )THEN                                        HYST3390
                IF( DEL .GT. -12. ) THEN                                        HYST3400
                  IF( .NOT. REVRSC )THEN                                        HYST3410
                     RULE=9                                                     HYST3420
                     STIP=SLOP1*ES                                              HYST3430
                     PLA=PL-SLOP2*DDEL                                          HYST3440
                     DLA=DEL-DDEL                                               HYST3450
                     DBOT=DLA                                                   HYST3460
                     PBOT=PLA                                                   HYST3470
                     PL=PLA+SLOP1*DDEL                                          HYST3480
                     P=PL*PY                                                    HYST3490
                     PTOP=VD+(SLOP4*(PBOT-VD-SLOP1*DBOT                         HYST3500
     *                    +SLOP1*HD)/(SLOP4-SLOP1))                            HYST3510
                     DTOP=(PBOT-VD+SLOP4*HD-SLOP1*DBOT)                         HYST3520
     *                   /(SLOP4-SLOP1)                                         HYST3530
                     EXI=ES*SLOP1                                               HYST3540
                  ELSE IF( REVRSC ) THEN                                        HYST3550
                     RULE=10                                                    HYST3560
                     DBOT=DEL                                                   HYST3570
                     PBOT=P/PY                                                  HYST3580
                     SLOP7=(PTOP-PBOT)/(DTOP-DBOT)                              HYST3590
                     STIP=SLOP7*ES                                              HYST3600
                     EXI=ES*SLOP7                                               HYST3610
                  ENDIF                                                         HYST3620
                ELSE IF ( DEL .LE. -12  ) THEN                                  HYST3630
                  REVRSC= .FALSE.                                               HYST3640
                  RULE=4                                                        HYST3650
                  STIP=SLOP4*ES                                                 HYST3660
                  EXI=ES*SLOP4                                                  HYST3670
                ENDIF                                                           HYST3680
                  ENDIF                                                         HYST3690
              ELSE IF( DEL .LE. HC ) THEN                                       HYST3700
                IF( DDE .LT. 0 ) THEN                                           HYST3710
                  RULE=3                                                        HYST3720
                  HC=DEL                                                        HYST3730
                  VC=PL                                                         HYST3740
                  IDUC=1.                                                       HYST3750
                  STIP=-SLOP6*ES                                                HYST3760
                  EXI=ES*((VD-VC)/(HD-HC))                                      HYST3770
                        IF( DEMAX .LT. 0 .AND. DE .LE. DEMAX ) THEN             HYST3780
                          RSN=((0.55*ABS(DE)/ESR)*0.0002*ABS(DE)**2)*RFAC       HYST3790
                        ELSE IF( DEMAX .GT. 0 ) THEN                            HYST3800
                          RSN=((0.55*ABS(DE)/ESR)*0.0002*ABS(DE)**2)*RFAC       HYST3810
                        ENDIF                                                   HYST3820
                ELSE IF( DDE .GT. 0 ) THEN                                      HYST3830
                  RULE=4                                                        HYST3840
                  STIP=SLOP4*ES                                                 HYST3850
                  EXI=ES*SLOP4                                                  HYST3860
                ENDIF                                                           HYST3870
              ENDIF                                                             HYST3880
      IF( REVRSC ) IRV=1.                                                       HYST3890
      IF ( .NOT. REVRSC) IRV=0                                                  HYST3900
      CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,EXI)                             HYST3910
            IF( ABS(DEMAX) .LT. ABS(DE) ) THEN                                  HYST3920
              CALL DNE(1, DUC, EXCR, DELY, DA, ESE, PSE, PSEOLD, CSE )          HYST3930
              DEMAX=DE                                                          HYST3940
            ENDIF                                                               HYST3950
      RETURN                                                                    HYST3960
      ENDIF                                                                     HYST3970
C                                                                               HYST3980
C  (4) RULE 7                                                                   HYST3990
C                                                                               HYST4000
      IF( RULE .EQ. 7.) THEN                                                    HYST4010
          IF( DEL .GT. DBOT .AND. DEL .LT. DTOP .AND. P .GT. PMAX) THEN         HYST4020
                STIP=SLOP1*ES                                                   HYST4030
                EXI=ES*SLOP1                                                    HYST4040
          ELSE IF(DEL .LT. DBOT .OR. P .LE. PMAX) THEN                          HYST4050
                IF( P .LE. PMAX ) THEN                                          HYST4060
                    REVRSC= .FALSE.                                             HYST4070
                    PTOP=C                                                      HYST4080
                    DTOP=0                                                      HYST4090
                    PBOT=PL                                                     HYST4100
                    DBOT=DEL                                                    HYST4110
                    HA=DEL                                                      HYST4120
                    VA=PL                                                       HYST4130
                    SLOPE2=(VB-VA)/(HB-HA)                                      HYST4140
                ENDIF                                                           HYST4150
                STIP=SLOP2*ES                                                   HYST4160
                RULE=2                                                          HYST4170
                PL=PBOT-(DBOT-DEL)*SLOP2                                        HYST4180
                P=PL*PY                                                         HYST4190
                EXI=ES*((PTOP-PL)/(DTOP-DEL))                                   HYST4200
          ELSE IF(DEL .GT. DTOP .AND. PL .LT. VE) THEN                          HYST4210
                STIP=SLOP5*ES                                                   HYST4220
                RULE=5                                                          HYST4230
                EXI=ES*SLOP1                                                    HYST4240
          ELSE IF(DEL .GT. DTOP .AND. PL .GE. VE) THEN                          HYST4250
                STIP=SLOP6*ES                                                   HYST4260
                RULE=6                                                          HYST4270
                EXI=ES*SLOP1                                                    HYST4280
          ENDIF                                                                 HYST4290
      IF( REVRSC ) IRV=1.                                                       HYST4300
      IF ( .NOT. REVRSC) IRV=0.                                                 HYST4310
      CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,EXI)                             HYST4320
      RETURN                                                                    HYST4330
      ENDIF                                                                     HYST4340
C                                                                               HYST4350
C  (5) RULE 8                                                                   HYST4360
C                                                                               HYST4370
      IF( RULE .EQ. 8.) THEN                                                    HYST4380
          IF( DEL .GT. DBOT .AND. DEL .LT. DTOP ) THEN                          HYST4390
                STIP=SLOP1*ES                                                   HYST4400
                EXI=ES*SLOP1                                                    HYST4410
          ELSE IF(DEL .LT. DBOT ) THEN                                          HYST4420
                STIP=SLOP2*ES                                                   HYST4430
                RULE=2                                                          HYST4440
                PL=PBOT-(DBOT-DEL)*SLOP2                                        HYST4450
                P=PL*PY                                                         HYST4460
                EXI=ES*((PTOP-PL)/(DTOP-DEL))                                   HYST4470
          ELSE IF(DEL .GT. DTOP ) THEN                                          HYST4480
                STIP=SLOP4*ES                                                   HYST4490
                RULE=4                                                          HYST4500
                EXI=ES*SLOP1                                                    HYST4510
          ENDIF                                                                 HYST4520
      IF( REVRSC ) IRV=1.                                                       HYST4530
      IF ( .NOT. REVRSC) IRV=0.                                                 HYST4540
      CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,EXI)                             HYST4550
      RETURN                                                                    HYST4560
      ENDIF                                                                     HYST4570
C                                                                               HYST4580
C  (6) RULE 9                                                                   HYST4590
C                                                                               HYST4600
      IF( RULE .EQ. 9.) THEN                                                    HYST4610
          IF( DEL .GT. DBOT .AND. DEL .LT. DTOP ) THEN                          HYST4620
                STIP=SLOP1*ES                                                   HYST4630
                EXI=ES*SLOP1                                                    HYST4640
          ELSE IF(DEL .LT. DBOT ) THEN                                          HYST4650
                STIP=SLOP3*ES                                                   HYST4660
                RULE=3                                                          HYST4670
                PL=PBOT-(DBOT-DEL)*SLOP3                                        HYST4680
                P=PL*PY                                                         HYST4690
                EXI=ES*((PTOP-PL)/(DTOP-DEL))                                   HYST4700
          ELSE IF(DEL .GT. DTOP ) THEN                                          HYST4710
                STIP=SLOP4*ES                                                   HYST4720
                RULE=4                                                          HYST4730
                EXI=ES*SLOP1                                                    HYST4740
          ENDIF                                                                 HYST4750
      IF( REVRSC ) IRV=1.                                                       HYST4760
      IF ( .NOT. REVRSC) IRV=0.                                                 HYST4770
      CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,EXI)                             HYST4780
      RETURN                                                                    HYST4790
      ENDIF                                                                     HYST4800
C                                                                               HYST4810
C  (7) RULE 4                                                                   HYST4820
C                                                                               HYST4830
      IF( RULE .EQ. 4.) THEN                                                    HYST4840
            IF (DEL .GE. HC1 .AND. DEL .LT. HD ) THEN                           HYST4850
              IF(DDE .GE. 0. ) THEN                                             HYST4860
                  STIP=SLOP4*ES                                                 HYST4870
                  EXI=ES*SLOP1                                                  HYST4880
              ELSE IF( DDE .LT. 0. ) THEN                                       HYST4890
                  PREF=VD+(SLOP4*(VB-VD-SLOP1*HB                                HYST4900
     *                 +SLOP1*HD)/(SLOP4-SLOP1))                                HYST4910
                  DREF=(VB-VD+SLOP4*HD-SLOP1*HB)/(SLOP4-SLOP1)                  HYST4920
                  IF( DEL .LT. DREF) THEN                                       HYST4930
                      PTOP=P/PY                                                 HYST4940
                      DTOP=DEL                                                  HYST4950
                      DBOT=(SLOP3*HB-VB+PTOP-SLOP1*DEL)/(SLOP3-SLOP1)           HYST4960
                      PBOT=VB+SLOP3*(DBOT-HB)                                   HYST4970
                      STIP=SLOP1*ES                                             HYST4980
                      RULE=9                                                    HYST4990
                      REVRSC= .TRUE.                                            HYST5000
                      EXI=ES*SLOP1                                              HYST5010
                  ELSE IF( DEL .GE. DREF) THEN                                  HYST5020
                      PTOP=P/PY                                                 HYST5030
                      DTOP=DEL                                                  HYST5040
                      DBOT=(SLOP2*HA-VA+PTOP-SLOP1*DEL)/(SLOP2-SLOP1)           HYST5050
                      PBOT=VA+SLOP2*(DBOT-HA)                                   HYST5060
                      STIP=SLOP1*ES                                             HYST5070
                      RULE=8                                                    HYST5080
                      REVRSC= .TRUE.                                            HYST5090
                      EXI=ES*SLOP1                                              HYST5100
                  ENDIF                                                         HYST5110
              ENDIF                                                             HYST5120
            ELSE IF( DEL .GT. HD .AND. DDE .GT. 0. ) THEN                       HYST5130
                  STIP=SLOP5*ES                                                 HYST5140
```

```
                RULE=5                                                  HYST5150
                EKI=ES*SLOP1                                            HYST5160
            ELSE IF( DEL .LT. HC1 .AND. DEL .GE. HC ) THEN              HYST5170
                REVRSC= .FALSE.                                         HYST5180
                RULE=6                                                  HYST5190
                STIF=SLOP6*ES                                           HYST5200
                EKI=ES*SLOP6                                            HYST5210
            ELSE IF( DEL .LT. HC ) THEN                                 HYST5220
                REVRSC= .FALSE.                                         HYST5230
                RULE=3                                                  HYST5240
                HC=DEL                                                  HYST5250
                VC=PL                                                   HYST5260
                IDUC=1.                                                 HYST5270
                STIF=.SLOP6*ES                                          HYST5280
                EKI=ES*((VD-VC)/(HD-HC))                                HYST5290
              ENDIF                                                     HYST5300
        IF( REVRSC ) IRV=1.                                             HYST5310
        IF ( .NOT. REVRSC) IRV=0.                                       HYST5320
        CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,EKI)                   HYST5330
            IF( ABS(DEMAX) .LT. ABS(DE) ) THEN                          HYST5340
              CALL DNE(1, DUC, EXCR, DELY, DA, ESE, PSE, PSEOLD, CSE )  HYST5350
              DEMAX=DE                                                  HYST5360
            ENDIF                                                       HYST5370
        RETURN                                                          HYST5380
        ENDIF                                                           HYST5390
C                                                                       HYST5400
C  (8) RULE 5                                                           HYST5410
C                                                                       HYST5420
        IF( RULE .EQ. 5.) THEN                                          HYST5430
            IF (DEL .LT. HE ) THEN                                      HYST5440
                IF(DDE .GE. 0. ) THEN                                   HYST5450
                    STIF=SLOP5*ES                                       HYST5460
                    EKI=ES*SLOP1                                        HYST5470
                ELSE IF( DDE .LT. 0. ) THEN                             HYST5480
                      BREF=VA-SLOP1*HA                                  HYST5490
                      HREF=(VD-SLOP5*HD-BREF)/(SLOP1-SLOP5)             HYST5500
                      IF( DEL .GE. HREF ) CYCLE=CYCLE+1                 HYST5510
                      PTOP=P/PY                                         HYST5520
                      DTOP=DEL                                          HYST5530
                      DBOT=(SLOP2*HA-VA+PTOP*SLOP1*DEL)/(SLOP2-SLOP1)   HYST5540
                      PBOT=VA-SLOP2*(DBOT-HA)                           HYST5550
                      STIF=SLOP1*ES                                     HYST5560
                      RULE=7                                            HYST5570
                      REVRSC= .TRUE.                                    HYST5580
                      EKI=ES*SLOP1                                      HYST5590
                ENDIF                                                   HYST5600
            ELSE IF( DEL .GE. HE .AND. DDE .GT. 0. ) THEN               HYST5610
                STIF=SLOP6                                              HYST5620
                RULE=6                                                  HYST5630
                EKI=ES*SLOP1                                            HYST5640
            ENDIF                                                       HYST5650
        IF( REVRSC ) IRV=1.                                             HYST5660
        IF ( .NOT. REVRSC) IRV=0.                                       HYST5670
        CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,EKI)                   HYST5680
            IF( ABS(DEMAX) .LT. ABS(DE) ) THEN                          HYST5690
              CALL DNE(1, DUC, EXCR, DELY, DA, ESE, PSE, PSEOLD, CSE )  HYST5700
              DEMAX=DE                                                  HYST5710
            ENDIF                                                       HYST5720
        RETURN                                                          HYST5730
        ENDIF                                                           HYST5740
C                                                                       HYST5750
C  (9) RULE 6                                                           HYST5760
C                                                                       HYST5770
        IF( RULE .EQ. 6 ) THEN                                          HYST5780
            IF(DDE .GE. 0.) THEN                                        HYST5790
                STIF=SLOP6                                              HYST5800
                EKI=ES*SLOP1                                            HYST5810
            ELSE IF(DDE .LT. 0.) THEN                                   HYST5820
                STIF=SLOP1*ES                                           HYST5830
                RULE=1                                                  HYST5840
                CYCLE=CYCLE+1.                                          HYST5850
                REVRSC= .FALSE.                                         HYST5860
                EKI=ES*SLOP1                                            HYST5870
            ENDIF                                                       HYST5880
        IF( REVRSC ) IRV=1.                                             HYST5890
        IF ( .NOT. REVRSC) IRV=0.                                       HYST5900
        CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,EKI)                   HYST5910
            IF( ABS(DEMAX) .LT. ABS(DE) ) THEN                          HYST5920
              CALL DNE(1, DUC, EXCR, DELY, DA, ESE, PSE, PSEOLD, CSE )  HYST5930
              DEMAX=DE                                                  HYST5940
            ENDIF                                                       HYST5950
        RETURN                                                          HYST5960
        ENDIF                                                           HYST5970
C                                                                       HYST5980
C  (10) RULE 10                                                         HYST5990
C                                                                       HYST6000
        IF( RULE .EQ. 10.) THEN                                         HYST6010
            IF( DEL .GT. DBOT .AND. DEL .LT. DTOP ) THEN                HYST6020
                STIF=SLOP7*ES                                           HYST6030
                EKI=ES*SLOP7                                            HYST6040
            ELSE IF(DEL .LT. DBOT ) THEN                                HYST6050
                STIF=SLOP3*ES                                           HYST6060
                RULE=3                                                  HYST6070
                REVRSC= .FALSE.                                         HYST6080
                PTOP=0                                                  HYST6090
                DTOP=0                                                  HYST6100
                PBOT=0                                                  HYST6110
                DBOT=0                                                  HYST6120
                EKI=ES*((PTOP-PL)/(DTOP-DEL))                           HYST6130
            ELSE IF(DEL .GT. DTOP ) THEN                                HYST6140
                STIF=SLOP4*ES                                           HYST6150
                RULE=4                                                  HYST6160
                EKI=ES*SLOP1                                            HYST6170
            ENDIF                                                       HYST6180
        IF( REVRSC ) IRV=1.                                             HYST6190
        IF ( .NOT. REVRSC) IRV=0.                                       HYST6200
        CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,EKI)                   HYST6210
        RETURN                                                          HYST6220
        ENDIF                                                           HYST6230
C                                                                       HYST6240
C  (11) RULE 11                                                         HYST6250
C                                                                       HYST6260
        IF( RULE .EQ. 11 ) THEN                                         HYST6270
            IF( DEL .GT. DBOT .AND. DEL .LT. DTOP ) THEN                HYST6280
                STIF=SLOP7*ES                                           HYST6290
                EKI=ES*SLOP7                                            HYST6300
            ELSE IF(DEL .LT. DBOT ) THEN                                HYST6310
                STIF=SLOP2*ES                                           HYST6320
                RULE=2                                                  HYST6330
                REVRSC= .FALSE.                                         HYST6340
                PTOP=0                                                  HYST6350
                DTOP=0                                                  HYST6360
                PBOT=0                                                  HYST6370
                DBOT=0                                                  HYST6380
                EKI=ES*((VD-PL)/(HD-DEL))                               HYST6390
            ELSE IF(DEL .GT. DTOP ) THEN                                HYST6400
                STIF=SLOP4*ES                                           HYST6410
                RULE=4                                                  HYST6420
                EKI=ES*SLOP1                                            HYST6430
            ENDIF                                                       HYST6440
        IF( REVRSC ) IRV=1.                                             HYST6450
        IF ( .NOT. REVRSC) IRV=0.                                       HYST6460
        CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,EKI)                   HYST6470
        RETURN                                                          HYST6480
        ENDIF                                                           HYST6490
C                                                                       HYST6500
C  (12) RULE 12                                                         HYST6510
C                                                                       HYST6520
        IF( RULE .EQ. 12.) THEN                                         HYST6530
            IF( DEL .GT. DBOT .AND. DEL .LT. DTOP ) THEN                HYST6540
                STIF=SLOP7*ES                                           HYST6550
                EKI=ES*SLOP7                                            HYST6560
            ELSE IF(DEL .LT. DBOT ) THEN                                HYST6570
                STIF=SLOP2*ES                                           HYST6580
                RULE=2                                                  HYST6590
                REVRSC= .FALSE.                                         HYST6600
                PTOP=0                                                  HYST6610
                DTOP=0                                                  HYST6620
                PBOT=0                                                  HYST6630
                DBOT=0                                                  HYST6640
                EKI=ES*((PTOP-PL)/(DTOP-DEL))                           HYST6650
            ELSE IF(DEL .GT. DTOP ) THEN                                HYST6660
                STIF=SLOP5*ES                                           HYST6670
                RULE=5                                                  HYST6680
                EKI=ES*SLOP1                                            HYST6690
            ENDIF                                                       HYST6700
        ENDIF                                                           HYST6710
        IF( REVRSC ) IRV=1.                                             HYST6720
        IF ( .NOT. REVRSC) IRV=0.                                       HYST6730
        CALL STRENG(ESE,PSE,PSEOLD,P,PLA,DE,DELA,EKI)                   HYST6740
        RETURN                                                          HYST6750
        END                                                             HYST6760
C-----------------------------------------------------------------------SOL00010
      SUBROUTINE SOL01                                                  SOL00020
$INCLUDE:'ZCOMM'                                                        SOL00030
C                                                                       SOL00040
      READ(105,*)  NLOAD,MAXELD                                         SOL00050
      WRITE(108,1) TITLE(1),TITLE(2),NLOAD                              SOL00060
    1 FORMAT('1 STRUCTURE.....: ',A,/'  SOLUTION..... ',A,//            SOL00070
     &       1X,' SOLUTION #1, STATIC - ELASTIC ANALYSIS '/             SOL00080
     &       1X,'--------------------------------------'//              SOL00090
     &       1X,' NUMBER OF LOAD CASES ..... ...',I5)                   SOL00100
C                                                                       SOL00110
      IF (NZ(IZXGDT-3).EQ.2) WRITE(108,2)                               SOL00120
    2 FORMAT(1X,' GEOMETRIC STIFFNESS IS NOT INCLUDED. ',               SOL00130
     &       ' USE KGDATA(3)=1 TO INCLUDE GEOMETRIC STIFFNESS.')        SOL00140
C                                                                       SOL00150
C      ---- INITIALIZE STORAGE FOR STIFFNESS, LOAD AND DISPL .. -----  SOL00160
      ELSTIC= TRUE.                                                     SOL00170
C                                                                       SOL00180
C      -- - INITIALIZE STORAGE FOR STIFFNESS, LOAD AND DISPL.   ----   SOL00190
C            Z(IZSTIF) - LOCATION OF THE STIFFNESS MATRIX               SOL00200
C            Z(IZLOAD) - LOCATION OF THE LOAD MATRIX                    SOL00210
C            Z(IZLOAD) - LOCATION OF THE DISPLACEMENT MATRIX            SOL00220
      IF (IZLOAD.EQ.0) THEN                                             SOL00230
          IZLOAD = IZ                                                   SOL00240
          IZ     = IZ+NDOF*NLOAD                                        SOL00250
          ENDIF                                                         SOL00260
      IF (IZDISP.EQ.0) THEN                                             SOL00270
          IZDISP = IZ                                                   SOL00280
          IZ     = IZ+NDOF*NLOAD                                        SOL00290
          ENDIF                                                         SOL00300
       IF (IZVEL .EQ.0) THEN                                            SOL00310
          IZVEL  = IZ                                                   SOL00320
          IZ     = IZ+NDOF                                              SOL00330
          ENDIF                                                         SOL00340
          DO 3 I=1,NDOF                                                 SOL00350
    3     Z(I+IZVEL -1) =0.                                             SOL00360
      IF (IZDVEL.EQ.0) THEN                                             SOL00370
          IZDVEL = IZ                                                   SOL00380
          IZ     = IZ+NDOF                                              SOL00390
          ENDIF                                                         SOL00400
          DO 4 I=1,NDOF                                                 SOL00410
    4     Z(I+IZDVEL-1) =0                                              SOL00420
      IF (IZFUB .EQ 0) THEN                                             SOL00430
          IZFUB  = IZ                                                   SOL00440
          IZ     = IZ+NDOF                                              SOL00450
          DO 5 I=1,NDOF                                                 SOL00460
    5     Z(I-IZFUB  1) =0.                                             SOL00470
          ENDIF                                                         SOL00480
C                                                                       SOL00490
C      ---- SET UP CONSTANTS - ---------------                          SOL00500
      L1=1                                                              SOL00510
      L2=NCOND +NFREE                                                   SOL00520
      L3=NCOND +NFREE+1                                                 SOL00530
      L4=NDOF                                                           SOL00540
      IF (MAXELD.GT.0) THEN                                             SOL00550
          IZIELD=IZ                                                     SOL00560
          IZ=IZ + MAXELD*5                                              SOL00570
          IZELD=IZ                                                      SOL00580
          IZ=IZ + MAXELD*12                                             SOL00590
      ENDIF                                                             SOL00600
      IW=IZ+1                                                           SOL00610
      IMD=NZ(IZLM+L4)                                                   SOL00620
      CALL SOL01A                                                       SOL00630
     &    (L1,L2,L3,L4,IMD,NLOAD,MAXELD,NELD,NNODE,NCOS,NELMT,          SOL00640
     &     IZDISP,IZLOAD,IBUG,MAXNOD,TITLE,STEPID,ESE,                  SOL00650
     &     Z(IZSTIF), Z(IZLOAD), Z(IZDISP),NZ(IZIELD), Z(IZELD ),       SOL00660
     &     Z(IW    ),NZ(IMD   ),NZ(IZIDP ),Z(IZCNST),                   SOL00670
     &     NZ(IZJFLG), Z(IZCORD), Z(IZCOS ),NZ(IZJCOS),SUMRCT)          SOL00680
C                                                                       SOL00690
      RETURN                                                            SOL00700
      END                                                               SOL00710
C                                                                       SOL00010
C-----------------------------------------------------------------------SOL00020
C                                                                       SOL00030
      SUBROUTINE SOL01A                                                 SOL00040
     &    (L1,L2,L3,L4,IMD,NLOAD,MAXELD,NELD,NNODE,NCOS,NELMT,          SOL00050
     &     IZDISP,IZLOAD,IBUG,MAXNOD,TITLE,STEPID,ESE,                  SOL00060
     &     STIF , LOAD , DISP , IELD , ELD  ,                           SOL00070
     &      WORK , MD   ,ID   , IDOF , CONST,                           SOL00080
     &     JTFLG, COORD,COSINE, JTCOS,SUMRCT)                           SOL00090
      CHARACTER*(*) TITLE(2),STEPID                                     SOL00100
      LOGICAL PRINT,DSPPLG,HEAD,AXIAL,BTEST,BUG,BUG3                    SOL00110
      CHARACTER*80 NAME                                                 SOL00120
      REAL LOAD                                                         SOL00130
      DIMENSION RINPUT(100),SUMRCT(6)                                   SOL00140
      DIMENSION STIF(IMD),LOAD(L4,NLOAD),DISP(L4,NLOAD)                 SOL00150
      DIMENSION IELD(MAXELD,5),ELD(MAXELD,12),WORK(1),MD(L4+1)          SOL00160
      DIMENSION COORD(MAXNOD,3),JTFLG(MAXNOD),JTCOS(MAXNOD),            SOL00170
     &          IDOF(MAXNOD,6) ,COSINE(3,3,NCOS),CONST(MAXNOD,6)        SOL00180
      BUG=BTEST(IBUG,5)                                                 SOL00190
      BUG3=BTEST(IBUG,3)                                                SOL00200
C                                                                       SOL00210
C*----------------------                                               SOL00220
C* INPUT LOADING DATA --                                               SOL00230
C*----------------------                                               SOL00240
C                                                                       SOL00250
C      ----- ZERO LOAD AND DISPLACEMENT MATRIX ----- ------- -------    SOL00260
      DO  6 I=1,L4                                                      SOL00270
      DO  6 J=1,NLOAD                                                   SOL00280
         LOAD(I,J)=0                                                    SOL00290
    6    DISP(I,J)=0                                                   SOL00300
C                                                                       SOL00310
C      ----- INPUT JOINT LOADS ------------------                      SOL00320
C      JOINT LOADS AND SUPPORT DISPLACEMENTS ARE STORED IN THE          SOL00330
C      DISPLACEMENT MATRIX                                              SOL00340
      CALL JOILOA(BUG3,MAXNOD,L4,NNODE,NLOAD,L2,ID,IDOF,CONST,DISP,     SOL00350
     &            DSPPLG,JTFLG,TITLE,.FALSE.)                           SOL00360
C                                                                       SOL00370
C      ----- INPUT ELEMENT LOADS ------------------                     SOL00380
C      ELEMENT LOADS ARE STORED IN THE LOAD MATRIX                      SOL00390
      IF (MAXELD.GT.0) THEN                                             SOL00400
          CALL ELELOA(BUG3,L3,NELMT,MAXELD,NELD,L4,NLOAD,               SOL00410
     &            LOAD,IELD,ELD,IZDISP,IZLOAD,NAME,TITLE,.FALSE.)       SOL00420
C                                                                       SOL00430
C        ..ADD ELEMENT LOADS ON FREE JOINTS TO THE JOINT LOADS ON FREE   SOL00440
C        ..JOINTS, STORE IN THE DISPLACEMENT MATRIX                      SOL00450
C        ..SAVE ELEMENT LOADS ON ALL JOINTS IN LOAD MATRIX, USED LATER   SOL00460
C        ..FOR CALCULATING THE REACTIONS.                                SOL00470
         DO 10 I=L1,L2                                                  SOL00480
         DO 10 J=1,NLOAD                                                SOL00490
   10      DISP(I,J)=DISP(I,J)+LOAD(I,J)                                SOL00500
C                                                                       SOL00510
C        ...SWAP THE SIGN OF THE FEP AT REACTIONS                       SOL00520
         DO 15 I=L3,L4                                                  SOL00530
         DO 15 J=1,NLOAD                                                SOL00540
   15      LOAD(I,J)= -LOAD(I,J)                                        SOL00550
C                                                                       SOL00560
      ENDIF                                                             SOL00570
C*                                                                      SOL00580
C*------------------------                                             SOL00590
```

```
C- FORMULATE STIFFNESS ----                                              SOL00600
C-------------------------                                               SOL00610
      CALL FORM(1)                                                       SOL00620
C                                                                        SOL00630
C-----------------------------------------                               SOL00640
C- MODIFY LOADS FOR RESTRAINT DISPLACEMENTS ---                          SOL00650
C-----------------------------------------                               SOL00660
      IF (DSPFLG) THEN                                                   SOL00670
        IOPT=1                                                           SOL00680
        CALL REACTN(IOPT,BUG,NNODE,L1,L2,L3,L4,DISP,DISP,                SOL00690
     &       MD,STIF,IMD,L4,NLOAD,1.0,                                   SOL00700
     &       MAXNOD,IDOF,COORD,ID,TITLE,STEPID,.FALSE.,                  SOL00710
     &       NCOS,COSINE,JTCOS,JTPLG,SUMRCT)                             SOL00720
      ENDIF                                                              SOL00730
C                                                                        SOL00740
C--------------------------                                              SOL00750
C- SOLVE FOR FREE DISPL -----                                            SOL00760
C--------------------------                                              SOL00770
C                                                                        SOL00780
      IF (BTEST(IBUG,5)) THEN                                            SOL00790
       CALL LMATRX(STIF,MD,L4,IMD,'GLOBAL STIFFNESS DUMP-7')             SOL00800
       CALL WMATRX(DISP,L4,NLOAD,'LOAD MATRIX  DUMP-7')                  SOL00810
      ENDIF                                                              SOL00820
C                                                                        SOL00830
      DO 12 IOPT=1,3,2                                                   SOL00840
       CALL MSOLL(IOPT,BTEST(IBUG,6),L2,IMD,L4,NLOAD,L1,L2,MD,STIF,      SOL00850
     &            DISP,WORK,.FALSE.,WORK,WORK,L4,.FALSE.,WORK,WORK,L4)  SOL00860
12    CONTINUE                                                           SOL00870
C                                                                        SOL00880
      IF (BTEST(IBUG,5))                                                 SOL00890
     &   CALL WMATRX(DISP,L4,NLOAD,'DISPLACEMENT  DUMP-7')               SOL00900
C                                                                        SOL00910
C------------------------                                                SOL00920
C- SOLVE FOR REACTIONS ---                                               SOL00930
C------------------------                                                SOL00940
      IOPT=2                                                             SOL00950
      CALL REACTN(IOPT,BUG,NNODE,L1,L2,L3,L4,LOAD,DISP,                  SOL00960
     &            MD,STIF,IMD,L4,NLOAD,1.0,                              SOL00970
     &            MAXNOD,IDOP,COORD,ID,TITLE,STEPID, FALSE.,             SOL00980
     &            NCOS,COSINE,JTCOS,JTPLG,SUMRCT)                        SOL00990
      IF (BTEST(IBUG,5))                                                 SOL01000
     &  CALL WMATRX(LOAD,L4,NLOAD,'FINAL LOADS AND REACTIONS')           SOL01010
C                                                                        SOL01020
C------------------------------                                          SOL01030
C- PRINT GLOBAL DISPLACEMENTS ---                                        SOL01040
C------------------------------                                          SOL01050
      CALL JOIDSP(1,MAXNOD,L4,NNODE,NLOAD,L2,ID,IDOF,CONST,DISP,         SOL01060
     &            TITLE,'DISPLACEMENTS',STEPID,.TRUE.,                   SOL01070
     &            NCOS,COSINE,JTCOS)                                     SOL01080
C                                                                        SOL01090
C----------------------------                                            SOL01100
C- PRINT GLOBAL REACTIONS ---                                            SOL01110
C----------------------------                                            SOL01120
      IOPT=3                                                             SOL01130
      CALL REACTN(IOPT,.TRUE ,NNODE,L1,L2,L3,L4,LOAD,DISP,               SOL01140
     &            MD,STIF,IMD,L4,NLOAD,1.0,                              SOL01150
     &            MAXNOD,IDOP,COORD,ID,TITLE,STEPID,.TRUE.,              SOL01160
     &            NCOS,COSINE,JTCOS,JTPLG,SUMRCT)                        SOL01170
C                                                                        SOL01180
C---------------------------------------                                 SOL01190
C- CALCULATE AND PRINT MEMBER FORCES ---                                 SOL01200
C---------------------------------------                                 SOL01210
      IOPT=4                                                             SOL01220
      LSTTYP=0                                                           SOL01230
      PRINT=.TRUE.                                                       SOL01240
      HEAD =.TRUE.                                                       SOL01250
      DO 300 IELNO=1,NELMT                                               SOL01260
  300   CALL ELELIB                                                      SOL01270
     &   (IOPT,LSTTYP,PRINT,IREL,HEAD,NAME,EESE,EPSE,DAMAGE,DUCFAG,      SOL01280
     &    IELNO ,IELDOP,KGDOP,PGEOM,RINPUT,AXIAL,IZDISP,IZLOAD,MSTOR)    SOL01290
C                                                                        SOL01300
C-----------------------------------------                               SOL01310
C- ADD LAST LOAD CASE TO TOTAL MEMBER LOADS ---                          SOL01320
C-----------------------------------------                               SOL01330
C THIS SETS THE TOTAL LOAD IN THE ELEMENT SUBROUTINE EQUAL TO THE        SOL01340
C SUM OF THE TOTAL LOAD AND THE LAST LOAD CASE                           SOL01350
C TWO MAJOR BENIFITS ARE DERIVED FROM THIS:                              SOL01360
C 1) TOTAL ELEMENT LOADS ARE USED AS INITIAL VALUES FOR BUCKLING AND     SOL01370
C    DYNAMIC LOADINGS.                                                   SOL01380
C 2) TOTAL ELEMENT LOADS ARE USED IN NONLINEAR ANALYSIS                  SOL01390
C                                                                        SOL01400
      IOPT=5                                                             SOL01410
      LSTTYP=0                                                           SOL01420
      PRINT=.FALSE.                                                      SOL01430
      HEAD =.FALSE.                                                      SOL01440
      DO 310 IELNO=1,NELMT                                               SOL01450
  310   CALL ELELIB                                                      SOL01460
     &   (IOPT,LSTTYP,PRINT,IREL,HEAD,NAME,EESE,EPSE,DAMAGE,DUCFAG,      SOL01470
     &    IELNO,IELDOP,KGDOP,PGEOM,RINPUT,AXIAL,IZDISP,IZLOAD,MSTOR)     SOL01480
C                                                                        SOL01490
C---------------------------------                                       SOL01500
C- OBTAIN TOTAL STRAIN ENERGY   ---                                      SOL01510
C---------------------------------                                       SOL01520
      ESE=0                                                              SOL01530
      DO 1312 IELNO=1,NELMT                                              SOL01540
        CALL ELELIB                                                      SOL01550
     &  (10,LSTTYP,.FALSE.,IREL,.FALSE.,NAME,EESE,EPSE,DAMAGE,DUCFAG,    SOL01560
     &   IELNO ,IELDOP,KGDOP,PGEOM,RINPUT,AXIAL,IZDISP,IZLOAD,MSTOR)     SOL01570
 1312    ESE=ESE+EESE                                                    SOL01580
         ESE=ESE/4                                                       SOL01590
      WRITE(108,1313) ESE                                                SOL01600
 1313 FORMAT(1X,/,1X,'THE STATICS STRAIN ENERGY OF SYSTEM =',G12.4/)     SOL01610
      WRITE(108,*) '----------------------------------------'            SOL01620
C                                                                        SOL01630
      IF (NLOAD.EQ.1) RETURN                                             SOL01640
C                                                                        SOL01650
C---------------------------------                                       SOL01660
      WRITE(108,1000) TITLE(1),TITLE(2)                                  SOL01670
 1000 FORMAT('1 STRUCTURE.....: ',A,/'  SOLUTION.....: ',A,//           SOL01680
     &       1X,' MAXIMUN VALUES FOR ALL LOAD CASES '/                   SOL01690
     &       1X,'-----------------------------------')                   SOL01700
C                                                                        SOL01710
C---------------------------------                                       SOL01720
C- GET AND PRINT MAXIMUM DISPL ---                                       SOL01730
C---------------------------------                                       SOL01740
C                                                                        SOL01750
      DO 1010 I=1,L4                                                     SOL01760
        WORK(I   )=DISP(I,1)                                             SOL01770
        DO 1010 J=2,NLOAD                                                SOL01780
          IF (ABS(DISP(I,J)).GT.ABS(WORK(I))) WORK(I)=DISP(I,J)          SOL01790
 1010 CONTINUE                                                           SOL01800
      CALL JOIDSP(2,MAXNOD,L4,NNODE,1,L2,ID,IDOF,CONST,WORK,             SOL01810
     &            TITLE,'MAXIMUM DISPLACEMENTS',STEPID,.FALSE.,          SOL01820
     &            NCOS,COSINE,JTCOS)                                     SOL01830
C                                                                        SOL01840
C-------------------------------------                                   SOL01850
C- GET AND PRINT MAXIMUM REACTIONS ---                                   SOL01860
C-------------------------------------                                   SOL01870
C                                                                        SOL01880
      WRITE(108,1000) TITLE(1),TITLE(2)                                  SOL01890
      DO 1020 I=1,L4                                                     SOL01900
        WORK(I   )=LOAD(I,1)                                             SOL01910
        DO 1020 J=2,NLOAD                                                SOL01920
          IF (ABS(LOAD(I,J)) GT.ABS(WORK(I))) WORK(I)=LOAD(I,J)          SOL01930
 1020 CONTINUE                                                           SOL01940
      DO 1030 I=1,6                                                      SOL01950
 1030  SUMRCT(I)=0.                                                      SOL01960
      CALL REACTN(4, TRUE.,NNODE,L1,L2,L3,L4,WORK,DISP,                  SOL01970
     &            MD,STIF,IMD,L4,1    ,1.0,                            SOL01980
     &            MAXNOD,IDOP,COORD,ID,TITLE,STEPID,.FALSE.,             SOL01990
     &            NCOS,COSINE,JTCOS,JTPLG,SUMRCT)                        SOL02000
C                                                                        SOL02010
C---------------------------------                                       SOL02020
C- PRINT MAXIMUN ELEMENT FORCES ---                                      SOL02030
C---------------------------------                                       SOL02040
C                                                                        SOL02050
      WRITE(108,1000) TITLE(1),TITLE(2)                                  SOL02060
      LSTTYP=0                                                           SOL02070
      PRINT=.TRUE.                                                       SOL02080
      HEAD =.FALSE.                                                      SOL02090
      DO 1310 IELNO=1,NELMT                                              SOL02100
 1310   CALL ELELIB                                                      SOL02110
     &   (12,LSTTYP,PRINT,IREL,HEAD,NAME,EESE,EPSE,DAMAGE,DUCFAG,       SOL02120
     &    IELNO,IELDOP,KGDOP,PGEOM,RINPUT,AXIAL,IZDISP,IZLOAD,MSTOR)     SOL02130
C                                                                        SOL02140
  220   FORMAT(5X,I5,1P,6G15.6)                                          SOL02150
  300   FORMAT('1 GLOBAL ',A,',  LOADING #',I5//4X,'NODE',7X,'DX',      SOL02160
     &         13X,'DY',13X,'DZ',13X,'RX',13X,'RY',13X,'RZ'/)            SOL02170
C                                                                        SOL02180
      RETURN                                                             SOL02190
      END                                                                SOL02200
C                                                                        JOIL0010
C*************************************************************************JOIL0020
C                                                                        JOIL0030
      SUBROUTINE JOILOA(BUG,MAXNOD,NDOF,NNODE,NLOAD,NF,ID,IDOF,CONST,    JOIL0040
     &                  FORCE,DSPFLG,JTPLG,TITLE,HEAD)                   JOIL0050
      CHARACTER*(*) TITLE(2)                                             JOIL0060
      CHARACTER  DIR*4,SUPT*25,DOF*18                                    JOIL0070
      CHARACTER*1  CX(6)                                                 JOIL0080
      CHARACTER*14 F(6)                                                  JOIL0090
      LOGICAL DSPFLG,FIRST,BUG,HEAD,BTEST                                JOIL0100
      DIMENSION IDOF(MAXNOD,6),CONST(MAXNOD,6),FORCE(NDOF,NLOAD)         JOIL0110
      DIMENSION ID(MAXNOD),JTPLG(MAXNOD)                                 JOIL0120
C                                                                        JOIL0130
      FIRST = TRUE.                                                      JOIL0140
      DSPFLG=.FALSE.                                                     JOIL0150
    5 FORMAT('1 STRUCTURE .... ',A,/'  SOLUTION. .... ',A,/)            JOIL0160
    6 FORMAT('    APPLIED JOINT LOADS',/'   ---------------------',/)   JOIL0170
C                                                                        JOIL0180
C----------------------                                                  JOIL0190
C- INPUT LOADING DATA --                                                 JOIL0200
C----------------------                                                  JOIL0210
   10 READ(105,*) LOAD,JOINT,JTGEN,JTINC,DIR,VALUE                       JOIL0220
      IF (INDEX(DIR,'END').NE.0) GO TO 100                               JOIL0230
   15 DOF=' '                                                            JOIL0240
      IF (FIRST) THEN                                                    JOIL0250
          IF (HEAD) WRITE(108,5) TITLE(1),TITLE(2)                       JOIL0260
          WRITE(108,6)                                                   JOIL0270
          FIRST= FALSE.                                                  JOIL0280
      ENDIF                                                              JOIL0290
C                                                                        JOIL0300
C------- CHECK FOR A VALID JOINT NUMBER...                               JOIL0310
      IF (JOINT.LT ID(1)  OR  JOINT.GT.ID(NNODE) ) THEN                  JOIL0320
         WRITE(108,21) LOAD                                              JOIL0330
         GO TO 10                                                        JOIL0340
      ENDIF                                                              JOIL0350
C                                                                        JOIL0360
       J=IQUICK(JOINT,ID,NNODE)                                          JOIL0370
       SUPT=' '                                                          JOIL0380
C                                                                        JOIL0390
C-------------------------------------------------------------------------JOIL0400
C------- PUT A LOAD ON A JOINT                                           JOIL0410
C NOTES:  - IX, IY, IZ, MX, MY AND Z ARE THE DOF THAT THE LOADS ARE      JOIL0420
C           APPLIED TO                                                   JOIL0430
C         - THE MOMENTS RESULTING FROM FX, FY AND FZ ARE DUE TO          JOIL0440
C           CONSTRAINTS.  IF A NODE IS NOT CONSTRAINED, THEN THE VALUE   JOIL0450
C           OF CONST IS ZERO.                                            JOIL0460
C         - LOADS APPLIED TO RESTRAINED DOF ARE INTERPERATED AS          JOIL0470
C           SUPPORT DISPLACEMENTS  IE: SUPPORT SETTLEMENT OR             JOIL0480
C           DISPLACEMENT CONTROLL.....                                   JOIL0490
C-- ------------------------------------------------------------ -------JOIL0500
      IF (INDEX(DIR,'FX').NE.0) THEN                                     JOIL0510
         IX=IDOF(J,1)                                                    JOIL0520
         FORCE(IX,LOAD)=FORCE(IX,LOAD)+VALUE                             JOIL0530
         IF ( BTEST(JTPLG(J),12) ) THEN                                  JOIL0540
           MY=IDOF(J,5)                                                  JOIL0550
           FORCE(MY,LOAD)=FORCE(MY,LOAD)+VALUE*CONST(J,3)                JOIL0560
           MZ=IDOF(J,6)                                                  JOIL0570
           FORCE(MZ,LOAD)=FORCE(MZ,LOAD)+VALUE*CONST(J,5)                JOIL0580
           WRITE(DOF,35) IX,MY,MZ                                        JOIL0590
         ELSE                                                            JOIL0600
           WRITE(DOF,35) IX                                              JOIL0610
         ENDIF                                                           JOIL0620
         IF (IX.GT.NF) THEN                                              JOIL0630
           DSPFLG=.TRUE.                                                 JOIL0640
           SUPT=' . JOINT DISPLACEMENT...'                               JOIL0650
         ENDIF                                                           JOIL0660
      ELSE IF (INDEX(DIR,'FY').NE.0) THEN                                JOIL0670
         IY=IDOF(J,2)                                                    JOIL0680
         FORCE(IY,LOAD)=FORCE(IY,LOAD)+VALUE                             JOIL0690
         IF (BTEST(JTPLG(J),13)) THEN                                    JOIL0700
           MX=IDOF(J,4)                                                  JOIL0710
           FORCE(MX,LOAD)=FORCE(MX,LOAD)+VALUE*CONST(J,1)                JOIL0720
           MZ=IDOF(J,6)                                                  JOIL0730
           FORCE(MZ,LOAD)=FORCE(MZ,LOAD)+VALUE*CONST(J,6)                JOIL0740
           WRITE(DOF,35) IY,MX,MZ                                        JOIL0750
         ELSE                                                            JOIL0760
           WRITE(DOF,35) IY                                              JOIL0770
         ENDIF                                                           JOIL0780
         IF (IY GT NF) THEN                                              JOIL0790
           DSPFLG=.TRUE.                                                 JOIL0800
           SUPT='...JOINT DISPLACEMENT...'                               JOIL0810
         ENDIF                                                           JOIL0820
      ELSE IF (INDEX(DIR,'FZ').NE.0) THEN                                JOIL0830
         JZ=IDOF(J,3)                                                    JOIL0840
         FORCE(JZ,LOAD)=FORCE(JZ,LOAD)+VALUE                             JOIL0850
         IF (BTEST(JTPLG(J),14)) THEN                                    JOIL0860
           MX=IDOF(J,4)                                                  JOIL0870
           FORCE(MX,LOAD)=FORCE(MX,LOAD)+VALUE*CONST(J,2)                JOIL0880
           MY=IDOF(J,5)                                                  JOIL0890
           FORCE(MY,LOAD)=FORCE(MY,LOAD)+VALUE*CONST(J,4)                JOIL0900
           WRITE(DOF,35) JZ,MX,MY                                        JOIL0910
         ELSE                                                            JOIL0920
           WRITE(DOF,35) JZ                                              JOIL0930
         ENDIF                                                           JOIL0940
         IF (JZ.GT.NF) THEN                                              JOIL0950
           DSPFLG=.TRUE.                                                 JOIL0960
           SUPT='. JOINT DISPLACEMENT ..'                                JOIL0970
         ENDIF                                                           JOIL0980
      ELSE IF (INDEX(DIR,'MX').NE.0) THEN                                JOIL0990
         MX=IDOF(J,4)                                                    JOIL1000
         FORCE(MX,LOAD)=FORCE(MX,LOAD)+VALUE                             JOIL1010
         WRITE(DOF,35) MX                                                JOIL1020
         IF (MX.GT.NF) THEN                                              JOIL1030
           DSPFLG=.TRUE.                                                 JOIL1040
           SUPT='...JOINT DISPLACEMENT...'                               JOIL1050
         ENDIF                                                           JOIL1060
      ELSE IF (INDEX(DIR,'MY').NE.0) THEN                                JOIL1070
         MY=IDOF(J,5)                                                    JOIL1080
         FORCE(MY,LOAD)=FORCE(MY,LOAD)+VALUE                             JOIL1090
         WRITE(DOF,35) MY                                                JOIL1100
         IF (MY.GT.NF) THEN                                              JOIL1110
           DSPFLG=.TRUE.                                                 JOIL1120
           SUPT='...JOINT DISPLACEMENT.  '                               JOIL1130
         ENDIF                                                           JOIL1140
      ELSE IF (INDEX(DIR,'MZ').NE.0) THEN                                JOIL1150
         MZ=IDOF(J,6)                                                    JOIL1160
         FORCE(MZ,LOAD)=FORCE(MZ,LOAD)+VALUE                             JOIL1170
         WRITE(DOF,35) MZ                                                JOIL1180
         IF (MZ.GT.NF) THEN                                              JOIL1190
           DSPFLG=.TRUE.                                                 JOIL1200
           SUPT='...JOINT DISPLACEMENT...'                               JOIL1210
         ENDIF                                                           JOIL1220
      ELSE                                                               JOIL1230
         WRITE(108,20) JOINT,DIR,VALUE                                   JOIL1240
      ENDIF                                                              JOIL1250
      WRITE(108,30) LOAD,JOINT,DIR,DOF,VALUE,SUPT                        JOIL1260
      IF (JTGEN.GT.0) THEN                                               JOIL1270
         JTGEN=JTGEN-1                                                   JOIL1280
         JOINT=JOINT+JTINC                                               JOIL1290
         GO TO 15                                                        JOIL1300
      ENDIF                                                              JOIL1310
```

```
      GO TO 10                                                          JOIL1320
C                                                                       JOIL1330
   20 FORMAT (' INVALID JOINT LOAD... ',I6,2X,A,2X,1P,G14 6,            JOIL1340
     &        ' --- LOAD IS SKIPPED')                                   JOIL1350
   21 FORMAT (' INVALID LOAD CASE ... ',I6,' --- LOAD IS SKIPPED')      JOIL1360
   30 FORMAT (' LOAD CASE:',I3,'  JOINT:',I6,'  DIRECTION  ',A,         JOIL1370
     &        ' DOF(S)',A,'  MAGNITUDE:',1P,G14.6,2X,A)                JOIL1380
   35 FORMAT (I5,',',',',I5,',',',',I5)                                 JOIL1390
C                                                                       JOIL1400
C----- - RETURN IF NO JOINT LOADS WERE INPUT                            JOIL1410
  100 IF (FIRST) RETURN                                                 JOIL1420
C                                                                       JOIL1430
      IF (.NOT BUG) RETURN                                              JOIL1440
C------- PRINT JOINT LOADS ON EACH DOF                                  JOIL1450
      N5=NDOF*5                                                         JOIL1460
      DO 120 L=1,NLOAD                                                  JOIL1470
        IF (HEAD) WRITE(108,5  ) TITLE(1),TITLE(2)                      JOIL1480
        WRITE(108,130) L                                                JOIL1490
        DO 120 I=1,N5,5                                                 JOIL1500
        I5=MIN((I-4),NDOF)                                              JOIL1510
        DO 110 J=I,I5                                                   JOIL1520
          K=J-I+1                                                       JOIL1530
          CX(K)=' '                                                     JOIL1540
 110      IF (J.GT.NF) CX(K)='*'                                        JOIL1550
          WRITE(108,140) (J,FORCE(J,L),CX(J-I+1),J=I,I5)                JOIL1560
  120 CONTINUE                                                          JOIL1570
C                                                                       JOIL1580
  130 FORMAT (' APPLIED JOINT LOADS,   LOADING #',I5,10X,               JOIL1590
     &        ' NOTE: * DENOTES SUPPORT DISPLACEMENT'/                  JOIL1600
     &        ' ---------------------- '//                             JOIL1610
     &       5X,5(' DOF',7X,'LOAD    ')/)                             JOIL1620
  140   FORMAT (5X,5(0P,I5,1P,G14.6,A) )                                JOIL1630
C                                                                       JOIL1640
      DO 220 L=1,NLOAD                                                  JOIL1650
        IF (HEAD) WRITE(108,5  ) TITLE(1),TITLE(2)                      JOIL1660
        WRITE(108,200) L                                                JOIL1670
        DO 220 I=1,NNODE                                                JOIL1680
        DO 210 J=1,6                                                    JOIL1690
          CX(J)=' '                                                     JOIL1700
          IF (BTEST(JTFLG(I),J-1)) THEN                                 JOIL1710
            F(J)=' '                                                    JOIL1720
          ELSE                                                          JOIL1730
            K=IDOF(I,J)                                                 JOIL1740
            WRITE(F(J),240) FORCE(K,L)                                  JOIL1750
            IF (K.GT.NF) CX(J)='*'                                      JOIL1760
          ENDIF                                                         JOIL1770
 210      CONTINUE                                                      JOIL1780
          WRITE(108,230) ID(I),(F(J),CX(J),J=1,6)                       JOIL1790
  220 CONTINUE                                                          JOIL1800
C                                                                       JOIL1810
  200 FORMAT (' APPLIED JOINT LOADS,   LOADING #',I5,10X,               JOIL1820
     &        ' NOTE. * DENOTES SUPPORT DISPLACEMENT'/                  JOIL1830
     &        ' ---------------------- '//6X,'NODE',7X,'FX',            JOIL1840
     &       13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ'/)             JOIL1850
  230   FORMAT (5X,I5,12A )                                             JOIL1860
  240   FORMAT (1P,G14.6)                                               JOIL1870
C                                                                       JOIL1880
      RETURN                                                            JOIL1890
      END                                                               JOIL1900
C                                                                       JOID0010
C-----------------------------------------------------------------------JOID0020
C                                                                       JOID0030
      SUBROUTINE JOIDSP(IOPT,MAXNOD,NDOF,NNODE,NLOAD,NF,ID,IDOF,        JOID0040
     &          CONST,DISP,TITLE,NAME,STEPID,HEAD,NCOS,COSINE,JTCOS)    JOID0050
      CHARACTER*(*) TITLE(2),STEPID,NAME                                JOID0060
      CHARACTER*15 CD(6),EQUAL*60                                       JOID0070
      LOGICAL HEAD                                                      JOID0080
      DIMENSION IDOF(MAXNOD,6),CONST(MAXNOD,6)                          JOID0090
      DIMENSION DISP(NDOF,NLOAD),ID(MAXNOD)                             JOID0100
      DIMENSION D(6),G(6),COSINE(3,3,NCOS),JTCOS(MAXNOD)                JOID0110
      DATA EQUAL                                                        JOID0120
     &  /'-----------------------------------------------------------'/ JOID0130
C                                                                       JOID0140
      NLEN=MIN(LEN(NAME)+1,60)                                          JOID0150
      NC=1                                                              JOID0160
C                                                                       JOID0170
C-----------------------------                                          JOID0180
C- LOOP FOR EACH LOAD CASE  ---                                         JOID0190
C-----------------------------                                          JOID0200
C                                                                       JOID0210
      DO 300 L=1,NLOAD                                                  JOID0220
        NDSPLS=0                                                        JOID0230
        IF (.NOT.(NCOS.EQ.1 .AND. COSINE(1,1,1).EQ.1 .AND.              JOID0240
     &       COSINE(2,2,1).EQ.1   AND. COSINE(3,3,1).EQ.1 )             JOID0250
     &       .AND. IOPT.EQ.2 ) GO TO 210                                JOID0260
C                                                                       JOID0270
C-----------------------------                                          JOID0280
C- PRINT GLOBAL DISPLACEMENTS ---                                       JOID0290
C-----------------------------                                          JOID0300
C  NOTE:DISP MAY CONTAIN DISPLACEMENTS, VELOCITIES, ACCELERATIONS OR    JOID0310
C       EIGENVALUES...                                                  JOID0320
C       ALL OF WHICH ARE PRINTED OUT AT EACH NODE....                   JOID0330
C                                                                       JOID0340
        IF (HEAD) WRITE(108,310) TITLE(1),TITLE(2)                      JOID0350
        IF (IOPT.NE.2) WRITE(108,315) NAME,L,STEPID,EQUAL(1:NLEN)       JOID0360
        IF (IOPT EQ.2) WRITE(108,415) NAME,          EQUAL(1:NLEN)      JOID0370
        DO 100 I=1,NNODE                                                JOID0380
C                                                                       JOID0390
C..... . ..  TRANSFER DISPL TO LOCAL VECTOR .... ... ...                JOID0400
        DO 20 J=1,6                                                     JOID0410
          IJ=IDOF(I,J)                                                  JOID0420
   20     D(J)=DISP(IJ,L)                                               JOID0430
C                                                                       JOID0440
C.. ........  ADD ROTATIONS TO DISPL FOR CONSTRAINTS                    JOID0450
C..........   IF UNCONSTRAINED, CONST(I,J)=0                           JOID0460
        D(1) = D(1) + D(5)*CONST(I,3) + D(6)*CONST(I,5)                 JOID0470
        D(2) = D(2) + D(4)*CONST(I,1) + D(6)*CONST(I,6)                 JOID0480
        D(3) = D(3) + D(4)*CONST(I,2) + D(5)*CONST(I,4)                 JOID0490
C                                                                       JOID0500
C. ..........  TRANSFER TO GLOBAL DISPLACEMENTS .....                   JOID0510
        JCOS=JTCOS(I)                                                   JOID0520
        DO 25 I1=1,3                                                    JOID0530
          I2=I1+3                                                       JOID0540
          G(I1)=0                                                       JOID0550
          G(I2)=0                                                       JOID0560
          DO 25 J1=1,3                                                  JOID0570
            J2=J1+3                                                     JOID0580
            G(I1)=G(I1)+COSINE(J1,I1,JCOS)*D(J1)                        JOID0590
   25       G(I2)=G(I2)+COSINE(J1,I1,JCOS)*D(J2)                        JOID0600
C                                                                       JOID0610
C. ..... .   WRITE DISPL                                                JOID0620
        DO 30 J=1,6                                                     JOID0630
   30   WRITE(CD(J),9) G(J)                                             JOID0640
    9   FORMAT(1P,G14.6,' ')                                            JOID0650
        WRITE(108,220) ID(I),(CD(K),K=1,6)                              JOID0660
C                                                                       JOID0670
        NDSPLS=NDSPLS+1                                                 JOID0680
  100 CONTINUE                                                          JOID0690
C                                                                       JOID0700
      IF (NCOS EQ.1 .AND. COSINE(1,1,1).EQ.1  AND. COSINE(2,2,1) EQ 1   JOID0710
     &              AND. COSINE(3,3,1) EQ 1 ) GO TO 300                 JOID0720
C                                                                       JOID0730
C-----------------------------                                          JOID0740
C- PRINT JOINT DISPLACEMENTS ---                                        JOID0750
C-----------------------------                                          JOID0760
C  NOTE:DISP MAY CONTAIN DISPLACEMENTS, VELOCITIES, ACCELERATIONS OR    JOID0770
C       EIGENVALUES...                                                  JOID0780
C       ALL OF WHICH ARE PRINTED OUT AT EACH NODE....                   JOID0790
C                                                                       JOID0800
  210   IF (HEAD.AND .NDSPLS.GT 20) WRITE(108,310) TITLE(1),TITLE(2)    JOID0810
        IF (IOPT.NE 2) WRITE(108,316) NAME,L,STEPID,EQUAL(1:NLEN)       JOID0820
        IF (IOPT EQ.2) WRITE(108,416) NAME,          EQUAL(1 NLEN)      JOID0830
        DO 200 I=1,NNODE                                                JOID0840
C                                                                       JOID0850
C.....  .... TRANSFER GLOBAL DISPL TO LOCAL VECTOR ..... .... .         JOID0860
        DO 219 J=1,6                                                    JOID0870
          IJ=IDOF(I,J)                                                  JOID0880
  219     D(J)=DISP(IJ,L)                                               JOID0890
C                                                                       JOID0900
C  . ......... ADD ROTATIONS TO DISPL FOR CONSTRAINTS                   JOID0910
C.. ........... IF UNCONSTRAINED, CONST(I,J)=0                         JOID0920
        D(1) = D(1) + D(5)*CONST(I,3) + D(6)*CONST(I,5)                 JOID0930
        D(2) = D(2) + D(4)*CONST(I,1) + D(6)*CONST(I,6)                 JOID0940
        D(3) = D(3) + D(4)*CONST(I,2) + D(5)*CONST(I,4)                 JOID0950
C                                                                       JOID0960
C.. ........... WRITE DISPL                                            JOID0970
        DO 230 J=1,6                                                    JOID0980
          WRITE(CD(J),9) D(J)                                           JOID0990
          IJ=IDOF(I,J)                                                  JOID1000
  230   WRITE(108,221) ID(I),(CD(K),K=1,6),JTCOS(I)                     JOID1010
C                                                                       JOID1020
  200 CONTINUE                                                          JOID1030
      WRITE(108,320)                                                    JOID1040
  300 CONTINUE                                                          JOID1050
C                                                                       JOID1060
   10 FORMAT (1P,G15.7)                                                 JOID1070
  220 FORMAT (5X,I5,6A)                                                 JOID1080
  221 FORMAT (5X,I5,6A,I10)                                             JOID1090
  310 FORMAT ('1 STRUCTURE... : ',A,/'  SOLUTION...  ... ',A)           JOID1100
  315 FORMAT (//2X, 'GCS ',A,',  LOADING #',I5,5X,A, /                 JOID1110
     &        1X,'-----',A//5X,'NODE',7X,'DX',13X,'DY',13X,'DZ',       JOID1120
     &        13X,'RX',13X,'RY',13X,'RZ'/)                             JOID1130
  316 FORMAT (//2X, 'GCS ',A,',  LOADING #',I5,5X,A, /1X,'-----',A      JOID1140
     &        //6X,'NODE',7X,'DX',13X,'DY',13X,'DZ',13X,'RX',          JOID1150
     &        13X,'RY',13X,'RZ',6X,'  COSINE #'/)                      JOID1160
  415 FORMAT (//2X, 'GCS ',A/1X,'-----',A/25X,                          JOID1170
     &     'NOTE: MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'//6X,     JOID1180
     &   'NODE',7X,'DX',13X,'DY',13X,'DZ',13X,'RX',13X,'RY',13X,'RZ'/)  JOID1190
  416 FORMAT (//2X, 'GCS ',A/1X,'-----',A/25X,                          JOID1200
     &     'NOTE: MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'//6X,     JOID1210
     &   'NODE',7X,'DX',13X,'DY',13X,'DZ',13X,'RX',13X,'RY',13X,'RZ',   JOID1220
     &   6X,'  COSINE #'/)                                              JOID1230
  320 FORMAT (4X,'NOTE. * DENOTES A RESTRAINED DEGREE OF FREEDOM')      JOID1240
C                                                                       JOID1250
      RETURN                                                            JOID1260
      END                                                               JOID1270
C                                                                       JOID1280
C                                                                       ELEL0010
C-----------------------------------------------------------------------ELEL0020
C                                                                       ELEL0030
      SUBROUTINE ELELOA(BUG,NF,NELMT,MAXELD,NELD,NDOF,NLOAD,FORCE,      ELEL0040
     &                  IELD,ELD,IDISP,ILOAD ,NAME,TITLE,HEAD)          ELEL0050
C                                                                       ELEL0060
C NOTE ON ARGUMENTS: THE LOAD MATRIX FORCE IS PASSED FROM THE CALLING   ELEL0070
C  ROUTINE.  FORCE IS EQUIVALENCED TO THE Z MATRIX BY THE CALLING ROUTINEELEL0080
C  IE  Z(ILOAD)=FORCE(1,1), WHERE ILOAD IS THE OFFSET IN THE Z MATRIX.   ELEL0090
C  BOTH THE CALLING ROUTINE AND SUBROUTINE ELE_LIB USE THE Z MATRIX      ELEL0100
C                                                                       ELEL0110
      LOGICAL FIRST,HEAD,FALSE,AXIAL,BUG                                ELEL0120
      CHARACTER*(*) TITLE(2),NAME                                       ELEL0130
      CHARACTER*20 TYPE,DIRI,DIR*40                                     ELEL0140
      CHARACTER*1 CK(6)                                                 ELEL0150
      DIMENSION FORCE(NDOF,NLOAD),IELD(5,MAXELD),ELD(12,MAXELD)         ELEL0160
      DIMENSION RINPUT(100)                                             ELEL0170
C                                                                       ELEL0180
      FIRST=.TRUE.                                                      ELEL0190
    5 FORMAT ('1 STRUCTURE..... ',A,/'  SOLUTION..... ',A,/)            ELEL0200
    6 FORMAT (/'  ELEMENT LOADS  ',/'  ---------------- '//            ELEL0210
     & 13X,' LOAD ELEM  GEN  INC  GROUP & DIRECTION    VALUES.... '/)  ELEL0220
C---------------------                                                  ELEL0230
C- INPUT LOADING DATA --                                                ELEL0240
C---------------------                                                  ELEL0250
      N=0                                                               ELEL0260
   10 N=N+1                                                             ELEL0270
      IF (N.GT.MAXELD) GO TO 100                                        ELEL0280
   20 READ(105,*) (IELD(I,N),I=1,4),TYPE,DIRI,ELD(1,N)                 ELEL0290
      DIR=TYPE//DIRI                                                    ELEL0300
      IF (INDEX(DIR,'END').NE.0) GO TO 100                              ELEL0310
      NELD=N                                                            ELEL0320
      IF (FIRST)  THEN                                                  ELEL0330
          IF (HEAD) WRITE(108,5) TITLE(1),TITLE(2)                      ELEL0340
          WRITE(108,6)                                                  ELEL0350
          FIRST=.FALSE                                                  ELEL0360
      ENDIF                                                             ELEL0370
C                                                                       ELEL0380
C------- PROCESS LOAD GROUPS AND READ ADDITIONAL INFO...                ELEL0390
      IF (INDEX(DIR,'CONC').NE.0) THEN                                  ELEL0400
         KODE=100                                                       ELEL0410
         J=2                                                            ELEL0420
         BACKSPACE(105)                                                 ELEL0430
         READ(105,*) (IELD(I,N),I=1,4),TYPE,DIRI,(ELD(I,N),I=1,J)       ELEL0440
         DIR=TYPE//DIRI                                                 ELEL0450
         CALL CKRNG(ELD(2,N),0.,1.,'DISTANCE')                          ELEL0460
      ELSE IF (INDEX(DIR,'UNIF').NE.0) THEN                             ELEL0470
         J=1                                                            ELEL0480
         KODE=200                                                       ELEL0490
      ELSE IF (INDEX(DIR,'VARR') NE.0) THEN                             ELEL0500
         KODE=300                                                       ELEL0510
         J=4                                                            ELEL0520
         BACKSPACE(105)                                                 ELEL0530
         READ(105,*) (IELD(I,N),I=1,4),TYPE,DIRI,(ELD(I,N),I=1,J)       ELEL0540
         DIR=TYPE//DIRI                                                 ELEL0550
         CALL CKRNG(ELD(3,N),0.,ELD(4,N),'DISTANCE')                    ELEL0560
         CALL CKRNG(ELD(4,N),ELD(3,N),1.,'DISTANCE')                    ELEL0570
      ELSE IF (INDEX(DIR,'PEM').NE.0) THEN                              ELEL0580
         KODE=400                                                       ELEL0590
         J=12                                                           ELEL0600
         BACKSPACE(105)                                                 ELEL0610
         READ(105,*) (IELD(I,N),I=1,4),TYPE,DIRI,(ELD(I,N),I=1,J)       ELEL0620
         DIR=TYPE//DIRI                                                 ELEL0630
      ELSE                                                              ELEL0640
         WRITE(108,30) 'INVALID LOAD TYPE',                             ELEL0650
     &                 (IELD(I,N),I=1,4),DIR,ELD(1,N)                   ELEL0660
         GO TO 20                                                       ELEL0670
      ENDIF                                                             ELEL0680
      IF (INDEX(DIR,'FX').NE.0) THEN                                    ELEL0690
         KODE=KODE+1                                                    ELEL0700
      ELSE IF (INDEX(DIR,'FY').NE.0) THEN                               ELEL0710
         KODE=KODE+2                                                    ELEL0720
      ELSE IF (INDEX(DIR,'FZ').NE.0) THEN                               ELEL0730
         KODE=KODE+3                                                    ELEL0740
      ELSE IF (INDEX(DIR,'MX').NE.0) THEN                               ELEL0750
         KODE=KODE+4                                                    ELEL0760
      ELSE IF (INDEX(DIR,'MY').NE.0) THEN                               ELEL0770
         KODE=KODE+5                                                    ELEL0780
      ELSE IF (INDEX(DIR,'MZ').NE.0) THEN                               ELEL0790
         KODE=KODE+6                                                    ELEL0800
      ELSE IF (KODE.NE.400) THEN                                        ELEL0810
         WRITE(108,30) 'INVALID LOAD DIRECTION',                        ELEL0820
     &                 (IELD(I,N),I=1,4),DIR,(ELD(1,N),I=1,J)         ELEL0830
         GO TO 20                                                       ELEL0840
      ENDIF                                                             ELEL0850
      IELD(5,N)=KODE                                                    ELEL0860
      WRITE(108,40) (IELD(I,N),I=1,4),DIR,(ELD(I,N),I=1,J)              ELEL0870
      GO TO 10                                                          ELEL0880
C                                                                       ELEL0890
   30 FORMAT (' ',A, ' - - LOAD IS SKIPPED'/                            ELEL0900
     &        1X,4I5,1X,A20,1P,6G14.6, /(42X,6G14 6) )                  ELEL0910
   40 FORMAT (1X,4I5,1X,A20,1P,6G14.6, /(42X,6G14.6) )                  ELEL0920
C                                                                       ELEL0930
C------- RETURN IF NO ELEMENT LOADS WERE INPUT                          ELEL0940
  100 IF (FIRST)  RETURN                                                ELEL0950
C                                                                       ELEL0960
C------- LOOP TO GENERATE MEMBER LOADS FOR EACH ELEMENT                 ELEL0970
      IOPT=6                                                            ELEL0980
      DO 101 IELNO=1,NELMT                                              ELEL0990
        FALSE=.FALSE.                                                   ELEL1000
  101   CALL ELELIB                                                     ELEL1010
     &  (IOPT,LSTTYP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DAMAGE,DUCFAG,     ELEL1020
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IDISP,ILOAD,MSTOR)      ELEL1030
C                                                                       ELEL1040
C------- PRINT LOADS AT EACH DOF                                        ELEL1050
```

```
      IF (BUG) THEN                                                     ELEL1060
           N5=NDOF*5                                                    ELEL1070
           DO 120 L=1,NLOAD                                             ELEL1080
             IF (HEAD) WRITE(108,5) TITLE(1),TITLE(2)                   ELEL1090
             WRITE(108,130) L                                           ELEL1100
             DO 120 I=1,N5,5                                            ELEL1110
             I5=MIN((I+4),NDOF)                                         ELEL1120
             DO 110 J=I,I5                                              ELEL1130
               K=J-I+1                                                  ELEL1140
               CX(K)=' '                                                ELEL1150
  110        IF (J.GT.NF) CX(K)='*'                                     ELEL1160
             WRITE(108,140) (J,FORCE(J,L),CX(J-I+1),J=I,I5)             ELEL1170
  120      CONTINUE                                                     ELEL1180
      ENDIF                                                             ELEL1190
C                                                                       ELEL1200
  130 FORMAT ('  GLOBAL JOINT FORCES DUE TO ELEMENT LOADS ',3X,         ELEL1210
     &        '  LOADING #',I5,10X,                                     ELEL1220
     &        '  NOTE: * DENOTES SUPPORT DISPLACEMENT'/                 ELEL1230
     &        ' ------------------------------------------'//          ELEL1240
     &        5X,5('  DOF',7X,'LOAD    ')/)                           ELEL1250
  140   FORMAT (5X,5(0P,I5,1P,G14.6,A) )                                ELEL1260
C                                                                       ELEL1270
      RETURN                                                            ELEL1280
      END                                                               ELEL1290
C                                                                       CKRN0010
C-----------------------------------------------------------------------CKRN0020
C                                                                       CKRN0030
      SUBROUTINE CKRNG(V,A,B,NAME)                                      CKRN0040
      CHARACTER*(*) NAME                                                CKRN0050
      IF (A.LT.V.AND.V.LT.B) RETURN                                     CKRN0060
      WRITE(108,10) NAME,A,V,B                                          CKRN0070
   10 FORMAT(' ***',A,' IS OUT OF RANGE, SET ',1P,G13.5,' LE INPUT=',   CKRN0080
     &       G13.5,' LE ',G13.5)                                        CKRN0090
      V=MIN(V,B)                                                        CKRN0100
      V=MAX(V,A)                                                        CKRN0110
      RETURN                                                            CKRN0120
      END                                                               CKRN0130
C-----------------------------------------------------------------------SOL00010
      SUBROUTINE SOL02                                                  SOL00020
      LOGICAL SLMASS,TEST,BTEST,UNBAL                                   SOL00030
      CHARACTER INTEG*20                                                SOL00040
$INCLUDE:'ZCOMN'                                                        SOL00050
C                                                                       SOL00060
C                                                                       SOL00070
C-----------------                                                      SOL00080
C- PRINT HEADER ---                                                     SOL00090
C-----------------                                                      SOL00100
      WRITE(108,10) TITLE(1),TITLE(2)                                   SOL00110
   10 FORMAT ('1 STRUCTURE..... ',A,/'  SOLUTION.. ..: ',A,/)           SOL00120
C                                                                       SOL00130
C----------------------                                                 SOL00140
C- INPUT LOADING DATA --                                                SOL00150
C----------------------                                                 SOL00160
      READ(105,*) INTEG,THETA, ELSTIC,UNBAL                             SOL00170
      READ(105,*) IPRNT,IWRITE,MAXACC,SLMASS                            SOL00180
      READ(105,*) TO,DT,TF,GRAV                                         SOL00190
      IPRNT=MAX(IPRNT,1)                                                SOL00200
C                                                                       SOL00210
      IF (TEST(INTEG,'LINEAR' ,.FALSE.)) THEN                           SOL00220
            INTEGO=1                                                    SOL00230
            WRITE(108,11) 'LINEAR ACCELERATION METHOD',                 SOL00240
     &                    '-------------------------'                   SOL00250
      ELSE IF (TEST(INTEG,'AVERAGE',.FALSE.)) THEN                      SOL00260
            INTEGO=2                                                    SOL00270
            WRITE(108,11) 'AVERAGE ACCELERATION METHOD',                SOL00280
     &                    '-------------------------'                   SOL00290
      ELSE IF (TEST(INTEG,'WILSON' ,.FALSE.)) THEN                      SOL00300
            INTEGO=3                                                    SOL00310
            WRITE(108,11) 'WILSON THETA METHOD-------------------'      SOL00320
            WRITE(108,12) THETA                                         SOL00330
      ELSE IF (TEST(INTEG,'WILSON' ,.FALSE.)) THEN                      SOL00340
            WRITE(108,*) 'INVALID INTEGRATION METHOD'                   SOL00350
            WRITE(108,11) INTEG,'--------------------'                  SOL00360
            RETURN                                                      SOL00370
      ENDIF                                                             SOL00380
C                                                                       SOL00390
C------- CHECK FOR INVALID INTEGRATION WITH PROPORTIONAL DAMPING         SOL00400
      IF ((INTEGO.EQ.1 .OR. INTEGO.EQ.2) .AND. FDAMP.NE.1) THEN         SOL00410
         WRITE(108,19)                                                  SOL00420
   19    FORMAT (5X,'INTEGRATION METHOD IS ONLY VALID FOR ',            SOL00430
     &           'PROPORTIONAL DAMPING'//,5X,'SOLN IS ABORTED')         SOL00440
         RETURN                                                         SOL00450
      ENDIF                                                             SOL00460
C                                                                       SOL00470
      WRITE(108,13) TO,DT,TF,GRAV,IPRNT                                 SOL00480
      IF (ELSTIC) WRITE(108,14)                                         SOL00490
C                                                                       SOL00500
   11 FORMAT(1X,' SOLUTION #2, ',A,' OF NUMERICAL INTEGRATION'/         SOL00510
     &       1X,'--------------',A,'------------------------'/)         SOL00520
   12 FORMAT(1X,' THETA ..................... ',1P,G15.6)             SOL00530
   13 FORMAT(1X,' INITIAL TIME ..................',1P,G15.6/           SOL00540
     &       1X,' TIME STEP ..................... ',1P,G15.6/         SOL00550
     &       1X,' FINAL TIME ..................... ',1P,G15.6/        SOL00560
     &       1X,' ACCELERATION DUE TO GRAVITY .... ',1P,G15.6/        SOL00570
     &       1X,' STEP INTERVAL FOR PRINTING ......',0P,I5)           SOL00580
   14 FORMAT(1X,' THE STRUCTURE IS ASSUMED TO BEHAVE ELASTICALLY')      SOL00590
C                                                                       SOL00600
C---------------------------------                                      SOL00610
C- SET GEOMETRIC STIFFNESS FLAGS ---                                    SOL00620
C---------------------------------                                      SOL00630
      KGLOAD=NZ(IZKGDT+1)                                               SOL00640
      KGTYPE=NZ(IZKGDT+2)                                               SOL00650
      KGFORM=NZ(IZKGDT+3)                                               SOL00660
      NEWKG=.TRUE.                                                      SOL00670
      IF (KGLOAD.EQ.0 .OR. KGTYPE.EQ.0 .OR. KGFORM.EQ.0) NEWKG=.FALSE.  SOL00680
C                                                                       SOL00690
C      ---- SET UP CONSTANTS ------------   ---                        SOL00700
      L3=NCOND+1                                                        SOL00710
      L4=NCOND +NFREE                                                   SOL00720
C                                                                       SOL00730
C      ----- INITIALIZE STORAGE FOR LOAD, DISPL, VELOC, ACCEL           SOL00740
C     Z(IZLOAD) - LOCATION OF THE TOTAL LOAD MATRIX                     SOL00750
C     Z(IZDISP) - LOCATION OF THE TOTAL DISPLACEMENT MATRIX             SOL00760
C     Z(IZVEL ) - LOCATION OF THE TOTAL VELOCITY MATRIX                 SOL00770
C     Z(IZACC ) - LOCATION OF THE TOTAL ACCELERATION MATRIX             SOL00780
C     Z(IZDLOA) - LOCATION OF THE INCRIMENTAL LOAD MATRIX               SOL00790
C     Z(IZDDSP) - LOCATION OF THE INCRIMENTAL DISPLACEMENT MATRIX       SOL00800
C     Z(IZDVEL) - LOCATION OF THE INCRIMENTAL VELOCITY MATRIX           SOL00810
C     Z(IZDACC) - LOCATION OF THE INCRIMENTAL ACCELERATION MATRIX       SOL00820
C                                                                       SOL00830
      IF (IZLOAD.EQ.0) THEN                                             SOL00840
          IZLOAD = IZ                                                   SOL00850
          IZ     = IZ+NDOF                                              SOL00860
          CALL CKSTOR(0,0)                                              SOL00870
          DO 1 I=1,NDOF                                                 SOL00880
    1     Z(I+IZLOAD-1) =0                                              SOL00890
      ENDIF                                                             SOL00900
C                                                                       SOL00910
      IF (IZDISP.EQ.0) THEN                                             SOL00920
          IZDISP = IZ                                                   SOL00930
          IZ     = IZ+NDOF                                              SOL00940
          CALL CKSTOR(0,0)                                              SOL00950
          DO 2 I=1,NDOF                                                 SOL00960
    2     Z(I-IZDISP-1) =0                                              SOL00970
      ENDIF                                                             SOL00980
C                                                                       SOL00990
      IF (IZVEL .EQ.0) THEN                                             SOL01000
          IZVEL  = IZ                                                   SOL01010
          IZ     = IZ+NDOF                                              SOL01020
          CALL CKSTOR(0,0)                                              SOL01030
          DO 3 I=1,NDOF                                                 SOL01040
    3     Z(I-IZVEL -1) =0                                              SOL01050
      ENDIF                                                             SOL01060
C                                                                       SOL01070
      IF (IZACC  EQ.0) THEN                                             SOL01080
          IZACC  = IZ                                                   SOL01090
          IZ     = IZ+NDOF                                              SOL01100
          CALL CKSTOR(0,0)                                              SOL01110
          DO 4 I=1,NDOF                                                 SOL01120
    4     Z(I-IZACC-1) =0                                               SOL01130
      ENDIF                                                             SOL01140
C                                                                       SOL01150
      IF (IZDLOA EQ.0) THEN                                             SOL01160
          IZDLOA = IZ                                                   SOL01170
          IZ     = IZ+NDOF                                              SOL01180
          CALL CKSTOR(0,0)                                              SOL01190
          DO 5 I=1,NDOF                                                 SOL01200
    5     Z(I-IZDLOA-1) =0.                                             SOL01210
      ENDIF                                                             SOL01220
      IF (IZDDSP.EQ.0) THEN                                             SOL01230
          IZDDSP = IZ                                                   SOL01240
          IZ     = IZ+NDOF                                              SOL01250
          CALL CKSTOR(0,0)                                              SOL01260
          DO 6 I=1,NDOF                                                 SOL01270
    6     Z(I+IZDDSP-1) =0                                              SOL01280
      ENDIF                                                             SOL01290
      IF (IZDVEL EQ.0) THEN                                             SOL01300
          IZDVEL = IZ                                                   SOL01310
          IZ     = IZ+NDOF                                              SOL01320
          CALL CKSTOR(0,0)                                              SOL01330
          DO 7 I=1,NDOF                                                 SOL01340
    7     Z(I+IZDVEL-1) =0.                                             SOL01350
      ENDIF                                                             SOL01360
      IF (IZDACC.EQ.0) THEN                                             SOL01370
          IZDACC = IZ                                                   SOL01380
          IZ     = IZ+NDOF                                              SOL01390
          CALL CKSTOR(0,0)                                              SOL01400
          DO 8 I=1,NDOF                                                 SOL01410
    8     Z(I+IZDACC-1) =0.                                             SOL01420
      ENDIF                                                             SOL01430
      IF (IZPUB  EQ.0) THEN                                             SOL01440
          IZPUB  = IZ                                                   SOL01450
          IZ     = IZ+NDOF                                              SOL01460
          CALL CKSTOR(0,0)                                              SOL01470
          DO 9 I=1,NDOF                                                 SOL01480
    9     Z(I+IZPUB-1) =0.                                              SOL01490
      ENDIF                                                             SOL01500
      IW=IZ+1                                                           SOL01510
C                                                                       SOL01520
C---------- FORM MASS MATRIX -----                                      SOL01530
C                                                                       SOL01540
      IMDMS =NZ(IZMDMS+NDOF)                                            SOL01550
      DO 400 I=1,IMDMS                                                  SOL01560
  400     Z(I-IZMASS-1)=0.                                              SOL01570
      CALL MFORM(3,INMASS,NMASS,Z(IZMASS),NZ(IZMDMS),MD,NDOF,IMDMS,     SOL01580
     &     NNODE,MAXNOD,BTEST(IBUG,5),IBUG,NZ(IZID),NZ(IZIDOF),        SOL01590
     &     Z(IZCNST),NZ(IZJCOS),Z(IZMDAT),Z(IW),NZ(IW+21),Z(IW+27),     SOL01600
     &     Z(IW+ 63),Z(IW+ 99),TITLE)                                   SOL01610
C                                                                       SOL01620
C---------- FORM SPECIAL MASS MATRIX -----                              SOL01630
C                                                                       SOL01640
      IF (SLMASS) THEN                                                  SOL01650
       READ(105,*) INMASS,IFMASS                                        SOL01660
       CALL MASSIN(INMASS,IFMASS,.FALSE.,BUG,IBUG,TITLE,MAXZ,           SOL01670
     &             MAXNOD,NNODE,IZ,IW,IZID,Z,NZ,IZSLMD,IZSLMS,        SOL01680
     &             IZLM,IZKE,IZMDT2,NDOF,NMASS,IZIDOF,IZCNST,         SOL01690
     &             IZJCOS,NCOND,IZMD)                                 SOL01700
       IMDMSL=NZ(IZSLMD+NDOF)                                           SOL01710
      ELSE                                                              SOL01720
          IZSLMS=0                                                      SOL01730
          IZSLMD=0                                                      SOL01740
          IMDMSL=1                                                      SOL01750
      ENDIF                                                             SOL01760
      IMDMS =NZ(IZMDMS+NDOF)                                            SOL01770
C                                                                       SOL01780
C------- SET UP STORAGE FOR MAIN DIAGONAL ADDRESS OF DYNAMIC STIFFNESS   SOL01790
      IZMDS= IZ                                                         SOL01800
      IZ   = IZ + NFREE +1                                              SOL01810
      CALL CKSTOR(0,IW)                                                 SOL01820
C                                                                       SOL01830
C------- DEVELOP SKYLINE OF DYNAMIC STIFFNESS (S)                       SOL01840
      CALL SKYLN2(NDOF,L3,L4,                                           SOL01850
     &            NZ(IZMDS),NZ(IZMD),NZ(IZMDMS),NZ(IZMDKG),KGFORM)      SOL01860
C                                                                       SOL01870
C---- DETERMINE THE LENGTH OF THE DYNAMIC STIFFNESS MATRIX              SOL01880
      IMDS=NZ(IZMDS-NFREE)   NZ(IZMDS) +1                               SOL01890
C                                                                       SOL01900
C---- DETERMINE THE ADDRESS OF THE DYNAMIC STIFFNESS MATRIX             SOL01910
      IZS    = IZ                                                       SOL01920
      IZ     = IZ+NZ(IZMDS+NFREE)                                       SOL01930
C                                                                       SOL01940
C---------- DEFINE STORAGE REQD FOR NUMERICAL INTEGRATION               SOL01950
C           AND INITILIZE VALUES                                        SOL01960
      IZAN   = IZ                                                       SOL01970
      IZ     = IZ+NFREE                                                 SOL01980
      IZBN   = IZ                                                       SOL01990
      IZ     = IZ+NFREE                                                 SOL02000
      IZFBAR = IZ                                                       SOL02010
      IZ     = IZ+NFREE                                                 SOL02020
C                                                                       SOL02030
C---- SAVE STORAGE FOR DYNAMIC GROUND ACCELERATIONS                     SOL02040
          IZAGTX=IZ                                                     SOL02050
          IZ    =IZ+MAXACC                                              SOL02060
          IZAGTY=IZ                                                     SOL02070
          IZ    =IZ+MAXACC                                              SOL02080
          IZAGTZ=IZ                                                     SOL02090
          IZ    =IZ+MAXACC                                              SOL02100
          IZAINP=IZ                                                     SOL02110
          NAINP =MAX(NDOF,MAXACC)                                       SOL02120
          IZ    =IZ+NAINP                                               SOL02130
          IZMCOS=IZ                                                     SOL02140
          IZ    =IZ+NDOF*3                                              SOL02150
C                                                                       SOL02160
C---- SAVE STORAGE FOR ENERGY CALCULATIONS                              SOL02170
          IZZERO=IZ                                                     SOL02180
          IZ    =IZ+NDOF                                                SOL02190
          IZDM  =IZ                                                     SOL02200
          IZ    =IZ+NDOF                                                SOL02210
          IZVM  =IZ                                                     SOL02220
          IZ    =IZ+NDOF                                                SOL02230
          IZDBAR=IZ                                                     SOL02240
          IZ    =IZ+NDOF                                                SOL02250
          IZ    =IZ+NDOF                                                SOL02260
          IZDLO2=IZ                                                     SOL02270
          IZ    =IZ+NDOF                                                SOL02280
          IZTFD =IZ                                                     SOL02290
          IZ    =IZ+NDOF                                                SOL02300
          IZDFD =IZ                                                     SOL02310
          IZ    =IZ+NDOF                                                SOL02320
          IZPG  =IZ                                                     SOL02330
          IZ    =IZ+NDOF                                                SOL02340
          IZDPG =IZ                                                     SOL02350
          IZ    =IZ+NDOF                                                SOL02360
      DO 25 I=0,NDOF-1                                                  SOL02370
        Z(IZPG  +I)=0                                                   SOL02380
        Z(IZDPG +I)=0.                                                  SOL02390
        Z(IZZERO+I)=0                                                   SOL02400
        Z(IZDM  +I)=0                                                   SOL02410
        Z(IZVM  +I)=0.                                                  SOL02420
        Z(IZDBAR+I)=0.                                                  SOL02430
        Z(IZDLO2+I)=0.                                                  SOL02440
        Z(IZTFD +I)=0.                                                  SOL02450
   25   Z(IZDFD +I)=0                                                   SOL02460
C                                                                       SOL02470
C-------------------------------------                                  SOL02480
      IZDMAX=IZ                                                         SOL02490
      IZ    =IZ+NDOF                                                    SOL02500
      IZVMAX=IZ                                                         SOL02510
      IZ    =IZ+NDOF                                                    SOL02520
      IZAMAX=IZ                                                         SOL02530
      IZ    =IZ+NDOF                                                    SOL02540
```

```
      IZRMAX=IZ                                                         SOL02560
      IZ    =IZ+NDOF +6                                                 SOL02570
      DO 28 I=0,NDOF                                                    SOL02580
        Z(IZDMAX+I)=0                                                   SOL02590
        Z(IZVMAX+I)=0                                                   SOL02600
        Z(IZAMAX+I)=0                                                   SOL02610
   28   Z(IZRMAX+I)=0                                                   SOL02620
      DO 29 I=1,6                                                       SOL02630
   29   Z(IZRMAX-1+NDOF+I)=0.                                           SOL02640
      DO 30 I=1,6                                                       SOL02650
   30   SUMRCT(I)=0.                                                    SOL02660
C                                                                       SOL02670
C-----------------------------------                                    SOL02680
C-INPUT FILE OUTPUT CONTROLL DATA ---                                   SOL02690
C-----------------------------------                                    SOL02700
      IF (IWRITE GT 0) THEN                                             SOL02710
        IOPT=1                                                          SOL02720
        CALL DMPDAT(IOPT,IWRITE,TC,DT)                                  SOL02730
      ENDIF                                                             SOL02740
C                                                                       SOL02750
C------ SET UP WORK STORAGE AREA    .                                   SOL02760
      IW    = IZ + 1                                                    SOL02770
C                                                                       SOL02780
C------ CHECK STORAGE REQUIRMENTS  ..                                   SOL02790
      CALL CKSTOR(NDOF,IW)                                              SOL02800
C                                                                       SOL02810
C--- DETERMINE THE LENGTH OF THE STIFFNESS MATRIX                       SOL02820
           IMD=NZ(IZMD+NDOF)  - NZ(IZMD) +1                             SOL02830
C---- DETERMINE THE LENGTH OF THE MASS MATRIX                           SOL02840
           IMDMS=NZ(IZMDMS+NDOF)   NZ(IZMDMS) +1                        SOL02850
C---- DETERMINE THE LENGTH OF THE GEOMETRIC STIFFNESS MATRIX            SOL02860
           IF (NZ(IZKGDT+3).EQ.2) THEN                                  SOL02870
             IMDKG =NZ(IZMDKG+NDOF) - NZ(IZMDKG) +1                     SOL02880
           ELSE                                                         SOL02890
             IMDKG =1                                                   SOL02900
           ENDIF                                                        SOL02910
C                                                                       SOL02920
C------ CALL SOL2A TO CALC DATA ....                                    SOL02930
      CALL SOL02A(IMD,IMDMS ,IMDKG     ,IMDS      ,IMDMSL    ,        SOL02940
     & NNODE    , NCOS     , NELMT    , IBUG     ,MAXNCD    ,         SOL02950
     & TITLE    , STEPID   , NAINP    , NDOF     , NFREE    ,         SOL02960
     & KGFORM   , KGLOAD   , KGTYPE   , NCOND    , INTEGO   ,         SOL02970
     & NEWKG    , MCOND    , KGCOND   , UNBAL    ,NEWK      ,         SOL02980
     & THETA    , ELSTIC   , IPRNT    , IWRITE   ,GROUND(1) ,         SOL02990
     & MAXACC   , SLMASS   , TO       , DT       , TF       ,GROUND(4),SOL03000
     & GRAV     , FDAMP    , IZDDSP   ,IZDLOA    ,GROUND(7),          SOL03010
     & Z(IZDISP), Z(IZVEL ), Z(IZACC ), Z(IZLOAD), Z(IZAN   ),Z(IZDMAX),SOL03020
     & Z(IZDDSP), Z(IZDVEL), Z(IZDACC), Z(IZDLOA), Z(IZBN   ),Z(IZVMAX),SOL03030
     & Z(IZSTIF), Z(IZMASS), Z(IZSLMS), Z(IZKG   ), Z(IZS    ),Z(IZAMAX),SOL03040
     & NZ(IZMD  ),NZ(IZMDMS),NZ(IZSLMD),NZ(IZMDKG),NZ(IZMDS ),Z(IZRMAX),SOL03050
     & Z(IZDAMP), Z(IZPBAR), Z(IZMCOS), Z(IZAGTX),          Z(IZDLO2),SOL03060
     & Z(IZAGTY), Z(IZAGTZ), Z(IZAINP), Z(IZFUB ), Z(IW     ),  SUMRCT ,SOL03070
     & NZ(IZID  ),NZ(IZIDOF), Z(IZCNST),NZ(IZJPLG), Z(IZCORD),Z(IZDFD ),SOL03080
     & Z(IZCOS ),NZ(IZJCOS), Z(IZTFD ),EIE,ESE,PSE,EKE,EDD,             SOL03090
     & Z(IZZERO), Z(IZPG)  , Z(IZDPG)  )                                SOL03100
C                                                                       SOL03110
      RETURN                                                            SOL03120
      END                                                               SOL03130
C                                                                       AVEA0010
C-----------------------------------------------------------------------AVEA0020
      SUBROUTINE AVEACC(IOPT,DT,THETA,NEWKDY,L2,L3,L4,                  AVEA0030
     &      IMDMAS,IMD,IMDS,IMDKG,NDOF,NFREE,KGLOAD,KGFORM,             AVEA0040
     &      A     ,V     ,DA    ,DV    ,DD    ,DP    ,                 AVEA0050
     &      WORK  ,MASS  ,MDMASS,DAMP  ,STIFF ,MD    ,                 AVEA0060
     &      KG    ,MDKG  ,ACC   ,S     ,MDS   ,Q     ,R     ,PBAR  ,   AVEA0070
     &      FDAMP,ALPHA,BETA)                                           AVEA0080
C                                                                       AVEA0090
      INTEGER FDAMP                                                     AVEA0100
      REAL MASS,KG                                                      AVEA0110
      LOGICAL NEWKDY                                                    AVEA0120
      DIMENSION A(NDOF),V(NDOF),DA(NDOF),DV(NDOF),DD(NDOF)              AVEA0130
      DIMENSION DP(NDOF),WORK(2*NDOF),XG(IMDKG),MDKG(NDOF+1)            AVEA0140
      DIMENSION PBAR(L3:L4),Q(L3:L4),R(L3.L4),MD(NDOF+1),STIFF(IMD)     AVEA0150
      DIMENSION MDMASS(NDOF-1),MASS(IMDMAS),MDS(NDOF+1),S(IMDS),DAMP(3) AVEA0160
C                                                                       AVEA0170
C------ ------------------------------------------------------------   AVEA0180
C     DETERMINE THE RESPONSE AT TIME DT*2                               AVEA0190
C                                                                       AVEA0200
      TWODT=DT*2.0                                                      AVEA0210
      THETA=2.0                                                         AVEA0220
      CALL LINACC(IOPT,TWODT,THETA,NEWKDY,L2,L3,L4,                     AVEA0230
     &      IMDMAS,IMD,IMDS,IMDKG,NDOF,NFREE,KGLOAD,KGFORM,             AVEA0240
     &      A     ,V     ,DA    ,DV    ,DD    ,DP    ,                 AVEA0250
     &      WORK  ,MASS  ,MDMASS,DAMP  ,STIFF ,MD    ,                 AVEA0260
     &      KG    ,MDKG  ,ACC   ,S     ,MDS   ,Q     ,R     ,PBAR  ,   AVEA0270
     &      FDAMP,ALPHA,BETA)                                           AVEA0280
C                                                                       AVEA0290
      IF (IOPT.NE.2) RETURN                                             AVEA0300
C                                                                       AVEA0310
C------- DETERMINE ACCELERATION, VELOCITY AND DISPLACEMENT AT TIME DT   AVEA0320
C       DA = INCRIMENTAL ACCELERATION BETWEEN TIMES T AND T+DT          AVEA0330
C       DV = INCRIMENTAL VELOCTIY BETWEEN TIMES T AND T+DT              AVEA0340
C       DD = INCRIMENTAL DISPLACMENT BETWEEN TIMES T AND T+DT           AVEA0350
C                                                                       AVEA0360
      DO 270 I=L3,L4                                                    AVEA0370
        DSP2DT=DD(I)                                                    AVEA0380
        VEL2DT=DV(I)                                                    AVEA0390
        ACC2DT=DA(I)                                                    AVEA0400
        DA(I)= ACC2DT/2.00                                              AVEA0410
        DV(I)= (4*A(I) + ACC2DT)*DT/4.                                  AVEA0420
  270   DD(I)=V(I)*DT + (6.00*A(I) + ACC2DT)*DT*DT/12.                  AVEA0430
C                                                                       AVEA0440
      RETURN                                                            AVEA0450
      END                                                               AVEA0460
C                                                                       WILS0010
C-----------------------------------------------------------------------WILS0020
      SUBROUTINE WILSON(IOPT,DT,THETA,XG ,MDKG,ACC,                     WILS0030
     &      L2,L3,L4,IMDMAS,IMD,IMDS,IMDKG,NDOF,NFREE,                  WILS0040
     &      KGLOAD,KGFORM,A     ,V     ,D     ,P     ,                 WILS0050
     &      DA    ,DV    ,DD    ,DP    ,WORK  ,                        WILS0060
     &      MASS  ,MDMASS,DAMP  ,STIFF ,MD    ,                        WILS0070
     &      S     ,MDS   ,Q     ,R     ,PBAR  ,                        WILS0080
     &      FDAMP,ALPHA,BETA)                                           WILS0090
C                                                                       WILS0100
      INTEGER FDAMP                                                     WILS0110
      REAL MASS,KG                                                      WILS0120
      DIMENSION A(NDOF),V(NDOF),D(NDOF),DA(NDOF),DV(NDOF),DD(NDOF)      WILS0130
      DIMENSION P(NDOF),DP(NDOF),DUMMY(1),IDUMMY(1),WORK(2*NDOF)        WILS0140
      DIMENSION MDMASS(NDOF-1),MASS(IMDMAS),S(IMDS),STIFF(IMD)          WILS0150
      DIMENSION MDKG(NDOF+1),XG(IMDKG),MDS(NDOF+1),MD(NDOF+1)           WILS0160
      DIMENSION PBAR(L3:L4),Q(L3.L4),R(L3:L4),DAMP(3)                   WILS0170
C                                                                       WILS0180
      GO TO (100,200,300) IOPT                                          WILS0190
C                                                                       WILS0200
C----------------------                                                 WILS0210
C- INITIALIZE VALUES ---  -                                             WILS0220
C----------------------                                                 WILS0230
  100 CONTINUE                                                          WILS0240
      C1=6 /(THETA*DT)**2                                               WILS0250
      C2=3./(THETA*DT)                                                  WILS0260
      C3=6./(THETA*DT)                                                  WILS0270
      C4=THETA*DT/2                                                     WILS0280
      C5=DT/2                                                           WILS0290
      C6=DT**2  /2                                                      WILS0300
      C7=DT**2  /6                                                      WILS0310
      ALPHA=DAMP(1)                                                     WILS0320
      BETA =DAMP(2)                                                     WILS0330
      C10=1+BETA*C2                                                     WILS0340
      C11=C1+ALPHA*C2                                                   WILS0350
C                                                                       WILS0360
C------- DEVELOP SKYLINE OF DYNAMIC STIFFNESS S                         WILS0370
C..... LSTIFF = # OF STIFFNESS TERMS IN COLUMN J                        WILS0380
C       LMASS  = # OF MASS TERMS IN COLUMN J                            WILS0390
C     ... LKG    = # OF KG TERMS IN COLUMN J                          WILS0400
C                                                                       WILS0410
      RETURN                                                            WILS0420
C                                                                       WILS0430
C-----------------------------------------------------------------------WILS0440
C- SOLVE FOR INCRIMENTAL DISPLACEMENTS, VELOCITIES AND ACCELERATIONS --WILS0450
C-----------------------------------------------------------------------WILS0460
  200 CONTINUE                                                          WILS0470
C                                                                       WILS0480
C------ - FORM DYNAMIC STIFFNESS                                        WILS0490
      IF (FDAMP.EQ 1) THEN                                              WILS0500
C                                                                       WILS0510
C . . .. SOME LOCAL VARIABLES                                           WILS0520
C J     - COLUMN NUMBER FOR MASS AND STIFF                              WILS0530
C I     - ROW NUMBER FOR MASS AND STIFF                                 WILS0540
C J1    - COLUMN NUMBER FOR S                                           WILS0550
C I1    - ROW NUMBER FOR S                                              WILS0560
C IJSTIF - ADDRESS OF ELEMENT I,J IN MATRIX STIFF                       WILS0570
C IJMASS - ADDRESS OF ELEMENT I,J IN MATRIX MASS                        WILS0580
C IJKG   - ADDRESS OF ELEMENT I,J IN GEOMETRIC STIFFNESS MATRIX         WILS0590
C IJS    - ADDRESS OF ELEMENT I1,J1 IN MATRIX S                         WILS0600
C LSTIFF - ROW # CONTAINING THE FIRST NONZERO TERM IN COLUMN J OF STIFF WILS0610
C LMASS  - ROW # CONTAINING THE FIRST NONZERO TERM IN COLUMN J OF MASS  WILS0620
C LKG    - ROW # CONTAINING THE FIRST NONZERO TERM IN COLUMN J OF KG    WILS0630
C LS     - ROW # CONTAINING THE FIRST NONZERO TERM IN COLUMN J OF S     WILS0640
C                                                                       WILS0650
C     DETERMINE THE FACTOR FOR KG                                       WILS0660
      IF (KGLOAD EQ.1) THEN                                             WILS0670
            C10KG=C10*(1-ACC)                                           WILS0680
      ELSE IF (KGLOAD.EQ.2) THEN                                        WILS0690
            C10KG=C10                                                   WILS0700
      ENDIF                                                             WILS0710
C                                                                       WILS0720
        DO 225 J=L3,L4                                                  WILS0730
          J1=J-L3+1                                                     WILS0740
          LSTIFF=J-(MD(J+1)-MD(J)-1)                                    WILS0750
          LS    =J1-(MDS(J1+1)-MDS(J1)-1)-L2                            WILS0760
          L=MAX(LS,LSTIFF)                                              WILS0770
          DO 210 I=J,L,-1                                               WILS0780
            I1=I-L3+1                                                   WILS0790
            IJS=(MDS(J1)+J1)-I1                                         WILS0800
            IJSTIF=(MD(J)+J) I                                          WILS0810
  210       S(IJS)=STIFF(IJSTIF)*C10                                    WILS0820
C                                                                       WILS0830
          IF (LS.LT LSTIFF) THEN                                        WILS0840
            DO 215 I=L-1,LS,-1                                          WILS0850
              IJS=(MDS(J1)+J1)-I1                                       WILS0860
  215         S(IJS)=0                                                  WILS0870
          ENDIF                                                         WILS0880
C                                                                       WILS0890
          LMASS=J-(MDMASS(J-1)-MDMASS(J)-1)                             WILS0900
          L=MAX(LS,LMASS)                                               WILS0910
          DO 220 I=J,L,-1                                               WILS0920
            I1=I-L3+1                                                   WILS0930
            IJS=(MDS(J1)+J1)-I1                                         WILS0940
            IJMASS=(MDMASS(J)+J)-I                                      WILS0950
  220       S(IJS)=MASS(IJMASS)*C11 + S(IJS)                          WILS0960
C                                                                       WILS0970
          IF (KGFORM.EQ.2) THEN                                         WILS0980
C                                                                       WILS0990
C           NOTE: THE GEOMETRIC STIFFNESS WAS DERIVED WITH COMPRESSION  WILS1000
C           BEING POSITIVE, KG HAS POSITIVE TERMS ON THE MAIN DIAGIONAL WILS1010
C           REPRESENTING COMPRESSION, THUS KG IS SUBTRACTED FROM S      WILS1020
C           CKG IS  AN INCREASE FOR VERT ACCEL                          WILS1030
      CKG=1.00                                                          WILS1040
C     ????????                                                          WILS1050
C                                                                       WILS1060
C---- K(BAR) = K(BAR) - CKG*GEOMETRIC STIFFNESS                         WILS1070
            LKG=J-(MDKG(J+1)-MDKG(J)-1)                                 WILS1080
            L=MAX(LS,LKG)                                               WILS1090
            DO 222 I=J,L,-1                                             WILS1100
              I1=I-L3+1                                                 WILS1110
              IJS=(MDS(J1)+J1)-I1                                       WILS1120
              IJKG=(MDKG(J)+J)-I                                        WILS1130
  222         S(IJS)=S(IJS) - XG(IJKG)*CKG                              WILS1140
          ENDIF                                                         WILS1150
C                                                                       WILS1160
  225   CONTINUE                                                        WILS1170
      ENDIF                                                             WILS1180
C                                                                       WILS1190
C--- --- CALCULATE Q AND R VECTORS                                      WILS1200
C..... QI IS TEMPROARLY STORED IN DA                                    WILS1210
C       Q(I) AND R(I) ARE MULTIPLIED BUY THE PROPER COIEPICIENTS FOR    WILS1220
C       PROPORTIONAL DAMPING                                            WILS1230
      DO 230 I=L3,L4                                                    WILS1240
        QI= C1*V(I) +  3*A(I)                                           WILS1250
        RI=  3*V(I) + C4*A(I)                                           WILS1260
        DA(I)= QI                                                       WILS1270
        Q(I)= QI + ALPHA*RI                                             WILS1280
  230   R(I)=      BETA *RI                                             WILS1290
C                                                                       WILS1300
C------- FORM DYNAMIC LOAD                                              WILS1310
C                                                                       WILS1320
C... ... INITILIZE PBAR-DP*THETA                                        WILS1330
      DO 240 I=L3,L4                                                    WILS1340
  240   PBAR(I)=  DP(I)*THETA                                           WILS1350
C                                                                       WILS1360
      IF (FDAMP.EQ.1) THEN                                              WILS1370
C                                                                       WILS1380
C..... PBAR=PBAR +MASS*Q                                                WILS1390
        DO 250 J=L3,L4                                                  WILS1400
          MDJ=MDMASS(J)+J                                               WILS1410
          M=J-(MDMASS(J+1)-MDMASS(J))+1                                 WILS1420
          M=MAX(M,L3)                                                   WILS1430
          DO 250 I=M,J                                                  WILS1440
            PBAR(I)=PBAR(I)+ MASS(MDJ-I)*Q(J)                           WILS1450
  250       CONTINUE                                                    WILS1460
        DO 255 I=L3+1,L4                                                WILS1470
          MDI=MDMASS(I)+I                                               WILS1480
          M=I-(MDMASS(I+1)-MDMASS(I))+1                                 WILS1490
          M=MAX(M,L3)                                                   WILS1500
          DO 255 J=M,I-1                                                WILS1510
            PBAR(I)=PBAR(I)+ MASS(MDI-J)*Q(J)                           WILS1520
  255       CONTINUE                                                    WILS1530
C                                                                       WILS1540
C... ..  PBAR=BBAR -STIFF*R                                             WILS1550
        DO 260 J=L3,L4                                                  WILS1560
          MDJ=MD(J)+J                                                   WILS1570
          M=J-(MD(J+1)-MD(J))+1                                         WILS1580
          M=MAX(M,L3)                                                   WILS1590
          DO 260 I=M,J                                                  WILS1600
            PBAR(I)=PBAR(I)- STIFF(MDJ-I)*R(J)                          WILS1610
  260       CONTINUE                                                    WILS1620
        DO 265 I=L3+1,L4                                                WILS1630
          MDI=MD(I)+I                                                   WILS1640
          M=I-(MDS(I+1)-MD(I))+1                                        WILS1650
          M=MAX(M,L3)                                                   WILS1660
          DO 265 J=M,I-1                                                WILS1670
            PBAR(I)=PBAR(I)+ STIFF(MDI J)*R(J)                          WILS1680
  265       CONTINUE                                                    WILS1690
      ENDIF                                                             WILS1700
C                                                                       WILS1710
C------- SOLVE FOR DISPLACEMENT AT TIME THETA*DT                        WILS1720
C..... NOTE:  - WORK IS USED FOR TEMPORARY WORK SPACE                   WILS1730
C               - PBAR CONTAINS THE INCRIMENTAL DISPLACMENT BETWEEN     WILS1740
C                 TIMES T AND T+THETA*DT AFTER THE SECOND CALL TO MSOLL WILS1750
C                                                                       WILS1760
      CALL MSOLL(1, FALSE ,NFREE,IMDS,NFREE,1,1,NFREE,MDS,S,PBAR,WORK, WILS1770
     &           .FALSE.,DUMMY,IDUMMY,1,.FALSE.,DUMMY,IDUMMY,1)         WILS1780
      CALL MSOLL(3, FALSE.,NFREE,IMDS,NFREE,1,1,NFREE,MDS,S,PBAR,WORK, WILS1790
     &           .FALSE.,DUMMY,IDUMMY,1,.FALSE.,DUMMY,IDUMMY,1)         WILS1800
C                                                                       WILS1810
C                                                                       WILS1820
C------- DETERMINE ACCELERATION, VELOCITY AND DISPLACEMENT AT TIME DT   WILS1830
C..... RECALL THAT QI WAS TEMPORARLY STORED IN DA(I)                    WILS1840
C       DA = INCRIMENTAL ACCELERATION BETWEEN TIMES T AND T+DT          WILS1850
C       DV = INCRIMENTAL VELOCTIY BETWEEN TIMES T AND T+DT              WILS1860
C       DD = INCRIMENTAL DISPLACMENT BETWEEN TIMES T AND T+DT           WILS1870
C                                                                       WILS1880
```

```
      DO 270 I=L3,L4                                                    WILS1890
        DA(I)=( C1*PBAR(I) -  DA(I) )/THETA                             WILS1900
        DV(I)=A(I)*DT + C5*DA(I)                                        WILS1910
  270   DD(I)=V(I)*DT + C6* A(I) + C7*DA(I)                             WILS1920
C                                                                       WILS1930
      RETURN                                                            WILS1940
C                                                                       WILS1950
C-----------------------------------------------------------           WILS1960
C- CALC TOTAL DISPLACEMENTS, VELOCITIES AND ACCELERATIONS --           WILS1970
C-----------------------------------------------------------           WILS1980
  300 CONTINUE                                                          WILS1990
      DO 310 I=1,NDOF                                                   WILS2000
        A(I)=DA(I) + A(I)                                               WILS2010
        V(I)=DV(I) + V(I)                                               WILS2020
        D(I)=DD(I) + D(I)                                               WILS2030
        P(I)=DP(I) + P(I)                                               WILS2040
  310   DP(I)=0.00                                                      WILS2050
C                                                                       WILS2060
      RETURN                                                            WILS2070
      END                                                               WILS2080
C                                                                       LINA0010
C-----------------------------------------------------------------------LINA0020
C                                                                       LINA0030
      SUBROUTINE LINACC(IOPT,DT,THETA,NEWKDY,L2,L3,L4,                  LINA0040
     &          IMDMAS,IMD,IMDS,IMDKG,NDOF,NFREE,KGLOAD,KGFORM,         LINA0050
     &          A     ,V     ,DA    ,DV    ,DD    ,DP    ,            LINA0060
     &          WORK  ,MASS  ,MDMASS,DAMP  ,STIFF ,MD    ,            LINA0070
     &          KG    ,MDKG  ,ACC   ,S     ,MDS   ,Q     ,R     ,PBAR  ,LINA0080
     &          FDAMP,ALPHA,BETA)                                       LINA0090
C                                                                       LINA0100
      INTEGER FDAMP                                                     LINA0110
      REAL MASS,KG                                                      LINA0120
      LOGICAL NEWKDY                                                    LINA0130
      DIMENSION A(NDOF),V(NDOF),DA(NDOF),DV(NDOF),DD(NDOF)              LINA0140
      DIMENSION DP(NDOF),DUMMY(1),IDUMMY(1),WORK(2*NDOF)                LINA0150
      DIMENSION PBAR(L3:L4),Q(L3:L4),R(L3:L4),MD(NDOF+1),STIFF(IMD)     LINA0160
      DIMENSION MDMASS(NDOF+1),MASS(IMDMAS),MDKG(NDOF+1),KG(IMDKG)      LINA0170
      DIMENSION MDS(NDOF+1)    ,S(IMDS)     ,DAMP(3)                    LINA0180
C                                                                       LINA0190
      GO TO (100,200,300) IOPT                                          LINA0200
C                                                                       LINA0210
C----------------------                                                 LINA0220
C- INITIALIZE VALUES ---                                               LINA0230
C----------------------                                                 LINA0240
  100 CONTINUE                                                          LINA0250
      ALPHA=DAMP(1)                                                     LINA0260
      BETA =DAMP(2)                                                     LINA0270
      C0=3 *ALPHA/DT + 6./(DT**2)                                       LINA0280
      C1=1.0/(1+3.*BETA/DT)                                             LINA0290
      C2=C0*C1                                                          LINA0300
      C3=ALPHA-C2*BETA                                                  LINA0310
      C4=6./DT                                                          LINA0320
      C5=DT/2.                                                          LINA0330
      C6=0.                                                             LINA0340
      C10=3./DT                                                         LINA0350
      C11=6./(DT**2)                                                    LINA0360
C                                                                       LINA0370
C------- DEVELOP SKYLINE OF DYNAMIC STIFFNESS (S)                       LINA0380
C         MDS(I)    - ADDRESS OF DYNAMIC STIFF. MAIN DIAGONAL TERM ROW I LINA0390
C    ..   MD(I)     - ADDRESS OF STIFFNESS MAIN DIAGONAL TERM ROW I      LINA0400
C....     MDMAS(I)  - ADDRESS OF MASS MAIN DIAGONAL TERM ROW I          LINA0410
C... ... MDKG(I)   - ADDRESS OF GEOMETRIC STIFF. MAIN DIAGONAL TERM ROW ILINA0420
C                                                                       LINA0430
C. .... LSTIFF - # OF STIFFNESS TERMS IN COLUMN J                       LINA0440
C.....  LMASS  - # OF MASS TERMS IN COLUMN J                            LINA0450
C.....  LKG    - # OF KG TERMS IN COLUMN J                              LINA0460
C.. .... L      - # OF S  TERMS IN COLUMN J                             LINA0470
C                                                                       LINA0480
C------ CHECK FOR PROPORTIONAL DAMPING ....                            LINA0490
      IF (FDAMP.NE.1) THEN                                              LINA0500
         WRITE(108,*) '*****************************************'      LINA0510
         WRITE(108,*) '*  PROPORTIONAL DAMPING WAS NOT SPECIFIED  *'   LINA0520
         WRITE(108,*) '*  SOLUTION IS ABORTED...                 *'   LINA0530
         WRITE(108,*) '*****************************************'      LINA0540
         STOP                                                           LINA0550
      ENDIF                                                             LINA0560
      RETURN                                                            LINA0570
C                                                                       LINA0580
C-----------------------------------------------------------------------LINA0590
C- SOLVE FOR INCRIMENTAL DISPLACEMENTS, VELOCITIES AND ACCELERATIONS -- LINA0600
C-----------------------------------------------------------------------LINA0610
  200 CONTINUE                                                          LINA0620
C                                                                       LINA0630
C------- FORM DYNAMIC STIFFNESS                                         LINA0640
C. ... . SOME LOCAL VARIABLES                                           LINA0650
C J      - COLUMN NUMBER FOR MASS AND STIFF                             LINA0660
C I      - ROW NUMBER FOR MASS AND STIFF                                LINA0670
C J1     - COLUMN NUMBER FOR S                                          LINA0680
C I1     - ROW NUMBER FOR S                                             LINA0690
C IJSTIF - ADDRESS OF ELEMENT I,J IN MATRIX STIFF                       LINA0700
C IJMASS - ADDRESS OF ELEMENT I,J IN MATRIX MASS                        LINA0710
C IJKG   - ADDRESS OF ELEMENT I,J IN GEOMETRIC STIFFNESS MATRIX         LINA0720
C IJS    - ADDRESS OF ELEMENT I1,J1 IN MATRIX S                         LINA0730
C LSTIFF - ROW # CONTAINING THE FIRST NONZERO TERM IN COLUMN J OF STIFF LINA0740
C LMASS  - ROW # CONTAINING THE FIRST NONZERO TERM IN COLUMN J OF MASS  LINA0750
C LKG    - ROW # CONTAINING THE FIRST NONZERO TERM IN COLUMN J OF KG    LINA0760
C LS     - ROW # CONTAINING THE FIRST NONZERO TERM IN COLUMN J OF S     LINA0770
C                                                                       LINA0780
C-------- FORM DYNAMIC STIFFNESS IF NEWKDY IS TRUE ------------------- LINA0790
      IF (NEWKDY) THEN                                                  LINA0800
C                                                                       LINA0810
C     DETERMINE THE FACTOR FOR KG                                       LINA0820
      IF (KGLOAD EQ.1) THEN                                             LINA0830
            CKG=(1+ACC)                                                 LINA0840
      ELSE IF (KGLOAD.EQ.2) THEN                                        LINA0850
            CKG= 1.00                                                   LINA0860
      ENDIF                                                             LINA0870
C                                                                       LINA0880
C---- K(BAR) = STIFFNESS                                               LINA0890
        DO 225 J=L3,L4                                                  LINA0900
          J1=J-L3+1                                                     LINA0910
          MDJP=MD(J+1)                                                  LINA0920
          MDJ =MD(J)                                                    LINA0930
          LSTIFF=J-(MD(J+1)-MD(J)-1)                                    LINA0940
          MDSJ1P=MDS(J1+1)                                              LINA0950
          MDSJ1 =MDS(J1)                                                LINA0960
          LS    =J1-(MDS(J1+1)-MDS(J1)-1)+L3                            LINA0970
          L=MAX(LS,LSTIFF)                                              LINA0980
          DO 210 I=J,L,-1                                               LINA0990
            I1=I-L3+1                                                   LINA1000
            IJS=(MDS(J1)+J1)-I1                                         LINA1010
            IJSTIF=(MD(J)+J)-I                                          LINA1020
  210       S(IJS)=STIFF(IJSTIF)                                        LINA1030
C                                                                       LINA1040
          IF (LS.LT LSTIFF) THEN                                        LINA1050
            DO 215 I=L-1,LS,-1                                          LINA1060
              IJS=(MDS(J1)+J1)-I1                                       LINA1070
  215         S(IJS)=0                                                  LINA1080
          ENDIF                                                         LINA1090
C                                                                       LINA1100
C---- K(BAR) = K(BAR) + C2*MASS                                        LINA1110
          LMASS=J-(MDMASS(J+1)-MDMASS(J)-1)                             LINA1120
          L=MAX(LS,LMASS)                                               LINA1130
          DO 220 I=J,L,-1                                               LINA1140
            I1=I-L3+1                                                   LINA1150
            IJS=(MDS(J1)+J1)-I1                                         LINA1160
            IJMASS=(MDMASS(J)+J)-I                                      LINA1170
  220       S(IJS)=MASS(IJMASS)*C2  + S(IJS)                          LINA1180
C                                                                       LINA1190
          IF (KGFORM.EQ.2) THEN                                         LINA1200
C                                                                       LINA1210
C.          NOTE  THE GEOMETRIC STIFFNESS WAS DERIVED WITH COMPRESSION  LINA1220
C.          BEING POSITIVE, KG HAS POSITIVE TERMS ON THE MAIN DIAGIONAL LINA1230
C.          REPRESENTING COMPRESSION, THUS KG IS SUBTRACTED FROM S      LINA1240
C           CKG IS  AN INCREASE FOR VERT ACCEL                          LINA1250
C                                                                       LINA1260
C  -- K(BAR) = K(BAR) - CKG*GEOMETRIC STIFFNESS                         LINA1270
            LKG=J-(MDKG(J+1)-MDKG(J)-1)                                 LINA1280
            L=MAX(LS,LKG)                                               LINA1290
            DO 222 I=J,L,-1                                             LINA1300
              I1=I-L3+1                                                 LINA1310
              IJS=(MDS(J1)+J1) I1                                       LINA1320
              IJKG=(MDKG(J)-J) I                                        LINA1330
  222       S(IJS)=S(IJS) - KG(IJKG)*CKG                              LINA1340
          ENDIF                                                         LINA1350
C                                                                       LINA1360
  225   CONTINUE                                                        LINA1370
      ENDIF                                                             LINA1380
C                                                                       LINA1390
C------- CALCULATE Q AND R VECTORS                                      LINA1400
      DO 230 I=L3,L4                                                    LINA1410
        Q(I)= C4*V(I) +  3*A(I)                                         LINA1420
  230   R(I)=  3*V(I) + C5*A(I)                                         LINA1430
C                                                                       LINA1440
C------- FORM DYNAMIC LOAD                                              LINA1450
C                                                                       LINA1460
C  ...   INITILIZE PBAR= DP*THETA                                       LINA1470
C       THETA=1 FOR LINEAR ACCELERATION                                 LINA1480
C       THETA=2 WHEN LINACC IS CALLED BY AVERAGE ACCELERATION           LINA1490
      DO 240 I=L3,L4                                                    LINA1500
  240   PBAR(I)=  DP(I)*THETA                                           LINA1510
      IF (FDAMP.EQ 1) THEN                                              LINA1520
C                                                                       LINA1530
C.....  PBAR=PBAR +MASS*( Q(I) + C3*R(I) )                             LINA1540
        DO 250 J=L3,L4                                                  LINA1550
          MDJ=MDMASS(J)+J                                               LINA1560
          M=J-(MDMASS(J+1)-MDMASS(J))+1                                 LINA1570
          M=MAX(M,L3)                                                   LINA1580
          DO 250 I=M,J                                                  LINA1590
            PBAR(I)=PBAR(I)+ MASS(MDJ-I)*( Q(J)+C3*R(J) )               LINA1600
  250     CONTINUE                                                      LINA1610
        DO 255 I=L3+1,L4                                                LINA1620
          MDI=MDMASS(I)+I                                               LINA1630
          M=I-(MDMASS(I+1)-MDMASS(I))+1                                 LINA1640
          M=MAX(M,L3)                                                   LINA1650
          DO 255 J=M,I-1                                                LINA1660
            PBAR(I)=PBAR(I)+ MASS(MDI-J)*( Q(J)+C3*R(J) )               LINA1670
  255     CONTINUE                                                      LINA1680
      ENDIF                                                             LINA1690
C                                                                       LINA1700
C                                                                       LINA1710
C------- SOLVE FOR INCRIMENTAL DISPLACEMENT OVER TIME INTERVAL DT       LINA1720
C..  ... NOTE.  WORK IS USED FOR TEMPORARY WORK SPACE                   LINA1730
C               IF NEWKDY=.FALSE. THE DYNAMIC STIFFNESS HAS ALREADY     LINA1740
C                    BEEN REDUCED TO UPPER TRIANGULAR FORM              LINA1750
C                                                                       LINA1760
      IF (NEWKDY) THEN                                                  LINA1770
         CALL MSOLL(1,.FALSE.,NFREE,IMDS,NFREE,1,1,NFREE,MDS,S,PBAR,    LINA1780
     &           WORK,.FALSE.,DUMMY,IDUMMY,1, FALSE.,DUMMY,IDUMMY,1)    LINA1790
      ELSE                                                              LINA1800
         CALL MSOLL(2,.FALSE.,NFREE,IMDS,NFREE,1,1,NFREE,MDS,S,PBAR,    LINA1810
     &           WORK,.FALSE.,DUMMY,IDUMMY,1, FALSE.,DUMMY,IDUMMY,1)    LINA1820
      ENDIF                                                             LINA1830
C                                                                       LINA1840
C ..... NOTE  - PBAR CONTAINS THE INCRIMENTAL DISPLACMENT BETWEEN       LINA1850
C               TIMES T AND T+DT AFTER THE SECOND CALL TO MSOLL         LINA1860
      CALL MSOLL(3,.FALSE.,NFREE,IMDS,NFREE,1,1,NFREE,MDS,S,PBAR,       LINA1870
     &           WORK,.FALSE.,DUMMY,IDUMMY,1, FALSE.,DUMMY,IDUMMY,1)    LINA1880
C                                                                       LINA1890
C------- DETERMINE ACCELERATION, VELOCITY AND DISPLACEMENT AT TIME DT   LINA1900
C       DA = INCRIMENTAL ACCELERATION BETWEEN TIMES T AND T+DT          LINA1910
C       DV = INCRIMENTAL VELOCTIY BETWEEN TIMES T AND T+DT              LINA1920
C       DD = INCRIMENTAL DISPLACMENT BETWEEN TIMES T AND T+DT           LINA1930
C                                                                       LINA1940
      DO 270 I=L3,L4                                                    LINA1950
        DD(I)=C1*( PBAR(I) + BETA*R(I) )                                LINA1960
        DV(I)=C10*DD(I)   R(I)                                          LINA1970
        DA(I)=C11*DD(I) - Q(I)                                          LINA1980
  270   CONTINUE                                                        LINA1990
C                                                                       LINA2000
      RETURN                                                            LINA2010
C                                                                       LINA2020
C-----------------------------------------------------------           LINA2030
C- CALC TOTAL DISPLACEMENTS, VELOCITIES AND ACCELERATIONS --           LINA2040
C-----------------------------------------------------------           LINA2050
  300 CONTINUE                                                          LINA2060
C                                                                       LINA2070
      RETURN                                                            LINA2080
      END                                                               LINA2090
C                                                                       ENER0010
C-----------------------------------------------------------------------ENER0020
C   ENERGY BALANCE                                                      ENER0030
C-----------------------------------------------------------------------ENER0040
      SUBROUTINE ENERGY(L1,L2,L3,L4,L5,MAXNOD,NCOS,PT2,                 ENER0050
     &          IMDMS,MDMS,DYN,WRITO,MASS,EIE,ESE,PSE,EKE,EDD,          ENER0060
     &          ALPHA,BETA,KGFORM,NELMT,NNODE,DT,SUMUB,SUMRC,           ENER0070
     &          DSUMRC,SUMFD,DSUMFD,FDAMP,DFDAMP,GA,GV,GD,GAT,          ENER0080
     &          GVT,GDT,WORK,WORK2,PG,DPG,DACC,DVEL,DDISP,ACC,          ENER0090
     &          VEL,DISP,LOAD,DLOAD,IDOF,JTFLG,COSINE,JTCOS)            ENER0100
C                                                                       ENER0110
      COMMON /ISPEC/ISP1,SGD(500,2),REACF(6)                            ENER0120
      REAL MASS,LOAD                                                    ENER0130
      CHARACTER*80 NAME                                                 ENER0140
      LOGICAL BTEST,PT2,AXIAL,DYN,WRITO                                 ENER0150
      DIMENSION JTFLG(MAXNOD,6) ,COSINE(3,3,NCOS),JTFLG(MAXNOD)         ENER0160
      DIMENSION MASS(IMDMS),MDMS(L5-1),JTCOS(MAXNOD)                    ENER0170
      DIMENSION RINPUT(100),FDAMP(L5),DFDAMP(L5),PG(6),DPG(6)           ENER0180
      DIMENSION  SUMRC(6),DSUMRC(6),SUMFD(6),DSUMFD(6),SUMUB(6)         ENER0190
      DIMENSION  GD(3), GV(3), GA(3),GDT(3),GVT(3),GAT(3)               ENER0200
      DIMENSION DDISP(L6),DVEL(L6),DACC(L6)                             ENER0210
      DIMENSION  DISP(L6), VEL(L6), ACC(L6)                             ENER0220
      DIMENSION WORK(L6),WORK2(L6),LOAD(L6),DLOAD(L6)                   ENER0230
C                                                                       ENER0240
C------------------ NOMENCLATURE ---------------------------------------CENER0250
C  DACC  INCRIMENTAL RELATIVE ACCELERATION                              CENER0260
C  DVEL  INCRIMENTAL RELATIVE VELOCITY                                  CENER0270
C  DDISP INCRIMENTAL RELATIVE DISPLACEMENT                              CENER0280
C  ACC   PREVIOUS STEP'S RELATIVE ACCELERATION                          CENER0290
C  VEL   PREVIOUS STEP'S RELATIVE VELOCITY                              CENER0300
C  DISP  PREVIOUS STEP'S RELATIVE DISPLACEMENT                          CENER0310
C  DLOAD INCRIMENTAL APPLIED LOAD, NOT M*AG                             CENER0320
C  LOAD  PREVIOUS STEP'S APPLIED LOAD, NOT M*AG                         CENER0330
C  DFDAMP INCRIMENTAL DAMPING FORCE                                     CENER0340
C  FDAMP PREVIOUS STEP'S DAMPING FORCE                                  CENER0350
C                                                                       CENER0360
C  GD    INCRIMENTAL GROUND DISPLACEMENT                                CENER0370
C  GV    INCRIMENTAL GROUND VELOCITY                                    CENER0380
C  GA    INCRIMENTAL GROUND ACCELERATION                                CENER0390
C  GDT   TOTAL GROUND DISPLACEMENT                                      CENER0400
C  GVT   TOTAL GROUND VELOCITY                                          CENER0410
C  GAT   TOTAL GROUND ACCELERATION                                      CENER0420
C  WORK  TEMP FORCE VECTOR                                              CENER0430
C  WORK2 TEMP DISPL OR VELOCITY VECTOR                                  CENER0440
C  ------------------------------------------------------------------- CENER0450
C                                                                       ENER0460
      L41=L4+1                                                          ENER0470
C                                                                       ENER0480
C---------------------------------------                                ENER0490
C---  INTEGRATE GROUND ACCELERATION  ---                               ENER0500
C---------------------------------------                                ENER0510
      IF (DYN) THEN                                                     ENER0520
      DO 10 I=1,3                                                       ENER0530
         GA(I)=GA(I)                                                    ENER0540
         GV(I)=(3*GAT(I)+GA(I)) *DT /2                                  ENER0550
         GD(I)=GVT(I)*DT + (3*GAT(I)+GA(I)) *DT*DT /6                   ENER0560
         GAT(I)=GA(I)+GAT(I)                                            ENER0570
         GVT(I)=GV(I)+GVT(I)                                            ENER0580
   10    GDT(I)=GD(I)+GDT(I)                                            ENER0590
      ENDIF                                                             ENER0600
       IF (PT2) THEN                                                    ENER0610
        IF (DYN) THEN                                                   ENER0620
             WRITE(108,22) (I,ACC(I),VEL(I),DISP(I),LOAD(I),I=1,L4)     ENER0630
```

```
            WRITE(108,23) (I,DACC(I),DVEL(I),DDISP(I),DLOAD(I),I=1,L4) ENER0640
            WRITE(108,24) (I,FG(I),DFG(I),FDAMP(I),DFDAMP(I),I=1,L6)  ENER0650
          ELSE                                                        ENER0660
            WRITE(106,22) (I,ACC(I),VEL(I),DISP(I),LOAD(I),I=1,L6)    ENER0670
            WRITE(108,23) (I,DACC(I),DVEL(I),DDISP(I),DLOAD(I),I=1,L6) ENER0680
            WRITE(108,24) (I,FG(I),DFG(I),FDAMP(I),DFDAMP(I),I=1,L6)  ENER0690
          ENDIF                                                       ENER0700
   22   FORMAT(/T10,'TOTAL RELATIVE RESPONSE FOR PREVIOUS TIME STEP'/ ENER0710
     &       (T10,'DOF:',I6,' ACC ',1P,G13.5,' VEL.',G13.5,          ENER0720
     &        ' DSP:',G13.5,'LOAD.',G13.5))                          ENER0730
   23   FORMAT(/T10,'INCRIMENTAL RELATIVE RESPONSE'/                  ENER0740
     &       (T10,'DOF ',I6,' ACC:',1P,G13.5,' VEL.',G13.5,          ENER0750
     &        ' DSP.',G13.5,'LOAD:',G13.5))                          ENER0760
   24   FORMAT(/T10,'SPECIAL FORCES '/(T10,'DOF:',I6,' FD:',1P,G13.5, ENER0770
     &       ' DFD.',G13.5,' FDAMP ',G13.5,' DFDAMP:',G13.5))        ENER0780
        ENDIF                                                         ENER0790
C                                                                     ENER0800
C-------------------------------------                              ENER0810
C---  CALCULATE INPUT EARTHQUAKE ENERGY  ---                        ENER0820
C-------------------------------------                              ENER0830
C                                                                     ENER0840
      DEIE=0                                                          ENER0850
C                                                                     ENER0860
C     ..CALCULATE THE INCR  INPUT ENERGY DUE TO GRND MOTION           ENER0870
      IF (DYN) THEN                                                   ENER0880
          IF (PT2) THEN                                               ENER0890
           WRITE(108,*) ' SUMRC', SUMRC                               ENER0900
           WRITE(108,*) 'DSUMRC',DSUMRC                               ENER0910
           WRITE(108,*) ' SUMFD', SUMFD                               ENER0920
           WRITE(108,*) 'DSUMFD',DSUMFD                               ENER0930
           WRITE(108,*) ' SUMUB', SUMUB                               ENER0940
          ENDIF                                                       ENER0950
          DO 48 I=1,3                                                 ENER0960
            FORCEI=-SUMRC(I)+0.5*DSUMRC(I) + SUMFD(I) + 0.5*DSUMFD(I) ENER0970
     &           + SUMUB(I)+ REACF(I)                                 ENER0980
            ENER=GD(I)*FORCEI                                         ENER0990
            IF (PT2) WRITE(108,*) I,'FORCEI',FORCEI,                  ENER1000
     &                             ' GD',GD(I),'  ENER',ENER          ENER1010
   48       DEIE=DEIE + ENER                                          ENER1020
        ENDIF                                                         ENER1030
C                                                                     ENER1040
C  . . . CALCULATE THE INCR. INPUT ENERGY DUE TO APPLIED LOADS  .     ENER1050
          N2=L6                                                       ENER1060
          IF (DYN) N2=L4                                              ENER1070
          DO 50 I=L3,N2                                               ENER1080
            FORCEI=LOAD(I)+DLOAD(I)/2                                 ENER1090
            ENER=FORCEI*DDISP(I)                                      ENER1100
            IF (PT2) WRITE(108,*) I,                                  ENER1110
     &           'FORCEI',FORCEI,'  DDISP',DDISP(I),'  ENER',ENER     ENER1120
   50       DEIE=DEIE + ENER                                          ENER1130
C                                                                     ENER1140
C.. . . CALCULATE THE ENERGY DUE TO GEOMETRIC STIFFNESS               ENER1150
        IF (KGFORM.EQ.2 ) THEN                                        ENER1160
C. ...    GET THE INCRIMENTAL DISPL AND STORE IN VECTOR WORK2 ....    ENER1170
C...      SAVE INCRIMENTAL IN VECTOR WORK2                            ENER1180
          DO 51 I=L3,L4                                               ENER1190
   51       WORK(I)=DDISP(I)                                          ENER1200
C ..      ADD GROUND RESPONSE IN JOINT CORRDINATES TO RELATIVE RESP   ENER1210
C           MULTIPLY GROUND DISPL * DIRECTION COSINE                  ENER1220
C           OMIT ALL DOF THAT ARE CONSTRAINED TO A MASTER DOF         ENER1230
          IF (GD(1).NE.0 .OR. GD(2).NE.0 .OR  GD(3) NE.0) THEN        ENER1240
            DO 52 NODE=1,NNODE                                        ENER1250
              ICOS=JTCOS(NODE)                                        ENER1260
              DO 52 K=1,3                                             ENER1270
              IF ( NOT  BTEST(JTFLG(NODE),K+11)) THEN                 ENER1280
                II=IDOF(NODE,K)                                       ENER1290
                IF (II LE.L4 .AND. II GE.L3)                          ENER1300
     &           WORK(II)=WORK(II)+ GD(1)*COSINE(1,K,ICOS)            ENER1310
     &                    + GD(2)*COSINE(2,K,ICOS)  + GD(3)*COSINE(3,K,ICOS) ENER1320
              ENDIF                                                   ENER1330
   52         CONTINUE                                                ENER1340
          ENDIF                                                       ENER1350
C                                                                     ENER1360
C. . . . CALCULATE THE INCRIMENTAL INPUT ENERGY DUE TO KG...          ENER1370
          DIKG=0                                                      ENER1380
          DO 53 I=L3,L4                                               ENER1400
            ENER=  (FG(I)+DFG(I)/2)*WORK(I)                           ENER1400
            IF (PT2) WRITE(108,*) I,                                  ENER1410
     &           'FG,DFG',FG(I),DFG(I),'  DDISP',WORK(I),'  ENER',ENER ENER1420
   53       DIKG=DIKG+ENER                                            ENER1430
          DEIE=DEIE+DIKG                                              ENER1440
        ENDIF                                                         ENER1450
C*                                                                    ENER1460
        IF ( ISP1  EQ. 1 ) THEN                                       ENER1470
          DO 61 I=L3,L4                                               ENER1480
   61       WORK(I)=DDISP(I)                                          ENER1490
C...      ADD GROUND RESPONSE IN JOINT CORRDINATES TO RELATIVE RESP   ENER1500
C           MULTIPLY GROUND DISPL * DIRECTION COSINE                  ENER1510
C           OMIT ALL DOF THAT ARE CONSTRAINED TO A MASTER DOF         ENER1520
          IF (GD(1) .NE.0 .OR. GD(2).NE.0 .OR  GD(3) .NE.0) THEN      ENER1530
            DO 62 NODE=1,NNODE                                        ENER1540
              ICOS=JTCOS(NODE)                                        ENER1550
              DO 62 K=1,3                                             ENER1560
              IF ( NOT. BTEST(JTFLG(NODE),K+11)) THEN                 ENER1570
                II=IDOF(NODE,K)                                       ENER1580
                IF (II.LE.L4 .AND. II.GE.L3)                          ENER1590
     &           WORK(II)=WORK(II)          + GD(1)*COSINE(1,K,ICOS)  ENER1600
     &                    + GD(2)*COSINE(2,K,ICOS) + GD(3)*COSINE(3,K,ICOS) ENER1610
              ENDIF                                                   ENER1620
   62         CONTINUE                                                ENER1630
         ENDIF                                                        ENER1640
        DEINT=0                                                       ENER1650
        DO 57 I=L3,L4                                                 ENER1660
          ENER=  SGD(I,1)*WORK(I)                                     ENER1670
   57     DEINT=DEINT+ENER                                            ENER1680
          DEIE=DEIE+DEINT                                             ENER1690
      ENDIF                                                           ENER1700
C*                                                                    ENER1710
C..... CALCULATE THE TOTAL INPUT ENERGY                               ENER1720
      EIE=EIE + DEIE                                                  ENER1730
C                                                                     ENER1740
C-------------------------------------                              ENER1750
C---  CALCULATE TOTAL STRAIN ENERGY   ---                           ENER1760
C-------------------------------------                              ENER1770
C ...  CALCULATE THE TOTAL STRAIN ENERGY                              ENER1780
      DESE=ESE                                                        ENER1790
      DPSE=PSE                                                        ENER1800
      ESE=0                                                           ENER1810
      PSE=0                                                           ENER1820
                                                                      ENER1830
C     ---- . GET ELASTIC AND PLASTIC STRAIN ENERGY FOR EACH ELEMENT   ENER1840
      LSTTYP=0                                                        ENER1850
      DO 60 IELNO=1,NELMT                                             ENER1860
        CALL ELELIB                                                   ENER1870
     &  (10,LSTTYP, FALSE.,IREL,.FALSE.,NAME,EESE,EPSE,DAMAGE,DUCFAG, ENER1880
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDISP,IZLOAD,MSTOR)  ENER1890
        IF (PT2) WRITE(108,55) IELNO,EESE,EPSE                        ENER1900
   55   FORMAT('   ELEMENT#',I5,'   ELASTIC SE=',1P,G13.4,           ENER1910
     &                          '   PLASTIC SE=',1P,G13.4)           ENER1920
        ESE = ESE + EESE                                              ENER1930
   60   PSE = PSE + EPSE                                              ENER1940
C                                                                     ENER1950
C..   . CALCULATE THE INCRIMENTAL STRAIN ENERGY                       ENER1960
      DESE=ESE-DESE                                                   ENER1970
      DPSE=PSE-DPSE                                                   ENER1980
C                                                                     ENER1990
C-------------------------------                                     ENER2000
C---  CALCULATE KINETIC ENERGY  ---                                 ENER2010
C-------------------------------                                     ENER2020
      IF (DYN) THEN                                                   ENER2030
C. . . .  GET THE ABSOLUTE VELOCITY AND STORE IN VECTOR WORK2..       ENER2040
C...      SAVE RELATIVE RESPONSE IN VECTOR WORK2                      ENER2050
          DO 15 I=1,L4                                                ENER2060
   15       WORK2(I)=VEL(I)+DVEL(I)                                   ENER2070
C ..      ADD GROUND RESPONSE IN JOINT CORRDINATES TO RELATIVE RESP   ENER2080
C           MULTIPLY GROUND VELOCITY * DIRECTION COSINE               ENER2090
C           OMIT ALL DOF THAT ARE CONSTRAINED TO A MASTER DOF ...     ENER2100
          IF (GVT(1).NE.0 .OR. GVT(2).NE.0 .OR. GVT(3) NE.0) THEN     ENER2110
            DO 20 NODE=1,NNODE                                        ENER2120
              ICOS=JTCOS(NODE)                                        ENER2130
              DO 20 K=1,3                                             ENER2140
              IF ( NOT. BTEST(JTFLG(NODE),K+11)) THEN                 ENER2150
                II=IDOF(NODE,K)                                       ENER2160
                IF (II.LE.L4)                                         ENER2170
     &           WORK2(II)=WORK2(II)        + GVT(1)*COSINE(1,K,ICOS) ENER2180
     &           + GVT(2)*COSINE(2,K,ICOS) + GVT(3)*COSINE(3,K,ICOS)  ENER2190
              ENDIF                                                   ENER2200
   20         CONTINUE                                                ENER2210
          ENDIF                                                       ENER2220
C                                                                     ENER2230
C ....  CALCULATE THE TOTAL KE                                        ENER2240
        DEKE=EKE                                                      ENER2250
        CALL ETMPLY(PT2,L1,L2,L3,L4,L41,L4,                           ENER2260
     &              IMDMS,MDMS,MASS,WORK,WORK2,WORK2,EKE)             ENER2270
        EKE=EKE/2.0                                                   ENER2280
C                                                                     ENER2290
C..... CALCULATE THE INCRIMENTAL KE                                   ENER2300
        DEKE=EKE-DEKE                                                 ENER2310
      ELSE                                                            ENER2320
C                                                                     ENER2330
C..... ZERO KE FOR STATIC CASE.                                       ENER2340
        DEKE=0                                                        ENER2350
        EKE =0                                                        ENER2360
      ENDIF                                                           ENER2370
C                                                                     ENER2380
C-----------------------------------                                ENER2390
C---  CALCULATE DAMPINED ENERGY   ---                               ENER2400
C-----------------------------------                                ENER2410
      IF (DYN) THEN                                                   ENER2420
C                                                                     ENER2430
C  . . . CALCULATE THE INCRIMENTAL DAMPING ENERGY                     ENER2440
        DEDD =0                                                       ENER2450
        IF (ALPHA.NE.0  OR. BETA.NE.0) THEN                           ENER2460
          DO 65 I=L3,L4                                               ENER2470
            FORCEI=-FDAMP(I)+DFDAMP(I)/2                              ENER2480
            VMI = VEL(I)+ DVEL(I)/2                                   ENER2490
            ENER=FORCEI*VMI                                           ENER2500
            IF (PT2) WRITE(108,*) I,                                  ENER2510
     &           'FORCEI',FORCEI,'  VM',VMI,'  ENER',ENER             ENER2520
   65       DEDD=DEDD + ENER                                          ENER2530
            DEDD=DEDD*DT                                              ENER2540
        ENDIF                                                         ENER2550
C                                                                     ENER2560
C. ....  CALCULATE THE TOTAL DAMPINED ENERGY                          ENER2570
        EDD=EDD+DEDD                                                  ENER2580
      ELSE                                                            ENER2590
C                                                                     ENER2600
C.....  ZERO DE FOR STATIC CASE.                                      ENER2610
        EDD=0                                                         ENER2620
        DEDD=0                                                        ENER2630
      ENDIF                                                           ENER2640
C                                                                     ENER2650
C-----------------------------                                       ENER2660
C---  CHECK ENERGY BALANCE  ---                                     ENER2670
C-----------------------------                                       ENER2680
      DIFF=EIE-ESE-EKE-PSE-EDD                                        ENER2690
      IF (EIE.NE.0) THEN                                              ENER2700
        RE=DIFF/EIE * 100                                             ENER2710
      ELSE                                                            ENER2720
        RE=100.00                                                     ENER2730
      ENDIF                                                           ENER2740
C                                                                     ENER2750
C-----------------                                                    ENER2760
C---  PRINT DATA ---                                                ENER2770
C-----------------                                                    ENER2780
      IF (WRIT0) WRITE(108,100) DEIE,EIE,DESE,ESE,DEKE,EKE,           ENER2790
     &                          DPSE,PSE,DEDD,EDD,RE                  ENER2800
  100 FORMAT (/,T10,' ENERGY BALANCE'/T10,'----------------',         ENER2810
     & T40,'INCRIMENTAL',T58,'TOTAL'/                                 ENER2820
     & T15,'INPUT ENERGY'          ,T40,1P,G12.5,T55,G12.5/           ENER2830
     & T15,'ELASTIC STRAIN ENERGY' ,T40,1P,G12.5,T55,G12.5/           ENER2840
     & T15,'KINETIC ENERGY '       ,T40,1P,G12.5,T55,G12.5/           ENER2850
     & T15,'PLASTIC STRAIN ENERGY' ,T40,1P,G12.5,T55,G12.5/           ENER2860
     & T15,'DAMPING ENERGY'        ,T40,1P,G12.5,T55,G12.5/           ENER2870
     & T15,'RELATIVE ER','ROR'                   ,T55,G12.5,'%')      ENER2880
      IF (DYN.AND.WRIT0) WRITE(108,110) ( GD(I), GV(I), GA(I),I=1,3), ENER2890
     &                                  (GDT(I),GVT(I),GAT(I),I=1,3) ENER2900
  110 FORMAT (/,T10,' GROUND RESPONSE' /T10,'---------------'/        ENER2910
     &       T10,'INCRIMENTAL VALUES',T46,'DISPLACEMENT',           ENER2920
     &       T61,'VELOCITY',T76,'ACCELERATION'/                     ENER2930
     &       T15,'GLOBAL X AXIS TRANSLATION...',T45,1P,3G15.5,/     ENER2940
     &       T15,'GLOBAL Y AXIS TRANSLATION . ',T45,1P,3G15.5,/     ENER2950
     &       T15,'GLOBAL Z AXIS TRANSLATION.. ',T45,1P,3G15.5,/     ENER2960
     &       T10,'TOTAL VALUES'/                                    ENER2970
     &       T15,'GLOBAL X AXIS TRANSLATION. ..',T45,1P,3G15 5,/    ENER2980
     &       T15,'GLOBAL Y AXIS TRANSLATION...',T45,1P,3G15.5,/     ENER2990
     &       T15,'GLOBAL Z AXIS TRANSLATION...',T45,1P,3G15 5)      ENER3000
C                                                                     ENER3010
C-------------------------------------                              ENER3020
C---  DUCTILITIES AND EXCURSION RATIOS ---                          ENER3030
C-------------------------------------                              ENER3040
        LSTTYP=0                                                      ENER3050
      DO 200 IELNO=1,NELMT                                            ENER3060
  200   CALL ELELIB                                                   ENER3070
     &  (11,LSTTYP, FALSE.,IREL,.FALSE.,NAME,EESE,EPSE,DAMAGE,DUCFAG, ENER3080
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDISP,IZLOAD,MSTOR)  ENER3090
C                                                                     ENER3100
      RETURN                                                          ENER3110
      END                                                             ENER3120
C                                                                     SKYL0010
C*********************************************************************SKYL0020
      SUBROUTINE SKYLN2(NDOF,L3,L4,MDS,MD,MDMASS,MDKG,KGFORM)         SKYL0030
      DIMENSION MDMASS(NDOF+1),MD(NDOF+1),MDS(NDOF+1),MDKG(NDOF+1)   SKYL0040
C                                                                     SKYL0050
C                                                                     SKYL0060
C------- DEVELOP SKYLINE OF DYNAMIC STIFFNESS (S)                    SKYL0070
C.....  MDS(I)   - ADDRESS OF DYNAMIC STIFF. MAIN DIAGONAL TERM ROW I SKYL0080
C ...   MD(I)    - ADDRESS OF STIFFNESS MAIN DIAGONAL TERM ROW I      SKYL0090
C. ....  MDMAS(I) - ADDRESS OF MASS MAIN DIAGONAL TERM ROW I          SKYL0100
C.......MDKG(I)  - ADDRESS OF GEOMETRIC STIFF  MAIN DIAGONAL TERM ROW ISKYL0110
C                                                                     SKYL0120
C. ... . LSTIFF - # OF STIFFNESS TERMS IN COLUMN J                    SKYL0130
C. .... LMASS  - # OF MASS TERMS IN COLUMN J                          SKYL0140
C........ LKG   - # OF KG TERMS IN COLUMN J                           SKYL0150
C....... L     - # OF B  TERMS IN COLUMN J                           SKYL0160
C                                                                     SKYL0170
      NFREE=L4-L3+1                                                   SKYL0180
      MDS(1)=1                                                        SKYL0190
      LKG=0                                                           SKYL0200
      DO 110 J1=2,NFREE+1                                             SKYL0210
        J=J1+L3-1                                                     SKYL0220
        LSTIFF=MD(J)-MD(J-1)                                          SKYL0230
        LMASS =MDMASS(J)-MDMASS(J-1)                                  SKYL0240
        IF (KGFORM.EQ.2)  LKG = MDKG(J)-MDKG(J-1)                     SKYL0250
        L=MIN(J1-1,MAX(LSTIFF,LMASS,LKG))                             SKYL0260
  110   MDS(J1)=MDS(J1-1)+L                                           SKYL0270
C                                                                     SKYL0280
      RETURN                                                          SKYL0290
      END                                                             SKYL0300
C                                                                     SMAX0010
C*********************************************************************SMAX0020
C--- SUBROUTINE SMAX,  FINDS THE MAXIMUN AND MINIMUN VALUES           SMAX0030
C                              (AFTER LANL)                           SMAX0040
      SUBROUTINE SMAX(N,DT,A,YMX,YMN,T1,T2,NLEN,AVE,STDEV,RMS)        SMAX0050
      DIMENSION A(NLEN)                                               SMAX0060
          AVE=0                                                       SMAX0070
          STDEV=0                                                     SMAX0080
          RMS=0                                                       SMAX0090
          YMX=A(1)                                                    SMAX0100
          YMN=A(1)                                                    SMAX0110
          T=DT                                                        SMAX0120
          T1=DT                                                       SMAX0130
```

```
          T2=DT                                                          SMAX0140
          IF (N LE.1) RETURN                                             SMAX0150
           SUMX=A(1)                                                     SMAX0160
           ADBL=A(1)                                                     SMAX0170
           SUMXSQ=ADBL**2                                                SMAX0180
        DO 40 I=2,N                                                      SMAX0190
         ADBL=A(I)                                                       SMAX0200
         SUMX   = SUMX   - ADBL                                          SMAX0210
         SUMXSQ = SUMXSQ - ADBL**2                                       SMAX0220
           IF (YMX LT.A(I)) THEN                                         SMAX0230
               YMX=A(I)                                                  SMAX0240
               T1=T                                                      SMAX0250
           ELSE IF (YMN.GT.A(I)) THEN                                    SMAX0260
               YMN=A(I)                                                  SMAX0270
               T2=T                                                      SMAX0280
           ENDIF                                                         SMAX0290
        T=T+DT                                                           SMAX0300
   40   CONTINUE                                                         SMAX0310
        AVE=SUMX/N                                                       SMAX0320
        IF ((N*SUMXSQ-SUMX**2).GT.0 )                                    SMAX0330
       &    STDEV=SQRT((N*SUMXSQ-SUMX**2)/(N*(N-1)))                    SMAX0340
        RMS=SQRT(SUMXSQ/N)                                               SMAX0350
        RETURN                                                           SMAX0360
        END                                                              SMAX0370
C-----------------------------------------------------------------------SOL00010
      SUBROUTINE SOL02A(IMD ,IMDMS     ,IMDKG,IMDS,IMDMSL,              SOL00020
     &  NNODE    , NCOS    , NELMT    , IBUG     , MAXNOD   ,           SOL00030
     &  TITLE    , STEPID  , NAINP    , NDOF     , NFREE    ,           SOL00040
     &  KGFORM   , KGLOAD  , KGTYPE   , NCOND    , INTEGO   ,           SOL00050
     &  NEWKG    , MCOND   , KGCOND   , UNBAL    , NEWK     ,           SOL00060
     &  THETA    , ELSTIC  , IPRNT    , IWRITE   , GAT      ,           SOL00070
     &  MAXACC   , SLMASS  , TO       , DT       , TP       , GVT    , SOL00080
     &  GRAV     , FDAMP   , IZDDSP   , IZDLOA   , GDT      ,           SOL00090
     &  DISP     , VEL     , ACC      , LOAD     , AN       , DSPMAX,  SOL00100
     &  DDISP    , DVEL    , DACC     , DLOAD    , BN       , VELMAX,  SOL00110
     &  STIFF    , MASS    , MASSSL   , KG       , SBAR     , ACCMAX,  SOL00120
     &  MD       , MDMS    , MDMSSL   , MDKG     , MDS      , RCTMAX,  SOL00130
     &  DAMP     , PBAR    , MCOS     , AGTX     , DLOAD2   ,           SOL00140
     &  AGTY     , AGTZ    , AINP     , FUB      , WORK     , SUMRC ,  SOL00150
     &  ID       , IDOF    , CNST     , JTFLG    , COORD    , TFDAMP,  SOL00160
     &  COSINE   , JTCOS   , DFDAMP   ,EIE,ESE,PSE,EKE,EDD,            SOL00170
     &   ZERO    , PG      , DPG      )                                 SOL00180
C                                                                        SOL00190
      COMMON /ISPEC/ISP1,SGD(500,2),REACF(6)                             SOL00200
      COMMON /IDSTEP/ ISTEP                                              SOL00210
      CHARACTER*(*) TITLE(2),STEPID                                      SOL00220
      CHARACTER NAME*80                                                  SOL00230
      LOGICAL PRINT,HEAD,REPEAT,SLMASS,AXIAL,NEWKDY,BTEST                SOL00240
      LOGICAL NEWKG,NEWK,ELSTIC,MCOND,KGCOND,UNBAL,MCND2,KGCND2,WRITD    SOL00250
      LOGICAL BUGS,BUG6,BUG11,DUCLOG                                     SOL00260
      REAL LOAD,MASS,MASSSL,KG,MCOS                                      SOL00270
      INTEGER FDAMP                                                      SOL00280
      DIMENSION ZERO(NDOF),DLOAD2(NDOF),PG(NDOF),DPG(NDOF),KG(IMDKG)     SOL00290
      DIMENSION SUMRC(6),DSUMRC(6),SUMFD(6),DSUMFD(6),SUMUB(6)           SOL00300
      DIMENSION TFDAMP(NDOF),DFDAMP(NDOF),RINPUT(100),MASSSL(IMDMSL)     SOL00310
      DIMENSION DSPMAX(NDOF),VELMAX(NDOF),ACCMAX(NDOF),RCTMAX(NDOF+6)    SOL00320
      DIMENSION GD(3),GV(3),GA(3),GDT(3),GVT(3),GAT(3),TSUM(6)           SOL00330
      DIMENSION STIFF(IMD),MASS(IMDMS),SBAR(IMDS),MDMSSL(NDOF+1)         SOL00340
      DIMENSION MD(NDOF+1),MDMS(NDOF+1),MDKG(NDOF+1),MDS(NDOF+1)         SOL00350
      DIMENSION LOAD(NDOF),DISP(NDOF),VEL(NDOF),ACC(NDOF),AN(NDOF)       SOL00360
      DIMENSION DLOAD(NDOF),DDISP(NDOF),DVEL(NDOF),DACC(NDOF),BN(NDOF)   SOL00370
      DIMENSION FUB(NDOF),PBAR(NDOF),MCOS(NDOF,3),DAMP(3),WORK(2*NDOF)   SOL00380
      DIMENSION AGTX(MAXACC),AGTY(MAXACC),AGTZ(MAXACC),AINP(NAINP)       SOL00390
      DIMENSION COORD(MAXNOD,3),ID(MAXNOD),JTFLG(MAXNOD),JTCOS(MAXNOD),  SOL00400
     &          IDOF(MAXNOD,6) ,COSINE(3,3,NCOS),CNST(MAXNOD,6)          SOL00410
      DO 5 I=1,5                                                         SOL00420
5     SUMUB(I)=0.                                                        SOL00430
      SUMFCR=0                                                           SOL00440
      SUMMCM=0                                                           SOL00450
      SUMMFO=0                                                           SOL00460
      SUMMMO=0                                                           SOL00470
      DUCLOG=  FALSE.                                                    SOL00480
      DUCFAG= -1                                                         SOL00490
C                                                                        SOL00500
      L1=1                                                               SOL00510
      L2=NCOND                                                           SOL00520
      L3=L2+1                                                            SOL00530
      L4=NCOND *NFREE                                                    SOL00540
      L5=L4+1                                                            SOL00550
      L6=NDOF                                                            SOL00560
C------------------------------------------                              SOL00570
      BUG5 =BTEST(IBUG, 5)                                               SOL00580
      BUG6 =BTEST(IBUG, 6)                                               SOL00590
      BUG11=BTEST(IBUG,11)                                               SOL00600
C------------------------------------------                              SOL00610
      IF (BUG5) CALL LMATRX(MASS,MDMS,NDOF,IMDMS,'MASS MATRIX DUMP-7')   SOL00620
C                                                                        SOL00630
C---------------------------------------                                 SOL00640
C- INITIALIZE GROUND ACCELERATION CONSTANTS ---                          SOL00650
C---------------------------------------                                 SOL00660
      DO 91 I=1,3                                                        SOL00670
        GD(I)=0                                                          SOL00680
        GV(I)=0                                                          SOL00690
        GA(I)=0                                                          SOL00700
        GDT(I)=0                                                         SOL00710
        GVT(I)=0                                                         SOL00720
   91   GAT(I)=0                                                        SOL00730
      ACCOLD=0                                                           SOL00740
      XGACC=0                                                            SOL00750
      YGACC=0                                                            SOL00760
      ZGACC=0                                                            SOL00770
C                                                                        SOL00780
C-----------------------------------                                     SOL00790
C- INITIALIZE ENERGY CONSTANTS ---                                       SOL00800
C-----------------------------------                                     SOL00810
      PSE=0                                                              SOL00820
      EKE=0                                                              SOL00830
      EDD=0                                                              SOL00840
      EIEM =0                                                            SOL00850
      ESEM =0                                                            SOL00860
      PSEM =0                                                            SOL00870
      EKEM =0                                                            SOL00880
      EDDM =0                                                            SOL00890
      DO 92 I=1,6                                                        SOL00900
92    SUMFD(I)=0                                                         SOL00910
C                                                                        SOL00920
C-----------------------------------                                     SOL00930
C- SET GEOMETRIC STIFFNESS FLAGS ---                                     SOL00940
C-----------------------------------                                     SOL00950
      NEWKG=.TRUE.                                                       SOL00960
      IF (KGLOAD EQ.0 .OR  KGTYPE.EQ.0 .OR. KGFORM.EQ.0) NEWKG= FALSE.   SOL00970
C                                                                        SOL00980
C------------------------                                                SOL00990
C- SET STIFFNESS FLAGS ---                                               SOL01000
C------------------------                                                SOL01010
      NEWK =  TRUE                                                       SOL01020
      KSAME= 0                                                           SOL01030
C                                                                        SOL01040
C       --- INPUT INITIAL VALUES FOR DISPL, VELOC, ACCEL                 SOL01050
      CALL INTDVA(NDOF,MAXNOD,NNODE,TITLE,IDOF,                          SOL01060
     &            ID,JTFLG,DISP  ,VEL   ,ACC   ,LOAD  )                SOL01070
C                                                                        SOL01080
      IF( ISP1 .EQ. 1 ) THEN                                             SOL01090
          EIE=ESE                                                        SOL01100
      ELSE IF( ISP1  EQ  0 ) THEN                                        SOL01110
          ESE=0                                                          SOL01120
          EIE=0                                                          SOL01130
          DO 96 I=1,6                                                    SOL01140
   96     REACF(I)=0                                                     SOL01150
      ENDIF                                                              SOL01160
C                                                                        SOL01170
C---- INITILIZE TIME TO CALC INITIAL GROUND ACCELERATION                 SOL01180
      T=TO                                                               SOL01190
      TL=T-DT                                                            SOL01200
C                                                                        SOL01210
C---  READ AND INITILIZE DYNAMIC GROUND ACCELERATIONS                   SOL01220
      IF (SLMASS) THEN                                                   SOL01230
        CALL DYLOA(1,MAXACC,NDOF,NAINP,T,TL,                             SOL01240
     &           ACCELZ,TGACC,DTACC,GRAV,                                SOL01250
     &           MAXNOD,NNODE,COSINE,IDOF,JTFLG,JTCOS,NCOS,              SOL01260
     &           AGTX  ,AGTY  ,AGTZ  ,MASSSL,MDMSSL,IMDMSL,              SOL01270
     &           AINP  ,MCOS  ,DLOAD ,GA    ,GAT    )                   SOL01280
      ELSE                                                               SOL01290
        CALL DYLOA(1,MAXACC,NDOF,NAINP,T,TL,                             SOL01300
     &           ACCELZ,TGACC,DTACC,GRAV,                                SOL01310
     &           MAXNOD,NNODE,COSINE,IDOF,JTFLG,JTCOS,NCOS,              SOL01320
     &           AGTX  ,AGTY  ,AGTZ  ,MASS  ,MDMS  ,IMDMS,               SOL01330
     &           AINP  ,MCOS  ,DLOAD ,GA    ,GAT    )                   SOL01340
      ENDIF                                                              SOL01350
C                                                                        SOL01360
C---  GET INITIAL DYNAMIC LOAD                                           SOL01370
      IF (SLMASS) THEN                                                   SOL01380
        CALL DYLOA(2,MAXACC,NDOF,NAINP,T,TL,                             SOL01390
     &           ACCELZ,TGACC,DTACC,GRAV,                                SOL01400
     &           MAXNOD,NNODE,COSINE,IDOF,JTFLG,JTCOS,NCOS,              SOL01410
     &           AGTX  ,AGTY  ,AGTZ  ,MASSSL,MDMSSL,IMDMSL,              SOL01420
     &           AINP  ,MCOS  ,LOAD ,GA    ,GAT    )                    SOL01430
      ELSE                                                               SOL01440
        CALL DYLOA(2,MAXACC,NDOF,NAINP,T,TL,                             SOL01450
     &           ACCELZ,TGACC,DTACC,GRAV,                                SOL01460
     &           MAXNOD,NNODE,COSINE,IDOF,JTFLG,JTCOS,NCOS,              SOL01470
     &           AGTX  ,AGTY  ,AGTZ  ,MASS  ,MDMS  ,IMDMS,               SOL01480
     &           AINP  ,MCOS  ,LOAD ,GA    ,GAT    )                    SOL01490
      ENDIF                                                              SOL01500
C                                                                        SOL01510
C---- INITILIZE INTEGRATION ROUTINES                                     SOL01520
C                                                                        SOL01530
      IF (INTEGO.EQ.1) THEN                                              SOL01540
        CALL LINACC(1,DT,1.00,NEWKDY,L2,L3,L4,                           SOL01550
     &      IMDMS,IMD,IMDS,IMDKG,NDOF,NFREE,KGLOAD,KGFORM,               SOL01560
     &      ACC  , VEL  , DACC  ,DVEL  , DDISP ,DLOAD2 ,                 SOL01570
     &      WORK ,MASS  ,MDMS   , DAMP  ,STIFF ,MD     ,                 SOL01580
     &      KG   ,MDKG  , PZ    ,SBAR  ,MDS   ,AN    , BN    , PBAR  , SOL01590
     &      FDAMP,ALPHA,BETA)                                            SOL01600
      ELSE IF (INTEGO.EQ.2) THEN                                         SOL01610
        CALL AVEACC(1,DT,THETA,NEWKDY,L2,L3,L4,                          SOL01620
     &      IMDMS,IMD,IMDS,IMDKG,NDOF,NFREE,KGLOAD,KGFORM,               SOL01630
     &      ACC  , VEL  ,DACC   , DVEL  , DDISP , DLOAD2,                SOL01640
     &      WORK ,MASS  ,MDMS   ,DAMP  ,STIFF , MD     ,                 SOL01650
     &      KG   ,MDKG  , PZ    ,SBAR  ,MDS   ,AN    , BN    , PBAR  , SOL01660
     &      FDAMP,ALPHA,BETA)                                            SOL01670
      ELSE IF (INTEGO.EQ 3) THEN                                         SOL01680
        CALL WILSON(1,DT,THETA,XG,MDKG, PZ,                              SOL01690
     &              L2,L3,L4,IMDMS,IMD,IMDS,IMDKG,NDOF,NFREE,            SOL01700
     &              XGLOAD,KGFORM,ACC, VEL  , DISP  , LOAD  ,            SOL01710
     &              DACC  , DVEL  , DDISP , DLOAD2,WORK,                 SOL01720
     &              MASS  ,MDMS   , DAMP  , STIFF ,MD     ,              SOL01730
     &              SBAR  ,MDS    , AN    , BN    , PBAR  ,FDAMP,        SOL01740
     &              ALPHA,BETA)                                          SOL01750
      ENDIF                                                              SOL01760
C                                                                        SOL01770
C---------------------------------------                                 SOL01780
C-WRITE INITIAL DATA TO OUTPUT FILE ---                                  SOL01790
C---------------------------------------                                 SOL01800
      IF (IWRITE GT.0) CALL DMPDAT(2,IWRITE,TO,DT)                       SOL01810
C                                                                        SOL01820
C------------------------------------------------------------------------SOL01830
C- LOOP FOR EACH TIME STEP , ISTEP=CURRENT STEP # ----------------------SOL01840
C------------------------------------------------------------------------SOL01850
C. . NSTEP = NUMBER OF TIME STEPS                                        SOL01860
C .. ISTEP = TIME STEP NUMBER                                            SOL01870
C... T     = CURRENT TIME                                                SOL01880
C... TL    = TIME AT PREVIOUS TIME STEP                                  SOL01890
C.. TO     = INITIAL TIME                                                SOL01900
C. . DT    = TIME INCRIMENT                                              SOL01910
C.  TL    = FINAL TIME                                                   SOL01920
C                                                                        SOL01930
      MCND2 = MCOND                                                      SOL01940
      NSTEP = (T2-TO)/DT                                                 SOL01950
      DO 500 ISTEP=1,NSTEP                                               SOL01960
        T =TO+ISTEP*DT                                                   SOL01970
        TL=T - DT                                                        SOL01980
        WRITE(STEPID,90) ISTEP,T                                         SOL01990
   90   FORMAT('STEP:',I6,',  TIME:',1P,G13.5)                          SOL02000
C                                                                        SOL02010
C---------------------------                                             SOL02020
C- CALC INCRIMENTAL LOADS  --                                            SOL02030
C---------------------------                                             SOL02040
      I2=2                                                               SOL02050
      CALL DYLOA(2,MAXACC,NDOF,NAINP,T,TL,                               SOL02060
     &         ACCELZ,TGACC,DTACC,GRAV,                                  SOL02070
     &         MAXNOD,NNODE,COSINE,IDOF,JTFLG,JTCOS,NCOS,                SOL02080
     &         AGTX  ,AGTY  ,AGTZ  ,MASS  ,MDMS   ,IMDMS,                SOL02090
     &         AINP  ,MCOS  ,DLOAD ,GA    ,GAT    )                     SOL02100
     PZ    =ACCELZ                                                       SOL02110
C                                                                        SOL02120
C---- ADD UNBALANCED LOAD'S TO THE LOAD VECTOR. ---> STORE AS DLOAD2     SOL02130
C     STORE OLD UNBALANCED LOADS AT REACTIONS IN DLOAD                   SOL02140
      IF (UNBAL) THEN                                                    SOL02150
        DO 195 I=L1,L4                                                   SOL02160
          DLOAD2(I)=DLOAD(I)+FUB(I)                                      SOL02170
  195     FUB(I)=0.0                                                     SOL02180
        DO 196 I=L5,L6                                                   SOL02190
          DLOAD2(I)=-FUB(I)                                              SOL02200
C***                                                                     SOL02210
C***   OMIT UNBALANCED LOAD FROM REACTIONS...                            SOL02220
C***   UNBALANCED LOADS ADDED TO THE FRAME ARE RESISTED BY               SOL02230
C***   1) INERTIAL FORCES                                                SOL02240
C***   2) DAMPING FORCES                                                 SOL02250
C***   3) MEMBER FORCES                                                  SOL02260
C***   SUBTRACTING UNBALANCED FORCES FROM REACTIONS DUE TO MEMBER FORCES SOL02270
C***   NEGLECTS THE REACTIONS DUE TO INERTIAL AND DAMPING FORCES          SOL02280
C***                                                                     SOL02290
  196   FUB(I)=0.0                                                       SOL02300
      ELSE                                                               SOL02310
        DO 197 I=L1,L6                                                   SOL02320
  197     DLOAD2(I)=DLOAD(I)                                             SOL02330
      ENDIF                                                              SOL02340
C                                                                        SOL02350
C--------------------------                                              SOL02360
C- FORMULATE STIFFNESS ----                                              SOL02370
C--------------------------                                              SOL02380
C  FORMULATE THE STIFFNESS IF THE FLAG NEWK IS TRUE, THIS FLAG IS        SOL02390
C       INITIALIZED ABOVE, AND SET WHEN THE ELEMENT FORCES ARE CALC.     SOL02400
C       THE STIFFNESS IS ALSO FORMULATED IF KG IS TO BE CALCULATED       SOL02410
C       AND CONDENSATION IS TO BE PREFORMED, (THE STIFFNESS IS NEEDED    SOL02420
C       FOR THE CONDENSATION PROCEDURE...)                               SOL02430
      REPEAT=.TRUE.                                                      SOL02440
      KSAME=0                                                            SOL02450
  100 IF (NEWK) THEN                                                     SOL02460
        I1=1                                                             SOL02470
        CALL FORM(I1)                                                    SOL02480
        IF (BUGS) CALL LMATRX                                            SOL02490
     &              (STIFF,MD,NDOF,IMD,'GLOBAL STIFFNESS DUMP-7')       SOL02500
      ENDIF                                                              SOL02510
C                                                                        SOL02520
C-----------------------------                                           SOL02530
C- FORM GEOMETRIC STIFFNESS ---                                          SOL02540
C-----------------------------                                           SOL02550
C  FORMULATE THE GEOMETRIC STIFFNESS IF THE FLAG NEWK IS TRUE. THIS      SOL02560
C       FLAG INITIALIZED ABOVE                                           SOL02570
C       IF KGDATA(1)=1, THE GEOMETRIC STIFFNESS IS FORMED ONCE, AND      SOL02580
C                       MULTIPLIED BY A SCALER TO ACCOUNT FOR CHANGES    SOL02590
C                       IN THE TOTAL VERTICAL LOAD. THUS THE GEOMETRIC   SOL02600
C                       STIFFNESS ONLY NEEDS TO BE FORMED ONCE           SOL02610
C       IF KGDATA(1)=2, THE GEOMETRIC STIFFNESS IS BASED ON THE AXIAL    SOL02620
C                       FORCE IN THE ELEMENTS, AND IS RECALCULATED EACH  SOL02630
C                       TIME THE AXIAL FORCE CHANGES.                    SOL02640
C       IF KGDATA(3)=1, THE GEOMETRIC STIFFNESS IS FORMED WITH THE       SOL02650
C                       STIFFNESS, THUS A SEPERATE CALL TO FORM IS       SOL02660
C                       NOT NEEDED                                       SOL02670
C       IF KGDATA(3)=2, A SEPERATE GEOMETRIC STIFFNESS MATRIX IS FORMED  SOL02680
```

```
      IF (NEWKG) THEN                                                   SOL02690
C                                                                       SOL02700
C           FORMULATE SEPERATE GEOMETRIC STIFFNESS MATRIX               SOL02710
            IF (KGFORM.EQ 2) THEN                                       SOL02720
               PZ    =0                                                 SOL02730
               I2=2                                                     SOL02740
               CALL PORM(I2)                                            SOL02750
               KGCND2=KGCOND                                            SOL02760
               IF (BUG5) CALL LMATRX                                    SOL02770
     &                    (KG,MDKG,NDOF,IMDKG,'KG MATRIX DUMP-7')      SOL02780
            ELSE                                                        SOL02790
               KGCND2=.FALSE.                                           SOL02800
            ENDIF                                                       SOL02810
C                                                                       SOL02820
      ENDIF                                                             SOL02830
C                                                                       SOL02840
C---------------------------                                            SOL02850
C- CONDENSE OUT FREE DISPL---                                           SOL02860
C---------------------------                                            SOL02870
C .. IF MCOND IS TRUE THE MASS MATRIX IS ALSO REDUCED BY GUYAN REDUCTIONSOL02880
C... IF KGCND2 IS TRUE THEN KG IS ALSO REDUCED                          SOL02890
      IF (NCOND.GT 0  AND. NEWK ) THEN                                  SOL02900
         CALL MSOLL(1,BUG5,L4,IMD,NDOF,1,L1,L2,MD,STIFF ,DLOAD2,WORK,   SOL02910
     &              MCND2,MASS,MDMS,IMDMS,KGCND2,KG,MDKG,IMDKG)         SOL02920
         MCND2 = FALSE                                                  SOL02930
         KGCND2= FALSE.                                                 SOL02940
      ELSE IF (NCOND.GT.0) THEN                                         SOL02950
C        REDUCE THE DYNAMIC LOAD MATRIX IF OLD STIFFNESS IS USED...     SOL02960
         CALL MSOLL(2,BUG5,L4,IMD,NDOF,1,L1,L2,MD,STIFF,DLOAD2,WORK,    SOL02970
     &              MCND2,MASS,MDMS,IMDMS,KGCND2,KG,MDKG,IMDKG)         SOL02980
      ENDIF                                                             SOL02990
C                                                                       SOL03000
C-------------------------------------------------------               SOL03010
C- DETERMINE IF A NEW DYNAMIC STIFFNESS NEEDS TO BE CALC ---            SOL03020
C-------------------------------------------------------               SOL03030
      IF (NEWK .OR. NEWXG .OR. (KGLOAD.EQ.1 .AND. ACCELZ.EQ.ACCOLD))THENSOL03040
         NEWKDY= TRUE.                                                  SOL03050
         ACCOLD=ACCELZ                                                  SOL03060
      ELSE                                                              SOL03070
         NEWKDY= FALSE.                                                 SOL03080
      ENDIF                                                             SOL03090
C                                                                       SOL03100
C-------------------------------------------                            SOL03110
C- SOLVE DYNAMIC EQUATION FOR INCRIMENTAL DISPL **                      SOL03120
C-------------------------------------------                            SOL03130
      I2=2                                                              SOL03140
      IF (INTEGO.EQ 1) THEN                                             SOL03150
       CALL LINACC(I2,DT,1.00,NEWKDY,L2,L3,L4,                          SOL03160
     &      IMDMS,IMD,IMDS,IMDKG,NDOF,NFREE,KGLOAD,KGFORM,              SOL03170
     &      ACC  , VEL   , DACC  , DVEL  , DDISP ,DLOAD2,               SOL03180
     &      WORK ,MASS   ,MDMS   , DAMP  ,STIFF , MD    ,               SOL03190
     &      KG   ,MDKG   , PZ    , SBAR  ,MDS   , AN    , BN    ,PBAR  , SOL03200
     &      FDAMP,ALPHA,BETA)                                           SOL03210
      ELSE IF (INTEGO.EQ 2) THEN                                        SOL03220
       CALL AVEACC(I2,DT,THETA,NEWKDY,L2,L3,L4,                         SOL03230
     &      IMDMS,IMD,IMDS,IMDKG,NDOF,NFREE,KGLOAD,KGFORM,              SOL03240
     &      ACC  ,VEL   ,DACC   , DVEL  , DDISP , DLOAD2,               SOL03250
     &      WORK ,MASS  ,MDMS   ,DAMP  ,STIFF , MD    ,               SOL03260
     &      KG   ,MDKG  , PZ    ,SBAR  ,MDS   , AN    , BN    , PBAR  , SOL03270
     &      FDAMP,ALPHA,BETA)                                           SOL03280
      ELSE IF (INTEGO.EQ.3) THEN                                        SOL03290
       CALL WILSON(I2,DT,THETA,KG ,MDKG, PZ ,                           SOL03300
     &             L2,L3,L4,IMDMS,IMD,IMDS,IMDKG,NDOF,NFREE,          SOL03310
     &             KGLOAD,KGFORM,ACC, VEL   , DISP  , LOAD  ,         SOL03320
     &             DACC  , DVEL  , DDISP , DLOAD2,WORK,               SOL03330
     &             MASS  ,MDMS   , DAMP  , STIFF ,MD    ,             SOL03340
     &             SBAR  ,MDS    , AN    , BN    , PBAR  ,FDAMP,       SOL03350
     &             ALPHA,BETA)                                        SOL03360
      ENDIF                                                             SOL03370
C                                                                       SOL03380
C-------------------------------------------------                      SOL03390
C- BACK SUBSTIUTION TO SOLVE FOR NON-DYNAMIC DOF ---                    SOL03400
C-------------------------------------------------                      SOL03410
C  DISP_C = T * DISP_F                                                  SOL03420
C  DISP_C = INV(K_CC) * (-K_CF)*DISP(F)                                 SOL03430
C  SUBRTN REA_CTN (IOPT=1):                                             SOL03440
C     PREFORMS THE MULTIPICATION DISP_C=(-K_CF)*DISP_F                  SOL03450
C  SUBRTN MSOLL  (IOPT=3):                                              SOL03460
C     PREFORMS THE BACK SUBSTITUTION TO EQUILIVENT TO                   SOL03470
C     DISP_C=INV(K_CC)*DISP_C                                           SOL03480
C                                                                       SOL03490
      ONE=1.00                                                          SOL03500
      IF (NCOND.GT.0) THEN                                              SOL03510
            I3=3                                                        SOL03520
            I1=1                                                        SOL03530
C---------- ------- DISPLACEMENTS ------------------------------------ SOL03540
         IF (BUG5) WRITE(108,*) '*---- CALC DISPL'                      SOL03550
         CALL REACTN(I1,BUG5,NNODE,L1,L2,L3,L4,DDISP,DDISP,             SOL03560
     &                  MD    ,STIFF ,MD(NDOF),NDOF,1,ONE,             SOL03570
     &                  MAXNOD,IDOF,COORD ,ID,TITLE,STEPID, FALSE.,     SOL03580
     &                  NCOS,COSINE,JTCOS,JTFLG,DSUMRC)                 SOL03590
         CALL MSOLL(I3,BUG5,L4,IMD,NDOF,I1,L1,L3,MD,STIFF,DDISP,        SOL03600
     &              WORK, FALSE.,MASS,MDMS,1,.FALSE.,KG,MDKG,1)         SOL03610
C------------------ VELOCITIES ------------------------------------     SOL03620
         IF (BUG5) WRITE(108,*) '*---- CALC VELOC'                      SOL03630
           CALL REACTN(I1,BUG5,NNODE,L1,L2,L3,L4,DVEL,DVEL,             SOL03640
     &                  MD    ,STIFF ,MD(NDOF),NDOF,1,ONE,             SOL03650
     &                  MAXNOD,IDOF,COORD ,ID,TITLE,STEPID, FALSE.,     SOL03660
     &                  NCOS,COSINE,JTCOS,JTFLG,DSUMRC)                 SOL03670
         CALL MSOLL(I3,BUG5,L4,IMD,NDOF,I1,L1,L3,MD,STIFF,DVEL,         SOL03680
     &              WORK, FALSE.,MASS,MDMS,1,.FALSE.,KG,MDKG,1)         SOL03690
C--------- -- ----- ACCELERATION ------------------------------------- SOL03700
         IF (BUG5) WRITE(108,*) '*---- CALC ACCEL'                     SOL03710
         CALL REACTN(I1,BUG5,NNODE,L1,L2,L3,L4,DACC,DACC,               SOL03720
     &                  MD    ,STIFF ,MD(NDOF),NDOF,1,ONE,             SOL03730
     &                  MAXNOD,IDOF,COORD ,ID,TITLE,STEPID, FALSE.,     SOL03740
     &                  NCOS,COSINE,JTCOS,JTFLG,DSUMRC)                 SOL03750
         CALL MSOLL(I3,BUG5,L4,IMD,NDOF,I1,L1,L3,MD,STIFF,DACC,         SOL03760
     &              WORK, FALSE.,MASS,MDMS,1,.FALSE.,KG,MDKG,1)         SOL03770
      ENDIF                                                             SOL03780
C                                                                       SOL03790
C-------------------------------------                                  SOL03800
C- CALCULATE INCRIMENTAL MEMBER FORCES ---                              SOL03810
C-------------------------------------                                  SOL03820
C ELELIB CALLS THE INDIVIDUAL ELEMENT LIBRARIES, EACH OF WHICH DETERMINSSOL03830
C  1) DID THE STIFFNESS CHANGE?  IF SO; NEWK= TRUE.                     SOL03840
C  2) SHOULD THE INCRIMENTAL DISPLACEMENT BE RECALCULATED?              SOL03850
C     IF SO, KSAME=KSAME+___ (A VALUE DEPENDING ON ELEMENT TYPE)        SOL03860
C                                                                       SOL03870
      LSTTYP=0                                                          SOL03880
      PRINT= FALSE                                                      SOL03890
      HEAD = FALSE.                                                     SOL03900
      NEWK=.FALSE                                                       SOL03910
      NELMT=NELMT                                                       SOL03920
      DO 150 IELNO=1,NELMT                                              SOL03930
  150   CALL ELELIB                                                     SOL03940
     &  (4,LSTTYP ,PRINT ,IREL,HEAD,NAME,EESE,EPSE,DAMAGE,DUCFAG,       SOL03950
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDDSP,IZDLOA,MSTOR)    SOL03960
C                                                                       SOL03970
C-----------------------------------------                              SOL03980
C- ZERO STIFFNESS CHANGE FLAGS IF ELASTIC ---                           SOL03990
C-----------------------------------------                              SOL04000
      IF (ELSTIC) THEN                                                  SOL04010
         NEWK=.FALSE.                                                   SOL04020
         KSAME=0                                                        SOL04030
      ENDIF                                                             SOL04040
C                                                                       SOL04050
C-----------------------------------------------                        SOL04060
C- DETERMINE IF KG IS TO BE UPDATED BASED ON  ---                       SOL04070
C-    CHANGING ELEMENT AXIAL FORCE            ---                       SOL04080
C-----------------------------------------------                        SOL04090
            IF (KGLOAD EQ 2) THEN                                       SOL04100
               NEWKG=.TRUE.                                             SOL04110
               IF (NCOND.GT.0) NEWK=.TRUE.                              SOL04120
            ELSE                                                        SOL04130
               NEWKG= FALSE.                                            SOL04140
            ENDIF                                                       SOL04150
C                                                                       SOL04160
C---------------------------                                            SOL04170
C- REPEAT STEP IF NESSARY ---                                           SOL04180
C---------------------------                                            SOL04190
C  IF AN ELEMENT STIFFNESS CHANGE DICTATED THAT THE STEP BE REANALIZED, SOL04200
C     REFORMULATE THE STIFFNESS AND REANALIZE                           SOL04210
C     THE VARIABLE REPEAT IS SET UP TO SUPRESS THE REANALIZING OF A STEPSOL04220
C     CURENTLY, EACH STEP MAY ONLY BE REANALIZED ONCE, THIS PREVENTS    SOL04230
C     AN ENDLESS LOOP.                                                  SOL04240
C                                                                       SOL04250
      IF (KSAME.NE.0  AND. REPEAT) THEN                                 SOL04260
          REPEAT= FALSE.                                                SOL04270
          GO TO 100                                                     SOL04280
      ENDIF                                                             SOL04290
C                                                                       SOL04300
C-------------------------                                              SOL04310
C- SOLVE FOR REACTIONS ---                                              SOL04320
C-------------------------                                              SOL04330
C REACTIONS ARE DUE TO INTERNAL MEMBER FORCES                           SOL04340
C                                                                       SOL04350
      IF (BUG5) WRITE(108,*)'*---- CALC REACT-K*X'                      SOL04360
      CALL REACTN(2,BUG5,NNODE,L1,L4,L5,L6,DLOAD ,DDISP ,               SOL04370
     &            MD    ,STIFF ,MD(NDOF),NDOF,1,1.0,                  SOL04380
     &            MAXNOD,IDOF,COORD ,ID,TITLE,STEPID,.FALSE.,         SOL04390
     &            NCOS,COSINE,JTCOS,JTFLG,DSUMRC)                     SOL04400
C                                                                       SOL04410
C--- - GET SUMMATION OF REACTIONS                                       SOL04420
      CALL REACTN(3,BUG5,NNODE,L1,L4,L5,L6,DLOAD ,DDISP,                SOL04430
     &            MDKG   ,KG    ,MDKG(NDOF),NDOF,1,1.0,                SOL04440
     &            MAXNOD,IDOF,COORD ,ID,TITLE,STEPID,.FALSE.,         SOL04450
     &            NCOS,COSINE,JTCOS,JTFLG,DSUMRC)                     SOL04460
C                                                                       SOL04470
C------ GET SUMMATION OF REACTIONS DUE TO UNBALANCED LOAD               SOL04480
      IF (.NOT ELSTIC)                                                  SOL04490
     & CALL REACTN(3,BUG5,NNODE,L1,L4,L5,L6,DLOAD ,DDISP,               SOL04500
     &            MDKG   ,KG    ,MDKG(NDOF),NDOF,1,1.0,                SOL04510
     &            MAXNOD,IDOF,COORD ,ID,TITLE,STEPID, FALSE.,         SOL04520
     &            NCOS,COSINE,JTCOS,JTFLG,SUMUB)                      SOL04530
C                                                                       SOL04540
C-------------------------------------------------------               SOL04550
C- SOLVE FOR INCRIMENTAL GEOMETRIC STIFFNESS FORCE, DFG ---             SOL04560
C-------------------------------------------------------               SOL04570
C                                                                       SOL04580
C------- INCRIMENTAL GEOMETRIC STIFFNESS FORCE                          SOL04590
      IF (KGFORM.EQ.2) THEN                                             SOL04600
         IF (BUG5) WRITE(108,*) '*------ INCRIMENTAL FORCE DUE TO KG '  SOL04610
         CALL TMPLY(BUG5,.TRUE .,IMD,L4,L3,L4,L3,L4,                    SOL04620
     &              MDKG,KG,DFG,DDISP)                                  SOL04630
C        SCALE FORCE TO INLCUDE GROUND ACCELERATION . .                 SOL04640
         DO 159 I=L3,L4                                                 SOL04650
  159    DFG(I)= (1+PZ)*DFG(I)                                          SOL04660
      ENDIF                                                             SOL04670
C                                                                       SOL04680
C-------------------------------------------                            SOL04690
C- SOLVE FOR INCRIMENTAL DAMPING FORCE,  DFDAMP ---                     SOL04700
C-------------------------------------------                            SOL04710
C                                                                       SOL04720
      IF (BETA.NE.0) THEN                                               SOL04730
C------- INCRIMENTAL DAMPING FORCE DUE TO STIFFNESS  ..                 SOL04740
         IF (BUG5) WRITE(108,*)                                         SOL04750
     &      ' *---- DAMPING FORCE/REACT DUE TO STIFFNESS'               SOL04760
         CALL TMPLY(BUG5,.TRUE.,IMD,L6,L3,L4,L3,L4,                     SOL04770
     &              MD,STIFF,DFDAMP,DVEL)                               SOL04780
         CALL TMPLY(BUG5, TRUE.,IMD,L6,L5,L6,L1,L4,                     SOL04790
     &              MD,STIFF,DFDAMP,DVEL)                               SOL04800
C------- INCRIMENTAL DAMPING FORCE DUE TO KG  ...                       SOL04810
         IF (KGFORM.EQ.2) THEN                                          SOL04820
           IF (BUG5) WRITE(108,*)                                       SOL04830
     &      ' *---- DAMPING FORCE/REACT DUE TO KG '                     SOL04840
           CALL TMPLY(BUG5,.TRUE.,IMD,L6,L3,L4,L3,L4,                   SOL04850
     &              MDKG,KG,WORK,DVEL)                                  SOL04860
           CALL TMPLY(BUG5, TRUE.,IMD,L6,L5,L6,L1,L4,                   SOL04870
     &              MDKG,KG,WORK,DVEL)                                  SOL04880
           DO 158 I=L3,L6                                               SOL04890
  158        DFDAMP(I)=DFDAMP(I)+WORK(I)                                SOL04900
         ENDIF                                                          SOL04910
C                                                                       SOL04920
C------------ CALC SUM OF STIFFNESS PROPORTIONAL DAMPING REACTIONS      SOL04930
      CALL REACTN(3,BUG5,NNODE,L1,L4,L5,L6,DFDAMP,DVEL ,                SOL04940
     &            MDMS   ,MASS  ,MDMS(NDOF),NDOF,1.1 00,                SOL04950
     &            MAXNOD,IDOF,COORD ,ID,TITLE,STEPID,.FALSE.,         SOL04960
     &            NCOS,COSINE,JTCOS,JTFLG,DSUMFD)                     SOL04970
C                                                                       SOL04980
C------- MULTIPLY BY BETA                                               SOL04990
         DO 160 I=1,6                                                   SOL05000
           DSUMFD(I)=DSUMFD(I)*BETA                                     SOL05010
  160      IF (BUG5) WRITE(108,*) 'I,DSUMFD(I),BETA',I,DSUMFD(I),BETA  SOL05020
         DO 161 I=L3,L4                                                 SOL05030
  161    DFDAMP(I)=DFDAMP(I)*BETA                                       SOL05040
      ENDIF                                                             SOL05050
C                                                                       SOL05060
      IF (ALPHA.NE.0) THEN                                              SOL05070
C------- INCRIMENTAL DAMPING FORCE DUE TO MASS ...                      SOL05080
         IF (BUG5) WRITE(108,*) '*------ DAMPING FORCE DUE TO MASS '    SOL05090
         DO 162 I=L1,L6                                                 SOL05100
  162      WORK(I)=0.00                                                 SOL05110
         CALL TMPLY(BUG5,.FALSE.,IMD,L4,L3,L4,L3,L4,                    SOL05120
     &              MDMS,MASS,WORK,DVEL)                                SOL05130
C------------ CALC SUM OF FD                                           SOL05140
         CALL REACTN(3,BUG5,NNODE,L1,L2,L3,L4,WORK,DVEL ,               SOL05150
     &                MDMS   ,MASS  ,MDMS(NDOF),NDOF,1.1 00,            SOL05160
     &                MAXNOD,IDOF,COORD ,ID,TITLE,STEPID,.FALSE.,     SOL05170
     &                NCOS,COSINE,JTCOS,JTFLG,TSUM)                   SOL05180
C------- MULTIPLY BY ALPHA ADD TO PREVIOUS SUM . .                      SOL05190
         IF (BETA NE.0) THEN                                            SOL05200
           DO 165 I=L3,L4                                               SOL05210
  165        DFDAMP(I)=DFDAMP(I)+ALPHA*WORK(I)                          SOL05220
           DO 167 I=1,6                                                 SOL05230
  167        DSUMFD(I)=DSUMFD(I)+TSUM(I)*ALPHA                          SOL05240
         ELSE                                                           SOL05250
           DO 168 I=L3,L4                                               SOL05260
  168        DFDAMP(I)=ALPHA*WORK(I)                                    SOL05270
           DO 166 I=1,6                                                 SOL05280
  166        DSUMFD(I)=-TSUM(I)*ALPHA                                   SOL05290
         ENDIF                                                          SOL05300
         IF (BUG5) WRITE(108,*) 'DSUMFD--',DSUMFD                       SOL05310
      ENDIF                                                             SOL05320
      IF (BUG5) THEN                                                    SOL05330
      DO 169 I=L3,L6                                                    SOL05340
  169   WRITE(108,*) 'I',I,'DFDAMP',DFDAMP(I),'FDAMP',TFDAMP(I)         SOL05350
      ENDIF                                                             SOL05360
C                                                                       SOL05370
C---------------------------------------                                SOL05380
C- CALC AND ENERGIES AND DUCTILITIES  ---                               SOL05390
C---------------------------------------                                SOL05400
      WRIT0=  MOD(ISTEP,IPRNT).EQ.0                                     SOL05410
      HEAD = WRIT0                                                      SOL05420
C                                                                       SOL05430
      IF (HEAD) WRITE(108,250) TITLE(1),TITLE(2),' ',ISTEP,STEPID       SOL05440
      IF (SLMASS) THEN                                                  SOL05450
C---- ------ ENERGY FORMULATION WITH SPECIAL LOADING MASS - -           SOL05460
C           BASED ON APPLIED LOADING                                    SOL05470
       CALL ENERGY(L1,L2,L3,L4,L5,MAXNOD,NCOS,BUG11,                    SOL05480
     &             IMDMS,MDMS, TRUE.,WRITO,MASS,EIE,ESE,PSE,EKE,EDD,    SOL05490
     &             ALPHA,BETA,KGFORM,NELMT,NNODE,DT,SUMUB,SUMRC,        SOL05500
     &             DSUMRC,SUMFD,DSUMFD,TFDAMP,DFDAMP,ZERO,GV,GD,ZERO,   SOL05510
     &             GVT,GDT,WORK,WORK(L6+1),FG,DFG,DACC,DVEL,DDISP,ACC,  SOL05520
     &             VEL,DISP,LOAD,DLOAD,IDOF,JTFLG,COSINE,JTCOS)         SOL05530
      ELSE                                                              SOL05540
C------------ ENERGY FORMULATION WITH/O SPECIAL LOADING MASS ---         SOL05550
C           BASED ON GROUND MOTION                                      SOL05560
       CALL ENERGY(L1,L2,L3,L4,L5,MAXNOD,NCOS,BUG11,                    SOL05570
```

```
     &           IMDMS,MDMS,.TRUE.,WRITO,MASS,EIE,ESE,PSE,EKE,EDD,      SOL05610
     &           ALPHA,BETA,KGFORM,NELMT,NNODE,DT,SUMUB,SUMRC,          SOL05620
     &           DSUMRC,SUMFD,DSUMFD,TFDAMP,DFDAMP,GA,GV,GD,GAT,        SOL05630
     &           GVT,GDT,WORK,WORX(L6+1),PG,DPG,DACC,DVEL,DDISP,ACC,    SOL05640
     &           VEL,DISP,ZERO,ZERO,IDOF,JTFLG,COSINE,JTCOS)            SOL05650
      ENDIF                                                             SOL05660
      EIEM =AMAX1(SNGL(EIE),EIEM )                                      SOL05670
      ESEM =AMAX1(SNGL(ESE),ESEM )                                      SOL05680
      PSEM =AMAX1(SNGL(PSE),PSEM )                                      SOL05690
      EKEM =AMAX1(SNGL(EKE),EKEM )                                      SOL05700
      EDDM =AMAX1(SNGL(EDD),EDDM )                                      SOL05710
C                                                                       SOL05720
C*****************************************************                SOL05730
C- GET TOTAL GLOBAL LOAD, DISPL, VELOC AND ACCEL ---                    SOL05740
C*****************************************************                SOL05750
      DO 310 I=1,NDOF                                                   SOL05760
        ACC(I)   = DACC(I)   + ACC(I)                                   SOL05770
        VEL(I)   = DVEL(I)   + VEL(I)                                   SOL05780
        DISP(I)  = DDISP(I)  + DISP(I)                                  SOL05790
        LOAD(I)  = DLOAD(I)  + LOAD(I)                                  SOL05800
        TFDAMP(I)= DFDAMP(I) + TFDAMP(I)                                SOL05810
        PG(I)    = DPG(I)    + PG(I)                                    SOL05820
        IF (ABS( ACC(I)) GT.ABS(ACCMAX(I))) ACCMAX(I)= ACC(I)           SOL05830
        IF (ABS( VEL(I)).GT.ABS(VELMAX(I))) VELMAX(I)= VEL(I)           SOL05840
        IF (ABS(DISP(I)) GT.ABS(DSPMAX(I))) DSPMAX(I)=DISP(I)           SOL05850
        IF (ABS(LOAD(I)).GT.ABS(RCTMAX(I))) RCTMAX(I)=LOAD(I)           SOL05860
        DACC(I)=0.00                                                    SOL05870
        DVEL(I)=0.00                                                    SOL05880
        DDISP(I)=0.00                                                   SOL05890
        DFDAMP(I)=0.00                                                  SOL05900
  310   DLOAD(I) =0.00                                                  SOL05910
C                                                                       SOL05920
        DO 309 I=1,6                                                    SOL05930
          SUMRC(I) = DSUMRC(I) + SUMRC(I)                               SOL05940
  309     SUMFD(I) = DSUMFD(I) + SUMFD(I)                               SOL05950
        SUMFOR=SQRT(SUMRC(1)**2+SUMRC(2)**2+SUMRC(3)**2)                SOL05960
        SUMMOM=SQRT(SUMRC(4)**2+SUMRC(5)**2+SUMRC(6)**2)                SOL05970
        SUMMFO=MAX(SUMFOR,SUMMFO)                                       SOL05980
        SUMMMO=MAX(SUMMOM,SUMMMO)                                       SOL05990
C                                                                       SOL06000
C*****************************************************************     SOL06010
C- PRINT TOTAL GLOBAL DISPLACEMENTS, VELOCITIES, ACCELERATIONS ---      SOL06020
C*****************************************************************     SOL06030
      IF (WRITO) THEN                                                   SOL06040
           CALL JOIDSP(1,MAXNOD,NDOF,NNODE,1,L4,ID,IDOF,CNST,DISP,      SOL06050
     &                 TITLE,'DISPLACEMENTS',STEPID,.TRUE.,           SOL06060
     &                 NCOS,COSINE,JTCOS)                             SOL06070
           CALL JOIDSP(1,MAXNOD,NDOF,NNODE,1,L4,ID,IDOF,CNST,VEL,       SOL06080
     &                 TITLE,'VELOCITIES',STEPID,.TRUE.,              SOL06090
     &                 NCOS,COSINE,JTCOS)                             SOL06100
           CALL JOIDSP(1,MAXNOD,NDOF,NNODE,1,L4,ID,IDOF,CNST,ACC,       SOL06110
     &                 TITLE,'ACCELERATIONS',STEPID,.TRUE.,           SOL06120
     &                 NCOS,COSINE,JTCOS)                             SOL06130
      ENDIF                                                             SOL06140
C                                                                       SOL06150
C---------------------------------                                      SOL06160
C- PRINT TOTAL GLOBAL REACTIONS ---                                     SOL06170
C---------------------------------                                      SOL06180
      CALL REACTN(3,WRITO,NNODE,L1,L4,L5,L6,LOAD,ACC ,                  SOL06190
     &     MDMS   ,MASS  ,MDMS(NDOF),NDOF,1,ONE,                       SOL06200
     &     MAXNOD,IDOF,COORD ,ID,TITLE,STEPID,.TRUE.,                   SOL06210
     &     NCOS,COSINE,JTCOS,JTFLG,SUMRC)                             SOL06220
      IF (WRITO) WRITE(108,313) SUMFOR,SUMMOM                           SOL06230
  313 FORMAT (/4X,'RESULTANT OF REACTIONS,   FORCE=',1P,G12.4,          SOL06240
     &        '    MOMENT=',G12.4)                                      SOL06250
      DO 311 I=1,6                                                      SOL06260
         J=NDOF+I                                                       SOL06270
         IF (ABS(SUMRC(I)).GT.ABS(RCTMAX(J))) RCTMAX(J)=SUMRC(I)        SOL06280
  311    CONTINUE                                                       SOL06290
C                                                                       SOL06300
C-----------------------------------------                             SOL06310
C- CALCULATE AND PRINT TOTAL MEMBER FORCES ---                          SOL06320
C-----------------------------------------                             SOL06330
      LSTTYP=0                                                          SOL06340
      IF (WRITO) THEN                                                   SOL06350
          PRNT=.TRUE.                                                   SOL06360
          HEAD =.TRUE                                                   SOL06370
      ELSE                                                              SOL06380
          PRNT= FALSE.                                                  SOL06390
          HEAD =.FALSE.                                                 SOL06400
      ENDIF                                                             SOL06410
      DO 300 IELNO=1,NELMT                                              SOL06420
  300   CALL ELELIB                                                     SOL06430
     &  (5,LSTTYP,PRINT,IREL,HEAD,NAME,EESE,EPSE,DAMAGE,DUCFAG,         SOL06440
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDDSP,IZDLOA,MSTOR)    SOL06450
C                                                                       SOL06460
C-----------------------------------------                             SOL06470
C- CALCULATE AND PRINT SYSTEM DUCTILITY BASE                            SOL06480
C-----------------------------------------                             SOL06490
C.... SYSTEM DUCTILITY IS BASED ON THAT THE FIRST ELEMENT REACH TO      SOL06500
C     ITS CRITICAL LOAD                                                 SOL06510
      IF ( .NOT. ELSTIC) THEN                                           SOL06520
       LSTTYP=0                                                         SOL06530
       IF ( .NOT. DUCLOG) THEN                                          SOL06540
        PRNT= FALSE.                                                    SOL06550
        HEAD = FALSE.                                                   SOL06560
        DO 3120 IELNO=1,NELMT                                           SOL06570
          DUCFAG=0                                                      SOL06580
          CALL ELELIB                                                   SOL06590
     &    (11,LSTTYP ,PRINT ,IREL,HEAD,NAME,EESE,EPSE,DAMAGE,DUCFAG,    SOL06600
     &     IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDDSP,IZDLOA,MSTOR)  SOL06610
          IF( DUCFAG .GE. 1 ) THEN                                      SOL06620
            WRITE(108,3140) IELNO                                       SOL06630
            CALL JOIDSP(1,MAXNOD,NDOF,NNODE,1,L4,ID,IDOF,CNST,DISP,     SOL06640
     &                  TITLE,'DISPLACEMENTS',STEPID,.FALSE.,           SOL06650
     &                  NCOS,COSINE,JTCOS)                              SOL06660
            DUCLOG= .TRUE.                                              SOL06670
            GO TO 3130                                                  SOL06680
          ENDIF                                                         SOL06690
 3120   CONTINUE                                                        SOL06700
       ENDIF                                                            SOL06710
      ENDIF                                                             SOL06720
 3140 FORMAT (' SYSTEM DISPLACEMENT ( FIRST ELEMENT REACH TO CRITICAL',  SOL06730
     &        ' LOAD'/                                                  SOL06740
     &       ' ------------------------------------------------------',  SOL06750
     &       '-----'/                                                   SOL06760
     &       ' ** ELEMENT ',I5,' FIRST REACH TO CRITICAL LOAD **')      SOL06770
C                                                                       SOL06780
C                                                                       SOL06790
C-------------------------------------------------------                SOL06800
C- CALCULATE AND PRINT DUCTILITY AND EXCURSION RATIOS ---               SOL06810
C-------------------------------------------------------                SOL06820
 3130 LSTTYP=0                                                          SOL06830
      IF (WRITO) THEN                                                   SOL06840
        PRNT=.TRUE.                                                     SOL06850
        HEAD = TRUE.                                                    SOL06860
        DO 312 IELNO=1,NELMT                                            SOL06870
  312     CALL ELELIB                                                   SOL06880
     &    (11,LSTTYP,PRINT,IREL,HEAD,NAME,EESE,EPSE,DAMAGE,DUCFAG,      SOL06890
     &     IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDDSP,IZDLOA,MSTOR)  SOL06900
      ENDIF                                                             SOL06910
C                                                                       SOL06920
C-------------------------------                                        SOL06930
C- WRITE DATA TO OUTPUT FILES  ---                                      SOL06940
C-------------------------------                                        SOL06950
      IF (IWRITE GT 0.AND MOD(ISTEP,IWRITE).EQ.0) CALL                  SOL06960
     &                                        DMPDAT(2,IWRITE,TO,DT)    SOL06970
C                                                                       SOL06980
      DO 390 I=1,NDOF                                                   SOL06990
        IF( ABS(DISP(I)) .LE. 300 ) GOTO 390                            SOL07000
        WRITE(108,*) ' DISP(',I,') GREATER THAN 300. PROGRAM STOP'      SOL07010
        GO TO 5000                                                      SOL07020
  390 CONTINUE                                                          SOL07030
C                                                                       SOL07040
  500 CONTINUE                                                          SOL07050
C                                                                       SOL07060
C*************************************                                  SOL07070
C- PRINT MAXIMUM GLOBAL DISPLACEMENTS ---                               SOL07080
C*************************************                                  SOL07090
 5000 STEPID=' '                                                        SOL07100
      ISTEP = 0                                                         SOL07110
      WRITE(108,251) TITLE(1),TITLE(2)                                  SOL07120
  251 FORMAT ('1 STRUCTURE.....: ',A,/'  SOLUTION. ...: ',A,//          SOL07130
     &        '  MAXIMUN VALUES FOR ALL STEPS '/                        SOL07140
     &        ' ------------------------------')                       SOL07150
      CALL JOIDSP(2,MAXNOD,NDOF,NNODE,1,L4,ID,IDOF,CNST,DSPMAX,         SOL07160
     &            TITLE,'MAXIMUM DISPLACEMENTS',STEPID, FALSE.,       SOL07170
     &            NCOS,COSINE,JTCOS)                                  SOL07180
      WRITE(108,251) TITLE(1),TITLE(2)                                  SOL07190
      CALL JOIDSP(2,MAXNOD,NDOF,NNODE,1,L4,ID,IDOF,CNST,VELMAX,         SOL07200
     &            TITLE,'MAXIMUM VELOCITIES',STEPID, FALSE.,          SOL07210
     &            NCOS,COSINE,JTCOS)                                  SOL07220
      WRITE(108,251) TITLE(1),TITLE(2)                                  SOL07230
      CALL JOIDSP(2,MAXNOD,NDOF,NNODE,1,L4,ID,IDOF,CNST,ACCMAX,         SOL07240
     &            TITLE,'MAXIMUM ACCELERATIONS',STEPID,.FALSE.,       SOL07250
     &            NCOS,COSINE,JTCOS)                                  SOL07260
C                                                                       SOL07270
C-----------------------------------                                    SOL07280
C- PRINT MAXIMUM GLOBAL REACTIONS ---                                   SOL07290
C-----------------------------------                                    SOL07300
      WRITE(108,251) TITLE(1),TITLE(2)                                  SOL07310
      DO 5100 I=1,6                                                     SOL07320
        SUMRC(I)=RCTMAX(I+NDOF)                                         SOL07330
 5100   CONTINUE                                                        SOL07340
        CALL REACTN(4,.TRUE.,NNODE,L1,L4,L5,L6,RCTMAX,DSPMAX,           SOL07350
     &              MDMS   ,MASS  ,MDMS(NDOF),NDOF,1,ONE,               SOL07360
     &              MAXNOD,IDOF,COORD ,ID,TITLE,STEPID,.FALSE.,         SOL07370
     &              NCOS,COSINE,JTCOS,JTFLG,SUMRC)                      SOL07380
      WRITE(106,5111) SUMMFO,SUMMMO                                     SOL07390
 5111 FORMAT (/4X,'MAXIMUM RESULTANT OF REACTIONS,   FORCE=',1P,G12.4,   SOL07400
     &        '    MOMENT=',G12.4)                                      SOL07410
C                                                                       SOL07420
C-----------------------------------                                    SOL07430
C- PRINT MAXIMUM MEMBER FORCES ---                                      SOL07440
C-----------------------------------                                    SOL07450
      LSTTYP=0                                                          SOL07460
      PRINT=.TRUE.                                                      SOL07470
      HEAD =.FALSE.                                                     SOL07480
      WRITE(108,251) TITLE(1),TITLE(2)                                  SOL07490
      DO 5300 IELNO=1,NELMT                                             SOL07500
 5300   CALL ELELIB                                                     SOL07510
     &  (12,LSTTYP ,PRINT,IREL ,HEAD,NAME,EESE,EPSE,DAMAGE,DUCFAG,      SOL07520
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDDSP,IZDLOA,MSTOR)    SOL07530
C                                                                       SOL07540
C---------------------------------------                               SOL07550
C- MEMBER DUCTILITY AND EXCURSION RATIOS ---                            SOL07560
C---------------------------------------                               SOL07570
      IF ( NOT. ELSTIC) THEN                                            SOL07580
           LSTTYP=0                                                     SOL07590
           PRINT=.TRUE.                                                 SOL07600
           HEAD =.FALSE.                                                SOL07610
           WRITE(108,251) TITLE(1),TITLE(2)                             SOL07620
           DO 5310 IELNO=1,NELMT                                        SOL07630
 5310        CALL ELELIB                                               SOL07640
     &       (11,LSTTYP,PRINT,IREL ,HEAD,NAME ,EESE,EPSE,DAMAGE,DUCFAG, SOL07650
     &        IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDDSP,IZDLOA,MSTOR)SOL07660
      ENDIF                                                             SOL07670
C                                                                       SOL07680
C---------------------------                                            SOL07690
C- PRINT DAMAGE INDICIES  ---                                           SOL07700
C---------------------------                                           SOL07710
      IF (.NOT. ELSTIC) THEN                                            SOL07720
           LSTTYP=0                                                     SOL07730
           HEAD =.FALSE.                                                SOL07740
           PRINT=.TRUE                                                  SOL07750
           WRITE(108,251) TITLE(1),TITLE(2)                             SOL07760
           DAMAGE=0                                                     SOL07770
           DO 5320 IELNO=1,NELMT                                        SOL07780
             CALL ELELIB                                                SOL07790
     &       (14,LSTTYP,PRINT,IREL ,HEAD,NAME ,EESE,EPSE,EDAM,DUCFAG,   SOL07800
     &       IELNO,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDDSP,IZDLOA,MSTOR) SOL07810
 5320        DAMAGE=DAMAGE+EDAM                                         SOL07820
           IF( (ESE+PSE) .EQ. 0 ) RETURN                                SOL07830
           DAMAGE=DAMAGE/(ESE+PSE)                                      SOL07840
           WRITE(108,5330) DAMAGE                                       SOL07850
 5330      FORMAT (/10X,'STRUCTURE DAMAGE INDEX=',1P,G15.5)             SOL07860
      ENDIF                                                             SOL07870
C                                                                       SOL07880
C---------------------------                                            SOL07890
C- PRINT PEAK ENERGY INFO ---                                           SOL07900
C---------------------------                                           SOL07910
      WRITE(108,251) TITLE(1),TITLE(2)                                  SOL07920
      WRITE(108,5340) EIEM,ESEM,PSEM,EKEM,EDDM                          SOL07930
 5340 FORMAT (/'   PEAK ENERGY VALUES ' / 1P,                          SOL07940
     &  ' MAX INPUT ENERGY.....',G12.5/                                 SOL07950
     &  ' MAX ELASTIC STRAIN ENERGY.....',G12.5/                        SOL07960
     &  ' MAX PLASTIC STRAIN ENERGY.....',G12.5/                        SOL07970
     &  ' MAX KINETIC ENERGY..... ',G12.5/                              SOL07980
     &  ' MAX DAMPING ENERGY .....',G12.5/)                             SOL07990
C                                                                       SOL08000
  250 FORMAT ('1 STRUCTURE.....: ',A,/'  SOLUTION.....: ',A,//          SOL08010
     &        2X,A,',',' LOADING #',I5,5X,A)                            SOL08020
C                                                                       SOL08030
      RETURN                                                            SOL08040
      END                                                               SOL08050
C                                                                       JOID0010
C-----------------------------------------------------------------------JOID0020
C                                                                       JOID0030
      SUBROUTINE JOIDSP(IOPT,MAXNOD,NDOF,NNODE,NLOAD,NP,ID,IDOF,        JOID0040
     &                  CNST,DISP,TITLE,NAME,STEPID,HEAD,NCOS,COSINE,JTCOS) JOID0050
      CHARACTER*(*) TITLE(2),STEPID,NAME                                JOID0060
      CHARACTER*15 CD(6),EQUAL*60                                       JOID0070
      LOGICAL HEAD                                                      JOID0080
      DIMENSION IDOF(MAXNOD,6),CONST(MAXNOD,6)                          JOID0090
      DIMENSION DISP(NDOF,NLOAD),ID(MAXNOD)                             JOID0100
      DIMENSION D(6),G(6),COSINE(3,3,NCOS),JTCOS(MAXNOD)                JOID0110
      DATA EQUAL                                                        JOID0120
     &  /'------------------------------------------------------------'/JOID0130
C                                                                       JOID0140
      NLEN=MIN(LEN(NAME)+1,60)                                          JOID0150
      NC=1                                                              JOID0160
C                                                                       JOID0170
C-------------------------------                                        JOID0180
C- LOOP FOR EACH LOAD CASE  ---                                         JOID0190
C-------------------------------                                        JOID0200
C                                                                       JOID0210
      DO 300 L=1,NLOAD                                                  JOID0220
        NDSPL=0                                                         JOID0230
        IF (.NOT.(NCOS.EQ.1 .AND. COSINE(1,1,1) EQ.1 .AND.              JOID0240
     &      COSINE(2,2,1).EQ.1  AND. COSINE(3,3,1).EQ.1 )               JOID0250
     &      AND. IOPT.EQ.2 ) GO TO 310                                  JOID0260
C                                                                       JOID0270
C-------------------------------                                        JOID0280
C- PRINT GLOBAL DISPLACEMENTS ---                                       JOID0290
C-------------------------------                                        JOID0300
C  NOTE:DISP MAY CONTAIN DISPLACEMENTS, VELOCITIES, ACCELERATIONS OR    JOID0310
C       EIGENVALUES                                                     JOID0320
C       ALL OF WHICH ARE PRINTED OUT AT EACH NODE. ..                   JOID0330
C                                                                       JOID0340
        IF (HEAD) WRITE(108,310) TITLE(1),TITLE(2)                      JOID0350
        IF (IOPT.NE.2) WRITE(108,315) NAME,L,STEPID,EQUAL(1:NLEN)       JOID0360
        IF (IOPT.EQ.2) WRITE(108,&15) NAME,          EQUAL(1:NLEN)      JOID0370
        DO 100 I=1,NNODE                                                JOID0380
C                                                                       JOID0390
C ... ..... TRANSFER DISPL TO LOCAL VECTOR :. ..... .                  JOID0400
          DO 20 J=1,6                                                   JOID0410
            IJ=IDOF(I,J)                                                JOID0420
   20       D(J)=DISP(IJ,L)                                             JOID0430
C                                                                       JOID0440
C.. . . . . ADD ROTATIONS TO DISPL FOR CONSTRAINTS                      JOID0450
C.... .. ..   IF UNCONSTRAINED, CONST(I,J)=0                           JOID0460
          D(1) = D(1) + D(5)*CONST(I,3) + D(6)*CONST(I,5)               JOID0470
```

```
          D(2) = D(2) + D(4)*CONST(I,1) + D(6)*CONST(I,6)                JOID0480
          D(3) = D(3) + D(4)*CONST(I,2) + D(5)*CONST(I,4)                JOID0490
C                                                                        JOID0500
C.....  ....  TRANSFER TO GLOBAL DISPLACEMENTS  ....  .                  JOID0510
          JCOS=JTCOS(I)                                                  JOID0520
          DO 25 I1=1,3                                                   JOID0530
            I2=I1+3                                                      JOID0540
            G(I1)=0                                                      JOID0550
            G(I2)=0                                                      JOID0560
            DO 25 J1=1,3                                                 JOID0570
              J2=J1+3                                                    JOID0580
              G(I1)=G(I1)+COSINE(J1,I1,JCOS)*D(J1)                       JOID0590
   25         G(I2)=G(I2)+COSINE(J1,I1,JCOS)*D(J2)                       JOID0600
C                                                                        JOID0610
C.....  .....  WRITE DISPL                                              JOID0620
          DO 30 J=1,6                                                    JOID0630
    30    WRITE(CD(J),9) G(J)                                            JOID0640
     9    FORMAT(1P,G14.6,' ')                                           JOID0650
          WRITE(108,220) ID(I),(CD(K),K=1,6)                             JOID0660
C                                                                        JOID0670
          NDSPLS=NDSPLS+1                                                JOID0680
  100 CONTINUE                                                           JOID0690
C                                                                        JOID0700
      IF (NCOS.EQ.1 .AND. COSINE(1,1,1).EQ.1 .AND. COSINE(2,2,1) EQ.1    JOID0710
     &                .AND. COSINE(3,3,1).EQ 1 ) GO TO 300              JOID0720
C                                                                        JOID0730
C------------------------------                                          JOID0740
C- PRINT JOINT DISPLACEMENTS ---                                         JOID0750
C------------------------------                                          JOID0760
C  NOTE DISP MAY CONTAIN DISPLACEMENTS, VELOCITIES, ACCELERATIONS OR     JOID0770
C       EIGENVALUES...                                                   JOID0780
C       ALL OF WHICH ARE PRINTED OUT AT EACH NODE...                     JOID0790
C                                                                        JOID0800
  210   IF (HEAD AND. NDSPLS GT.20) WRITE(108,310) TITLE(1),TITLE(2)     JOID0810
        IF (IOPT.NE.2) WRITE(108,316) NAME,L,STEPID,EQUAL(1:NLEN)        JOID0820
        IF (IOPT.EQ 2) WRITE(108,416) NAME,          EQUAL(1:NLEN)       JOID0830
        DO 200 I=1,NNODE                                                 JOID0840
C                                                                        JOID0850
C.. .....  TRANSFER GLOBAL DISPL TO LOCAL VECTOR .....  .               JOID0860
          DO 219 J=1,6                                                   JOID0870
            IJ=IDOF(I,J)                                                 JOID0880
  219       D(J)=DISP(IJ,L)                                            JOID0890
C                                                                        JOID0900
C.....  .. . .  ADD ROTATIONS TO DISPL FOR CONSTRAINTS                   JOID0910
C.....  .....  IF UNCONSTRAINED, CONST(I,J)=0                           JOID0920
          D(1) = D(1) + D(5)*CONST(I,3) + D(6)*CONST(I,5)                JOID0930
          D(2) = D(2) + D(4)*CONST(I,1) + D(6)*CONST(I,6)                JOID0940
          D(3) = D(3) + D(4)*CONST(I,2) + D(5)*CONST(I,4)                JOID0950
C                                                                        JOID0960
C  ..  .  .  .  WRITE DISPL                                              JOID0970
          DO 230 J=1,6                                                   JOID0980
            WRITE(CD(J),9) D(J)                                          JOID0990
            IJ=IDOF(I,J)                                                 JOID1000
  230       IF (IJ.GT.NF) CD(J)(15:15)='*'                             JOID1010
        WRITE(108,221) ID(I),(CD(K),K=1,6),JTCOS(I)                      JOID1020
C                                                                        JOID1030
  200 CONTINUE                                                           JOID1040
      WRITE(108,320)                                                     JOID1050
  300 CONTINUE                                                           JOID1060
C                                                                        JOID1070
   10 FORMAT (1P,G15.7)                                                  JOID1080
  220 FORMAT (5X,I5,6A)                                                  JOID1090
  221 FORMAT (5X,I5,6A,I10)                                              JOID1100
  310 FORMAT ('1 STRUCTURE.....: ',A,/'  SOLUTION.....: ',A)             JOID1110
  315 FORMAT (//2X, 'GCS ',A,',  LOADING #',I5,5X,A, /                  JOID1120
     &         1X,'------',A//6X,'NODE',7X,'DX',13X,'DY',13X,'DZ',      JOID1130
     &         13X,'RX',13X,'RY',13X,'RZ'/)                            JOID1140
  316 FORMAT (//2X, 'JCS ',A,',  LOADING #',I5,5X,A, /1X,'------',A      JOID1150
     &         //6X,'NODE',7X,'DX',13X,'DY',13X,'DZ',13X,'RX',         JOID1160
     &         13X,'RY',13X,'RZ',6X,'  COSINE #'/)                     JOID1170
  415 FORMAT (//2X, 'GCS ',A/1X,'------',A/25X,                         JOID1180
     &       'NOTE: MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'//6X,    JOID1190
     &  'NODE',7X,'DX',13X,'DY',13X,'DZ',13X,'RX',13X,'RY',13X,'RZ'/)    JOID1200
  416 FORMAT (//2X, 'JCS ',A/1X,'------',A/25X,                         JOID1210
     &       'NOTE: MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'//6X,    JOID1220
     &   'NODE',7X,'DX',13X,'DY',13X,'DZ',13X,'RX',13X,'RY',13X,'RZ',    JOID1230
     &   6X,'  COSINE #'/)                                               JOID1240
  320 FORMAT (4X,'NOTE: * DENOTES A RESTRAINED DEGREE OF FREEDOM')       JOID1250
C                                                                        JOID1260
      RETURN                                                             JOID1270
      END                                                                JOID1280
C                                                                        TMPL0010
C-----------------------------------------------------------------------TMPL0020
C   TRIANGLAR MATRIX MULTIPLICATION .....                                TMPL0030
C   MATRIX A IS AN UPPER TRIANGULAR MATRIX, WITH MAIN DIAGONAL ADDRESS   TMPL0040
C   MATRIX MD.  X IS A VECTOR.  F=AX                                     TMPL0050
C   THE MULTIPLICATION IS CARRIED OUT FROM ROW IR1 TO IR2  AND FROM      TMPL0060
C   COLUMN JC1 TO JC2.                                                   TMPL0070
C-----------------------------------------------------------------------TMPL0080
      SUBROUTINE TMPLY(PRINT,ZERO,IMD,N,IR1,IR2,JC1,JC2,MD,A,P,X)        TMPL0090
      DIMENSION A(IMD),X(N),P(N),MD(N+1)                                 TMPL0100
      LOGICAL PRINT,ZERO                                                 TMPL0110
C                                                                        TMPL0120
      IJI(I,J)=MD(J)+J-I                                                 TMPL0130
      IF (PRINT) THEN                                                    TMPL0140
         WRITE(108,500) 'TRIANGULAR MULTIPLACATION',IR1,IR2,JC1,JC2,N    TMPL0150
         CALL LMATRX(A,MD,N,MD(N+1),'MATRIX A ')                         TMPL0160
      ENDIF                                                              TMPL0170
C     .....  ZERO P  .....                                              TMPL0180
      IF (ZERO) THEN                                                     TMPL0190
        DO 320 J=IR1,IR2                                                 TMPL0200
  320     P(J)=0                                                         TMPL0210
      ENDIF                                                              TMPL0220
C                                                                        TMPL0230
C.......  MULTIPLY TERMS ABOVE AND INCLUDING THE MAIN DIAGONAL          TMPL0240
       DO 20 J=JC1,JC2                                                   TMPL0250
         IRC=J-MD(J-1)+MD(J)+1                                           TMPL0260
         MDJ=MD(J)+J                                                     TMPL0270
         I1=MAX(IRC,IR1)                                                 TMPL0280
         I2=MIN(J,IR2)                                                   TMPL0290
         IF (I2.GE.I1)  THEN                                             TMPL0300
           DO 10 I=I1,I2                                                 TMPL0310
   10      P(I)=P(I)+A(MDJ-I)*X(J)                                       TMPL0320
         ENDIF                                                           TMPL0330
   20  CONTINUE                                                          TMPL0340
C                                                                        TMPL0350
C........  MULTIPLY TERMS BELOW THE MAIN DIAGONAL                       TMPL0360
       DO 30 I=IR1,IR2                                                   TMPL0370
         IRC=I-MD(I-1)+MD(I)+1                                           TMPL0380
         MDI=MD(I)+I                                                     TMPL0390
         J1=MAX(IRC,JC1)                                                 TMPL0400
         J2=MIN(I-1,JC2)                                                 TMPL0410
         IF (J2.GE.J1) THEN                                              TMPL0420
           DO 40 J=J1,J2                                                 TMPL0430
   40      P(I)=P(I)+A(MDI-J)*X(J)                                       TMPL0440
         ENDIF                                                           TMPL0450
   30  CONTINUE                                                          TMPL0460
  500 FORMAT (/2X,A/,2X,' IR1.',I5,' IR2.',I5,                           TMPL0470
     &                 ' JC1:',I5,' JC2 ',I5,' N ',I5)                 TMPL0480
      IF (PRINT) THEN                                                    TMPL0490
        DO 515 I=IR1,IR2                                                 TMPL0500
  515   WRITE(108,520) I,X(I),P(I)                                       TMPL0510
        WRITE(108,*) ' '                                                 TMPL0520
  520 FORMAT (2X,'DOF ',I6,'  X(DOF)',1P,G14.5,' A*X=P(DOF):',G14.5)     TMPL0530
      ENDIF                                                              TMPL0540
      RETURN                                                             TMPL0550
      END                                                                TMPL0560
C                                                                        ETMP0010
C-----------------------------------------------------------------------ETMP0020
C   CALC THE ENERGY (E) FOR U(TRANSP) * P = U(TRANSP) * A * X            ETMP0030
C                                                                        ETMP0040
C - MATRIX A IS AN UPPER TRIANGULAR MATRIX, WITH MAIN DIAGONAL ADDRESS   ETMP0050
C    MATRIX MD.                                                          ETMP0060
C    DOF L1 TO L2 HAVE BEEN CONDENSED OUT                                ETMP0070
C    DOF L3 TO L4 HAVE BEEN MODIFIED FOR THE CONDENSATION OF L1 TO L2    ETMP0080
C    DOF L5 TO L6 HAVE NOT BEEN MODIFIED                                 ETMP0090
C - X IS A VECTOR.                                                       ETMP0100
C - P IS A DUMMY INPUT VECTOR.                                           ETMP0110
C-----------------------------------------------------------------------ETMP0120
      SUBROUTINE ETMPLY(PRINT,L1,L2,L3,L4,L5,L6,IMD,MD,A,P,X,U,E)        ETMP0130
      DIMENSION A(IMD),X(L6),P(L6),MD(L6+1),U(L6)                        ETMP0140
      LOGICAL PRINT,T,F                                                  ETMP0150
      IJI(I,J)=MD(J)+J-I                                                 ETMP0160
C                                                                        ETMP0170
      T=.TRUE.                                                           ETMP0180
      F=.FALSE.                                                          ETMP0190
C                                                                        ETMP0200
      IF (PRINT) THEN                                                    ETMP0210
         WRITE(108,500) 'CALC ENERGY FOR PARTIALLY REDUCED A MATRIX',    ETMP0220
     &          L1,L2,L3,L4,L5,L6                                       ETMP0230
         CALL LMATRX(A,MD,L6,IMD,'MATRIX A ')                            ETMP0240
         DO 515 I=L1,L6                                                  ETMP0250
  515    WRITE(108,520) I,X(I),U(I)                                      ETMP0260
      ENDIF                                                              ETMP0270
C                                                                        ETMP0280
C--- ------- - CALC PP" = KPP"*X1                                       ETMP0290
      CALL TMPLY(F,T,IMD,L6,L3,L4,L3,L4,MD,A,P,X)                        ETMP0300
C                                                                        ETMP0310
C--------- ---- CALC E1 = XP' * PP"                                     ETMP0320
      E=0                                                                ETMP0330
      DO 10 L=L3,L4                                                      ETMP0340
   10   E = E + U(L)*P(L)                                               ETMP0350
      E1=E                                                               ETMP0360
C                                                                        ETMP0370
C-------- ---- CALC PP  = KFR *XR                                       ETMP0380
      CALL TMPLY(F,T,IMD,L6,L1,L4,L5,L6,MD,A,P,X)                        ETMP0390
C                                                                        ETMP0400
C------------ CALC PP  = KFR *KR                                        ETMP0410
      CALL TMPLY(F,T,IMD,L6,L5,L6,L1,L6,MD,A,P,X)                        ETMP0420
C                                                                        ETMP0430
C------------ CALC E2 + E3 + E4 = XP * PP + XR * PR                     ETMP0440
      DO 20 L=L1,L6                                                      ETMP0450
   20   E = E + U(L)*P(L)                                               ETMP0460
C                                                                        ETMP0470
C------------ PRINT ANSWER  . .                                         ETMP0480
      IF (PRINT) WRITE(108,510) E,E1                                     ETMP0490
C                                                                        ETMP0500
  500 FORMAT (/2X,A,/2X,'L1:',I5,'   L2:',I5,'   L3:',I5,'   L4.',I5,   ETMP0510
     &          '   L5 ',I5,'   L6:',I5)                                ETMP0520
  510 FORMAT (/2X,' TOTAL ENERGY:',1P,G13.5,'  E1:',G11.5)               ETMP0530
  520 FORMAT (2X,'DOF:',I6,'  X(DOF)',1P,G14.5,'  U(DOF)=',G14.5)        ETMP0540
      RETURN                                                             ETMP0550
      END                                                                ETMP0560
C                                                                        INTD0010
C-----------------------------------------------------------------------INTD0020
C                                                                        INTD0030
      SUBROUTINE INTDVA(NDOF,MAXNOD,NNODE,TITLE,                         INTD0040
     &          IDOF,ID,JTFLG,DISP,VEL,ACC,FORCE )                       INTD0050
C                                                                        INTD0060
      COMMON /ISPEC/ISP1,SGD(500,2),REACF(6)                             INTD0070
      DIMENSION DISP(NDOF),VEL(NDOF),ACC(NDOF),A(6),FORCE(NDOF)          INTD0080
      DIMENSION ID(MAXNOD),JTFLG(MAXNOD),IDOF(MAXNOD,6)                  INTD0090
      CHARACTER*(*) TITLE(2)                                             INTD0100
      CHARACTER*20 OPTION                                                INTD0110
      LOGICAL HEAD,TEST,FLAG,BTEST                                       INTD0120
C                                                                        INTD0130
C----------------------------------------                                INTD0140
C- INPUT INITIAL DISPL, VELOC AND ACCEL  ---                             INTD0150
C----------------------------------------                                INTD0160
C                                                                        INTD0170
      HEAD=.TRUE.                                                        INTD0180
C- CONSIDER EXTERNAL DEFORMATION DUE TO STATICS CASE, ISP1=1 ---          INTD0190
C     NOTE: 'RELEASE' SHOULD NOT USED AFTER 'SOL01'                      INTD0200
C- NEGLECT  EXTERNAL DEFORMATION DUE TO STATICS CASE, ISP1=0 ---          INTD0210
C     NOTE: 'RELEASE' SHOULD USED AFTER 'SOL01'                          INTD0220
      READ(105,*) ISP1                                                   INTD0230
      WRITE(109,*)                                                       INTD0240
      IF (ISP1 .EQ. 0 ) THEN                                             INTD0250
        WRITE(108,*) ' NOT INCLUDE STATICS CASE EXTERNAL DEFORMATIONS '  INTD0260
        WRITE(108,*) ' ------------------------------------------------' INTD0270
      ELSE                                                               INTD0280
        WRITE(108,*) ' STATICS CASE EXTERNAL DEFORMATIONS ARE INCLUDED'  INTD0290
        WRITE(108,*) ' ------------------------------------------------' INTD0300
        DO 49 I=1,NDOF                                                   INTD0310
          DISP(I)=SGD(I,2)                                               INTD0320
          VEL(I)=0                                                       INTD0330
          ACC(I)=0                                                       INTD0340
          FORCE(I)=0                                                     INTD0350
   49   CONTINUE                                                         INTD0360
      ENDIF                                                              INTD0370
C*                                                                       INTD0380
C--- INPUT: NODE,IGEN,INC,OPTION,DX,DY,DZ,RX,RY,RZ                       INTD0390
C                                                                        INTD0400
      IF( ISP1 .NE. 1 ) THEN                                             INTD0410
   10  READ(105,*) NODE,IGEN,INC,OPTION,(A(I),I=1,6)                     INTD0420
        IF (NODE.LE.0) GO TO 500                                         INTD0430
        IF (HEAD) THEN                                                   INTD0440
          HEAD=.FALSE.                                                   INTD0450
          WRITE(108,5) TITLE(1),TITLE(2)                                 INTD0460
        ENDIF                                                            INTD0470
        WRITE(108,7) NODE,IGEN,INC,OPTION,(A(I),I=1,6)                   INTD0480
    5   FORMAT ('1 STRUCTURE.....: ',A,/'  SOLUTION.....: ',A,//        INTD0490
     &       '  INITIAL ACCELERATIONS, VELOCITIES DISPLACEMENTS ',       INTD0500
     &       ' AND LOADS '/                                              INTD0510
     &       ' ------------------------------------------------',        INTD0520
     &       '----------'//4X,'NODE',2X,'IGEN',3X,'INC',1X,'COMPONENT',  INTD0530
     &       T44,'TX',10X,'TY',10X,'TZ',10X,'RX',10X,'RY',10X,'RZ')      INTD0540
    7   FORMAT (2X,3I6,1X,A,1P,6G12.4)                                   INTD0550
C                                                                        INTD0560
C------- DETERMINE IF DISPLACMENT, VELOCITY, ACCELERATION OR LOAD        INTD0570
      IOPT=0                                                             INTD0580
      IF (TEST(OPTION,'DISPL',.FALSE.)) THEN                             INTD0590
        IOPT=IBSET(IOPT,1)                                               INTD0600
      ELSE IF (TEST(OPTION,'VEL',.FALSE.)) THEN                          INTD0610
        IOPT=IBSET(IOPT,2)                                               INTD0620
      ELSE IF (TEST(OPTION,'ACC',.FALSE.)) THEN                          INTD0630
        IOPT=IBSET(IOPT,3)                                               INTD0640
      ELSE IF (TEST(OPTION,'LOA', FALSE.)) THEN                          INTD0650
        IOPT=IBSET(IOPT,4)                                               INTD0660
      ELSE IF (TEST(OPTION,'FOR', FALSE.)) THEN                          INTD0670
        IOPT=IBSET(IOPT,4)                                               INTD0680
      ELSE IF (TEST(OPTION,'END', FALSE.)) THEN                          INTD0690
        GO TO 500                                                        INTD0700
      ELSE                                                               INTD0710
        WRITE(108,*) 'INVALID OPTION:',OPTION                            INTD0720
        GO TO 10                                                         INTD0730
      ENDIF                                                              INTD0740
C                                                                        INTD0750
C ------- DETERMINE INTERNAL JOINT NUMBER                                INTD0760
   20 JOINT=IQUICK(NODE,ID,NNODE)                                        INTD0770
      IF (JOINT.LE.0) THEN                                               INTD0780
        WRITE(108,*) 'NODE #',NODE,                                      INTD0790
     &          ' WAS NOT FOUND, CHECK INPUT, INITIAL VALUE IS OMITTED ' INTD0800
        GO TO 10                                                         INTD0810
      ENDIF                                                              INTD0820
C                                                                        INTD0830
C-- ---- PUT INITIAL VALUES IN PROPER ARRAYS                            INTD0840
      DO 30 I=1,6                                                        INTD0850
        IF (A(I).EQ.0) GO TO 30                                          INTD0860
        II=IDOF(JOINT,I)                                                 INTD0870
        IF (BTEST(JTFLG(JOINT),I-1)) THEN                                INTD0880
          WRITE(108,*) 'CONSTRAINED DOF #',I,' COMPONENT IS SKIPED'      INTD0890
        ELSE IF (BTEST(IOPT,1)) THEN                                     INTD0900
          DISP(II)=A(I)                                                  INTD0910
        ELSE IF (BTEST(IOPT,2)) THEN                                     INTD0920
          VEL(II)=A(I)                                                   INTD0930
        ELSE IF (BTEST(IOPT,3)) THEN                                     INTD0940
          ACC(II)=A(I)                                                   INTD0950
        ELSE IF (BTEST(IOPT,4)) THEN                                     INTD0960
          FORCE(II)=A(I)                                                 INTD0970
        ENDIF                                                            INTD0980
   30 CONTINUE                                                           INTD0990
```

```
C                                                                       INTD1000
C------- GENERATE VALUES AT OTHER NODES                                 INTD1010
      IF (IGEN GT.0) THEN                                               INTD1020
        IGEN=IGEN-1                                                     INTD1030
        NODE=NODE+INC                                                   INTD1040
        GO TO 20                                                        INTD1050
      ENDIF                                                             INTD1060
C                                                                       INTD1070
C------- LOOP TO READ ADDITIONAL VALUES                                 INTD1080
      GO TO 10                                                          INTD1090
      ENDIF                                                             INTD1100
C                                                                       INTD1110
  500 CONTINUE                                                          INTD1120
C     ISP1=0                                                            INTD1130
C                                                                       INTD1140
C------- PRINT OUT NON ZERO INITIAL VALUES                              INTD1150
      FLAG= TRUE                                                        INTD1160
      DO 520 I=1,NDOF                                                   INTD1170
        IF (DISP(I) NE.0 .OR. VEL(I).NE.0  OR. ACC(I) NE.0 .OR.         INTD1180
     &      FORCE(I) NE 0 ) THEN                                        INTD1190
          IF (FLAG) WRITE(108,510)                                      INTD1200
          WRITE(108,530) I,FORCE(I),DISP(I),VEL(I),ACC(I)               INTD1210
          FLAG=.FALSE.                                                  INTD1220
        ENDIF                                                           INTD1230
  520 CONTINUE                                                          INTD1240
C                                                                       INTD1250
  510 FORMAT(//5X,'INITIAL, NON ZERO VALUES'/7X,'DOF',                  INTD1260
     &       7X,'FORCE',10X,'DISP',11X,'VEL ',11X,'ACC ')               INTD1270
  530 FORMAT (I10,4G15.6)                                               INTD1280
C                                                                       INTD1290
      RETURN                                                            INTD1300
      END                                                               INTD1310
C                                                                       DYL00010
C-----------------------------------------------------------------------DYL00020
C                                                                       DYL00030
      SUBROUTINE DYLOA(IOPT,N,NDOF,NAIN,T,TL,ACCEL2,TO,DT,GRAV,         DYL00040
     &                 MAXNOD,NNODE,COSINE,IDOF,JTFLG,JTCOS,NCOS,       DYL00050
     &                 TX,TY,TZ,MASS,MD,IMD,A,MC,P,GA,GAT)              DYL00060
C                                                                       DYL00070
      LOGICAL FMT,ECHO,PRINT,REWIND,BTEST                               DYL00080
      CHARACTER*80 FORMAT(2),ATITLE(2)                                  DYL00090
      REAL      MASS(IMD),MC(NDOF,3)                                    DYL00100
      DIMENSION COSINE(3,3,NCOS),IDOF(MAXNOD,6),JTCOS(MAXNOD)          DYL00110
      DIMENSION TX(N),TY(N),TZ(N),A(NAIN),P(NDOF),JTFLG(MAXNOD)         DYL00120
      DIMENSION V1(3),V2(3),V3(3),MD(NDOF+1),GA(3),GAT(3)               DYL00130
C                                                                       DYL00140
      ACCEL2=0                                                          DYL00150
      GO TO (100,200) IOPT                                              DYL00160
C                                                                       DYL00170
C-------------------------                                              DYL00180
C- INPUT ACCELERATIONS ---                                              DYL00190
C-------------------------                                              DYL00200
  100 CONTINUE                                                          DYL00210
C ------ ZERO GLOBAL BASE ACCELERATIONS                                 DYL00220
      DO 101 I=1,N                                                      DYL00230
        TX(I)=0                                                         DYL00240
        TY(I)=0                                                         DYL00250
  101   TZ(I)=0                                                         DYL00260
C                                                                       DYL00270
C------- INPUT NUMBER OF ACCEL, AMP  FACTOR SCALE, AND TIME INCREMENT   DYL00280
      READ(105,*) NA,ASCALE,TO,DT,PRINT                                 DYL00290
C                                                                       DYL00300
C---- -- INPUT DIRECTION VECTORS V1 AND V2                              DYL00310
      READ(105,*) V1,V2                                                 DYL00320
C                                                                       DYL00330
C------- CALCULATE V3 = V1 X V2                                         DYL00340
      CALL CROSS(V3,V1,V2,0)                                            DYL00350
C                                                                       DYL00360
C--------CALCULATE V2 = V3 X V1, AND NORMALIZE ALL VECTORS. .           DYL00370
      CALL CROSS(V2,V3,V1,1)                                            DYL00380
C                                                                       DYL00390
C-------- INPUT INDIVIDUAL ACCELEROGRAMS                                DYL00400
      DO 30 I=1,NA                                                      DYL00410
        READ(105,*) IN,NPTS,IDIR,FMT,ECHO ,REWIND                       DYL00420
        NPTS=MIN(NPTS,N)                                                DYL00430
        IF (REWIND.AND.IN.NE.105) REWIND (IN)                           DYL00440
C                                                                       DYL00450
        IF (FMT) THEN                                                   DYL00460
          READ(105,91)  FORMAT(1),FORMAT(2)                             DYL00470
          READ(IN,FORMAT(1)) ATITLE(1),ATITLE(2)                        DYL00480
        ELSE                                                            DYL00490
          READ(IN,91) ATITLE(1),ATITLE(2)                               DYL00500
        ENDIF                                                           DYL00510
C                                                                       DYL00520
        IF (IDIR.GE.1 .AND. IDIR.LE.3) THEN                             DYL00530
          WRITE(108,10) I,IN,ATITLE(1),ATITLE(2),NPTS,TO,DT             DYL00540
   10     FORMAT (/5X,' GROUND ACCELERATION RECORD '/                   DYL00550
     &            5X,'---------------------------'//                   DYL00560
     &            5X,'INPUTING TRANSLATIONAL ACCELERATION RECORD #',   DYL00570
     &            I2,', FROM UNIT:',I4,/,5X,A/,5X,A/                   DYL00580
     &            5X,'THE RECORD CONTAINS ',I6,' POINTS, ',            DYL00590
     &               ' BEGINING AT TIME:',1P,G16.6,                    DYL00600
     &               ' WITH A TIME INCREMENT OF:',1P,G15.6)            DYL00610
        ELSE                                                            DYL00620
          WRITE(108,*) 'INVALID DIRECTION:',IDIR,' SET 0 < IDIR < 4'    DYL00630
          GO TO 30                                                      DYL00640
        ENDIF                                                           DYL00650
C                                                                       DYL00660
        IF (IDIR EQ 1) THEN                                             DYL00670
          CX=V1(1)                                                      DYL00680
          CY=V1(2)                                                      DYL00690
          CZ=V1(3)                                                      DYL00700
          WRITE(108,92) V1                                              DYL00710
        ELSE IF (IDIR EQ.2) THEN                                        DYL00720
          CX=V2(1)                                                      DYL00730
          CY=V2(2)                                                      DYL00740
          CZ=V2(3)                                                      DYL00750
          WRITE(108,92) V2                                              DYL00760
        ELSE IF (IDIR.EQ.3) THEN                                        DYL00770
          CX=V3(1)                                                      DYL00780
          CY=V3(2)                                                      DYL00790
          CZ=V3(3)                                                      DYL00800
          WRITE(108,92) V3                                              DYL00810
        ENDIF                                                           DYL00820
C                                                                       DYL00830
        IF (FMT) THEN                                                   DYL00840
          READ(IN,FORMAT(2),END=25) (A(J),J=1,NPTS)                     DYL00850
        ELSE                                                            DYL00860
          READ(IN,*,END=25) (A(J),J=1,NPTS)                             DYL00870
        ENDIF                                                           DYL00880
C                                                                       DYL00890
        GO TO 27                                                        DYL00900
C---------- END OF FILE WAS ENCOUNTERED BEFORE ALL INPUT WAS READ       DYL00910
   25   NPTS=J                                                          DYL00920
        WRITE(108,26) J                                                 DYL00930
   26   FORMAT(5X,'ONLY',I5,                                            DYL00940
     &         ' POINTS WERE READ BEFORE END OF FILE WAS ENCOUNTERED')   DYL00950
C                                                                       DYL00960
   27   IF (ECHO) WRITE(108,94) (A(J),J=1,NPTS)                         DYL00970
        WRITE(108,9280)                                                 DYL00980
        CALL SMAX (NPTS,DT,A,YMX,YMN,T1,T2,NPTS,AVE,STDEV,RMS)          DYL00990
        T1=T1+TO                                                        DYL01000
        T2=T2+TO                                                        DYL01010
        WRITE(108,9290) 'INPUT ACCELERATION ',                          DYL01020
     &                  YMX,T1,YMN,T2,AVE,STDEV,RMS                     DYL01030
C                                                                       DYL01040
        DO 20 J=1,NPTS                                                  DYL01050
          ACC=A(J)*ASCALE                                               DYL01060
          TX(J)=TX(J) + ACC*CX                                          DYL01070
          TY(J)=TY(J) + ACC*CY                                          DYL01080
          TZ(J)=TZ(J) + ACC*CZ                                          DYL01090
   20   CONTINUE                                                        DYL01100
C                                                                       DYL01110
   30 CONTINUE                                                          DYL01120
      IF (PRINT) THEN                                                   DYL01130
        WRITE(108,31) 'X'                                               DYL01140
        WRITE(108,93) (TX(J),J=1,N)                                     DYL01150
        WRITE(108,31) 'Y'                                               DYL01160
        WRITE(108,93) (TY(J),J=1,N)                                     DYL01170
        WRITE(108,31) 'Z'                                               DYL01180
        WRITE(108,93) (TZ(J),J=1,N)                                     DYL01190
   31   FORMAT(///5X,'GLOBAL BASE ACCELERATION IN THE ',A,              DYL01200
     &          ' DIRECTION (g''s)  ')                                  DYL01210
      ENDIF                                                             DYL01220
C                                                                       DYL01230
      WRITE(108,9280)                                                   DYL01240
      CALL SMAX (N,DT,TX,YMX,YMN,T1,T2,N,AVE,STDEV,RMS)                 DYL01250
      T1=T1+TO                                                          DYL01260
      T2=T2+TO                                                          DYL01270
      WRITE(108,9290) 'X-AXIS ACCELERATION ',YMX,T1,YMN,T2,AVE,STDEV,RMSDYL01280
      CALL SMAX (N,DT,TY,YMX,YMN,T1,T2,N,AVE,STDEV,RMS)                 DYL01290
      T1=T1+TO                                                          DYL01300
      T2=T2+TO                                                          DYL01310
      WRITE(108,9290) 'Y-AXIS ACCELERATION ',YMX,T1,YMN,T2,AVE,STDEV,RMSDYL01320
      CALL SMAX (N,DT,TZ,YMX,YMN,T1,T2,N,AVE,STDEV,RMS)                 DYL01330
      T1=T1+TO                                                          DYL01340
      T2=T2+TO                                                          DYL01350
      WRITE(108,9290) 'Z-AXIS ACCELERATION ',YMX,T1,YMN,T2,AVE,STDEV,RMSDYL01360
C                                                                       DYL01370
C----- --- CALCULATE MASS * COSINE  MATRIX FOR EACH DOF..              DYL01380
C..... LOOP FOR DIRECTION   JDIR                                       DYL01390
      DO 90 JDIR=1,3                                                    DYL01400
C                                                                       DYL01410
C. .... ZERO MC MATRIX AND TEMP COSINE  MATRIX FOR JDIR                 DYL01420
        DO 40 I=1,NDOF                                                  DYL01430
          MC(I,JDIR)=0                                                  DYL01440
   40     A(I)=0                                                        DYL01450
C                                                                       DYL01460
C  ..  CONSTRUCT COSINE  MATRIX FOR JDIR                                DYL01470
C       OMIT ALL DOF THAT ARE CONSTRAINED TO A MASTER DOF ..            DYL01480
        DO 50 NODE=1,NNODE                                              DYL01490
          ICOS=JTCOS(NODE)                                              DYL01500
          DO 50 K=1,3                                                   DYL01510
            IF (.NOT. BTEST(JTFLG(NODE),K+11)) THEN                     DYL01520
              II=IDOF(NODE,K)                                           DYL01530
              A(II)=A(II)+COSINE(JDIR,K,ICOS)                           DYL01540
            ENDIF                                                       DYL01550
   50     CONTINUE                                                      DYL01560
C                                                                       DYL01570
C.....  MULTIPLY MASS * COSINE  MATRIX FOR JDIR                         DYL01580
C             STORE IN MC(___,JDIR)                                     DYL01590
        DO 60 J=1,NDOF                                                  DYL01600
          MDJ=MD(J)+J                                                   DYL01610
          M  =J-(MD(J+1)-MD(J))+1                                       DYL01620
          DO 60 I=M,J                                                   DYL01630
   60       MC(I,JDIR)=MC(I,JDIR) + MASS(MDJ-I) * A(J)                 DYL01640
        DO 70 I=2,NDOF                                                  DYL01650
          MDI=MD(I)+I                                                   DYL01660
          M  =I-(MD(I+1)-MD(I))+1                                       DYL01670
          DO 70 J=M,I-1                                                 DYL01680
   70       MC(I,JDIR)=MC(I,JDIR) + MASS(MDI-J) * A(J)                 DYL01690
C                                                                       DYL01700
   90 CONTINUE                                                          DYL01710
C                                                                       DYL01720
   91 FORMAT (A)                                                        DYL01730
   92 FORMAT(5X,'DIRECTION OF ACCELERATION:',                           DYL01740
     &       4X,SS,F9.5,' I',SP,F9.5,' J',F9.5,' K  ',SS)               DYL01750
   93 FORMAT (1X,1P,10G12.4)                                            DYL01760
   94 FORMAT (/5X,'INPUT ACCELERATION.',                                DYL01770
     &        '                    (g''s) '/(1X,1P,10G12.4))            DYL01780
C                                                                       DYL01790
C---------------------------------------                                DYL01800
C-   INITIAL GROUND ACCELERATION  ---                                   DYL01810
C---------------------------------------                                DYL01820
      IF (T.GE.TO .AND. T.LE.N*DT ) THEN                                DYL01830
          I=(T-TO)/DT +1                                                DYL01840
          J=I+1                                                         DYL01850
          R=(I*DT-TO-T)/DT                                              DYL01860
          GAT(1)=TX(I)*R                                                DYL01870
          GAT(2)=TY(I)*R                                                DYL01880
          GAT(3)=TZ(I)*R                                                DYL01890
          IF (J.LE.N .AND. R.NE.1 ) THEN                                DYL01900
              GAT(1)=GAT(1) + (1-R)*TX(J)                               DYL01910
              GAT(2)=GAT(2) + (1-R)*TY(J)                               DYL01920
              GAT(3)=GAT(3) + (1 R)*TZ(J)                               DYL01930
          ENDIF                                                         DYL01940
      ELSE                                                              DYL01950
          GAT(1)=0                                                      DYL01960
          GAT(2)=0                                                      DYL01970
          GAT(3)=0                                                      DYL01980
      ENDIF                                                             DYL01990
      GAT(1)=  GAT(1)*GRAV                                              DYL02000
      GAT(2)=  GAT(2)*GRAV                                              DYL02010
      GAT(3)=  GAT(3)*GRAV                                              DYL02020
      RETURN                                                            DYL02030
C                                                                       DYL02040
C---------------------------------------                                DYL02050
C- DETERMIN INCRIMENTAL ACCELERATIONS  ---                              DYL02060
C---------------------------------------                                DYL02070
  200 CONTINUE                                                          DYL02080
C------ DETERMINE THE INCRIMENTAL ACCELERATION BETWEEN TIME T AND TL    DYL02090
      DO 201 I=1,3                                                      DYL02100
  201 GA(I)=0                                                           DYL02110
      T=T                                                               DYL02120
      DT=DT                                                             DYL02130
C                                                                       DYL02140
C..... ...CURRENT ACCELERATION                                          DYL02150
      IF (T.GE.TO .AND. T.LE.N*DT ) THEN                                DYL02160
          I=(T-TO)/DT +1                                                DYL02170
          J=I+1                                                         DYL02180
          R=(I*DT-TO-T)/DT                                              DYL02190
          GA(1)=TX(I)*R                                                 DYL02200
          GA(2)=TY(I)*R                                                 DYL02210
          GA(3)=TZ(I)*R                                                 DYL02220
          IF (J.LE.N .AND. R.NE.1 ) THEN                                DYL02230
              GA(1)=GA(1) + (1-R)*TX(J)                                 DYL02240
              GA(2)=GA(2) + (1-R)*TY(J)                                 DYL02250
              GA(3)=GA(3) + (1-R)*TZ(J)                                 DYL02260
          ENDIF                                                         DYL02270
          TXI=TX(I)                                                     DYL02280
          TXJ=TX(J)                                                     DYL02290
      ENDIF                                                             DYL02300
C                                                                       DYL02310
C ..  ..PREVIOUS ACCELERATION                                           DYL02320
      IF (TL.GE.TO .AND. TL.LE.N*DT ) THEN                              DYL02330
          I=(TL-TO)/DT +1                                               DYL02340
          J=I+1                                                         DYL02350
          R=(I*DT+TO-TL)/DT                                             DYL02360
          GA(1)=GA(1)-TX(I)*R                                           DYL02370
          GA(2)=GA(2)-TY(I)*R                                           DYL02380
          GA(3)=GA(3)-TZ(I)*R                                           DYL02390
          IF (J.LE.N .AND. R.NE.1 ) THEN                                DYL02400
              GA(1)=GA(1) - (1-R)*TX(J)                                 DYL02410
              GA(2)=GA(2)   (1-R)*TY(J)                                 DYL02420
              GA(3)=GA(3)   (1-R)*TZ(J)                                 DYL02430
          ENDIF                                                         DYL02440
          TXI=TX(I)                                                     DYL02450
          TXJ=TX(J)                                                     DYL02460
      ENDIF                                                             DYL02470
C                                                                       DYL02480
C------ MULTIPLY B BY GRAV.                                            DYL02490
      DO 215 K=1,3                                                      DYL02500
  215 GA(K)=  GA(K)*GRAV                                                DYL02510
C                                                                       DYL02520
C- ---- CALC. ACCELERATION VECTOR = -MASS * COSINE * GROUND ACCELERATIONDYL02530
      DO 220 I=1,NDOF                                                   DYL02540
      DO 220 K=1,3                                                      DYL02550
  220   P(I)=P(I) - MC(I,K)*GA(K)                                       DYL02560
C                                                                       DYL02570
 9280 FORMAT(/3X,20X,'     MAXIMUM   AT    TIME      ',1X,              DYL02580
     &'     MINIMUM   AT    TIME          AVERAGE     STANDARD DEV.',1X,DYL02590
     &'      RMS ')                                                     DYL02600
```

```
 9290 FORMAT (3X,A,1P,7G15.5)                                                    DYL02610
C                                                                               DYL02620
      RETURN                                                                    DYL02630
      END                                                                       DYL02640
C-----------------------------------------------------------------------------SOL00010
      SUBROUTINE SOL03                                                          SOL00020
      LOGICAL BTEST                                                             SOL00030
$INCLUDE''ZCOMM'                                                                SOL00040
C                                                                               SOL00050
C       ---- SET UP CONSTANTS ------------------                                SOL00060
        L1=1                                                                    SOL00070
        L2=NCOND                                                                SOL00080
        L3=L2+1                                                                 SOL00090
        L4=NCOND +NFREE                                                         SOL00100
        L5=L4+1                                                                 SOL00110
        L6=NDOF                                                                 SOL00120
        ELSTIC= TRUE.                                                           SOL00130
C                                                                               SOL00140
C       --- INITIALIZE STORAGE                                                  SOL00150
C     Z(IZEVAL) = LOCATION OF THE EIGENVALUES                                   SOL00160
C     Z(IZEVEC) = LOCATION OF THE EIGENVECTORS                                  SOL00170
C     Z(IZA   ) = LOCATION OF THE SYMMETRIC STIFFNESS                           SOL00180
C     Z(IZB   ) = LOCATION OF THE SYMMETRIC MASS OR KG                          SOL00190
C                                                                               SOL00200
        IZEVAL = IZ                                                             SOL00210
        IZ     = IZ+NFREE                                                       SOL00220
        IZEVEC = IZ                                                             SOL00230
        IZ     = IZ+NFREE*NFREE                                                 SOL00240
        IZA    = IZ                                                             SOL00250
        IZ     = IZ+NFREE*NFREE                                                 SOL00260
        IZB    = IZ                                                             SOL00270
        IZ     = IZ+NFREE*NFREE                                                 SOL00280
        IZTMP  = IZ                                                             SOL00290
        IZ     = IZ+NFREE                                                       SOL00300
        IZTMP2 = IZ                                                             SOL00310
        IZ     = IZ+NDOF                                                        SOL00320
        IW     = IZ                                                             SOL00330
C                                                                               SOL00340
      CALL CKSTOR(NDOF,IW)                                                      SOL00350
C                                                                               SOL00360
      IMD  =NZ(IZMD  -NDOF)                                                     SOL00370
      IMDMS=NZ(IZMDMS-NDOF)                                                     SOL00380
      IMDKG=NZ(IZMDKG-NDOF)                                                     SOL00390
C                                                                               SOL00400
C---------  FORM MASS MATRIX -----                                              SOL00410
C                                                                               SOL00420
      DO 400 I=1,IMDMS                                                          SOL00430
  400     Z(I+IZMASS-1)=0.                                                      SOL00440
C                                                                               SOL00450
      CALL MFORM(2,INMASS,NMASS,Z(IZMASS),NZ(IZMDMS),MD,NDOF,IMDMS,             SOL00460
     &      NNODE,MAXNOD,BTEST(IBUG,5),IBUG,NZ(IZID),NZ(IZIDOF),               SOL00470
     &      Z(IZCNST),NZ(IZJCOS),Z(IZMDAT),Z(IW),NZ(IW+21),Z(IW+27),           SOL00480
     &      Z(IW+ 63),Z(IW+ 99),TITLE)                                         SOL00490
C                                                                               SOL00500
      CALL SOL03A                                                               SOL00510
     & (L1,L2,L3,L4,L5,L6,IMD,IMDMS,IMDKG,                                      SOL00520
     & KGCOND   ,MCOND    , Z(IZKGDT),NZ(IZKGDT+3),                             SOL00530
     & NDOF     , NCOND    , NFREE    , TITLE    , STEPID   ,                  SOL00540
     & Z(IZSTIF), Z(IZMASS), Z(IZKG   ), Z(IZA   ), Z(IZB   ),                  SOL00550
     & NZ(IZMD  ),NZ(IZMDMS),NZ(IZMDKG), Z(IW    ), Z(IZTMP ),                 SOL00560
     & NNODE    , NCOS     , MAXNOD   ,IBUG     , Z(IZTMP2),                   SOL00570
     & NZ(ISID  ),NZ(IZIDOF), Z(IZCNST),NZ(IZJFLG), Z(IZCORD),                  SOL00580
     & Z(IZCOS ),NZ(IZJCOS), Z(IZEVAL), Z(IZEVEC) )                             SOL00590
C                                                                               SOL00600
      RETURN                                                                    SOL00610
      END                                                                       SOL00620
C                                                                               JOID0010
C-----------------------------------------------------------------------------JOID0020
C                                                                               JOID0030
      SUBROUTINE JOIDSP(IOPT,MAXNOD,NDOF,NNODE,NLOAD,NF,ID,IDOF,                JOID0040
     &            CONST,DISP,TITLE,NAME,STEPID,HEAD,NCOS,COSINE,JTCOS)          JOID0050
      CHARACTER*(*) TITLE(2),STEPID,NAME                                        JOID0060
      CHARACTER*15 CD(6),EQUAL*60                                               JOID0070
      LOGICAL HEAD                                                              JOID0080
      DIMENSION IDOF(MAXNOD,6),CONST(MAXNOD,6)                                  JOID0090
      DIMENSION DISP(NDOF,NLOAD),ID(MAXNOD)                                     JOID0100
      DIMENSION D(6),G(6),COSINE(3,3,NCOS),JTCOS(MAXNOD)                        JOID0110
      DATA EQUAL                                                                JOID0120
     & /'------------------------------------------------------------'/JOID0130
C                                                                               JOID0140
      NLEN=MIN(LEN(NAME)+1,60)                                                  JOID0150
      NC=1                                                                      JOID0160
C                                                                               JOID0170
C-----------------------------                                                  JOID0180
C- LOOP FOR EACH LOAD CASE  ---                                                 JOID0190
C-----------------------------                                                  JOID0200
C                                                                               JOID0210
      DO 300 L=1,NLOAD                                                          JOID0220
        NDSPLS=0                                                                JOID0230
        IF (.NOT.(NCOS.EQ.1 .AND. COSINE(1,1,1).EQ.1 .AND.                      JOID0240
     &       COSINE(2,2,1).EQ.1  .AND. COSINE(3,3,1).EQ.1 )                     JOID0250
     &       .AND. IOPT.EQ.2 ) GO TO 210                                        JOID0260
C                                                                               JOID0270
C-----------------------------                                                  JOID0280
C- PRINT GLOBAL DISPLACEMENTS ---                                               JOID0290
C-----------------------------                                                  JOID0300
C  NOTE:DISP MAY CONTAIN DISPLACEMENTS, VELOCITIES, ACCELERATIONS OR            JOID0310
C       EIGENVALUES.                                                            JOID0320
C       ALL OF WHICH ARE PRINTED OUT AT EACH NODE....                           JOID0330
C                                                                               JOID0340
        IF (HEAD) WRITE(108,310) TITLE(1),TITLE(2)                              JOID0350
        IF (IOPT.NE.2) WRITE(108,315) NAME,L,STEPID,EQUAL(1:NLEN)               JOID0360
        IF (IOPT.EQ.2) WRITE(108,415) NAME,          EQUAL(1:NLEN)              JOID0370
        DO 100 I=1,NNODE                                                        JOID0380
C                                                                               JOID0390
C... . . .  TRANSFER DISPL TO LOCAL VECTOR   .... . ....                       JOID0400
          DO 20 J=1,6                                                           JOID0410
            IJ=IDOF(I,J)                                                        JOID0420
   20       D(J)=DISP(IJ,L)                                                     JOID0430
C                                                                               JOID0440
C..... . . . ADD ROTATIONS TO DISPL FOR CONSTRAINTS                             JOID0450
C... . .... . IF UNCONSTRAINED, CONST(I,J)=0                                    JOID0460
          D(1) = D(1) + D(5)*CONST(I,3) + D(6)*CONST(I,5)                       JOID0470
          D(2) = D(2) + D(4)*CONST(I,1) + D(6)*CONST(I,6)                       JOID0480
          D(3) = D(3) + D(4)*CONST(I,2) + D(5)*CONST(I,4)                       JOID0490
C                                                                               JOID0500
C. ..... . .  TRANSFER TO GLOBAL DISPLACEMENTS .....                           JOID0510
          JCOS=JTCOS(I)                                                         JOID0520
          DO 25 I1=1,3                                                          JOID0530
            I2=I1+3                                                             JOID0540
            G(I1)=0                                                             JOID0550
            G(I2)=0                                                             JOID0560
            DO 25 J1=1,3                                                        JOID0570
              J2=J1+3                                                           JOID0580
              G(I1)=G(I1)+COSINE(J1,I1,JCOS)*D(J1)                              JOID0590
   25         G(I2)=G(I2)+COSINE(J1,I1,JCOS)*D(J2)                            JOID0600
C                                                                               JOID0610
C..  . . . .  WRITE DISPL                                                       JOID0620
          DO 30 J=1,6                                                           JOID0630
   30     WRITE(CD(J),9) G(J)                                                   JOID0640
    9     FORMAT(1P,G14.6,' ')                                                  JOID0650
          WRITE(108,220) ID(I),(CD(K),K=1,6)                                    JOID0660
C                                                                               JOID0670
          NDSPLS=NDSPLS+1                                                       JOID0680
  100 CONTINUE                                                                  JOID0690
C                                                                               JOID0700
      IF (NCOS.EQ.1 .AND. COSINE(1,1,1).EQ.1 .AND. COSINE(2,2,1) EQ.1           JOID0710
     &              .AND. COSINE(3,3,1).EQ.1 ) GO TO 300                        JOID0720
C                                                                               JOID0730
C-----------------------------                                                  JOID0740
C- PRINT JOINT DISPLACEMENTS ---                                                JOID0750
C-----------------------------                                                  JOID0760
C  NOTE:DISP MAY CONTAIN DISPLACEMENTS, VELOCITIES, ACCELERATIONS OR            JOID0770
C       EIGENVALUES                                                             JOID0780
C       ALL OF WHICH ARE PRINTED OUT AT EACH NODE. .                            JOID0790
C                                                                               JOID0800
  210   IF (HEAD .AND. NDSPLS.GT.20) WRITE(108,310) TITLE(1),TITLE(2)           JOID0810
        IF (IOPT.NE.2) WRITE(108,316) NAME,L,STEPID,EQUAL(1:NLEN)               JOID0820
        IF (IOPT.EQ.2) WRITE(108,416) NAME,          EQUAL(1:NLEN)              JOID0830
        DO 200 I=1,NNODE                                                        JOID0840
C                                                                               JOID0850
C. ... ..... TRANSFER GLOBAL DISPL TO LOCAL VECTOR .  ..... ...                JOID0860
          DO 219 J=1,6                                                          JOID0870
            IJ=IDOF(I,J)                                                        JOID0880
  219       D(J)=DISP(IJ,L)                                                     JOID0890
C                                                                               JOID0900
C............ ADD ROTATIONS TO DISPL FOR CONSTRAINTS                            JOID0910
C. ..... . . IF UNCONSTRAINED, CONST(I,J)=0                                     JOID0920
          D(1) = D(1) + D(5)*CONST(I,3) + D(6)*CONST(I,5)                       JOID0930
          D(2) = D(2) + D(4)*CONST(I,1) + D(6)*CONST(I,6)                       JOID0940
          D(3) = D(3) + D(4)*CONST(I,2) + D(5)*CONST(I,4)                       JOID0950
C .                                                                             JOID0960
C.  ....... WRITE DISPL                                                         JOID0970
          DO 230 J=1,6                                                          JOID0980
            WRITE(CD(J),9) D(J)                                                 JOID0990
            IJ=IDOF(I,J)                                                        JOID1000
  230       IF (IJ.GT.NF) CD(J)(15:15)='*'                                      JOID1010
        WRITE(108,221) ID(I),(CD(K),K=1,6),JTCOS(I)                             JOID1020
C                                                                               JOID1030
  200 CONTINUE                                                                  JOID1040
      WRITE(108,320)                                                            JOID1050
  300 CONTINUE                                                                  JOID1060
C                                                                               JOID1070
   10 FORMAT (1P,G15.7)                                                         JOID1080
  220 FORMAT (5X,I5,6A)                                                         JOID1090
  221 FORMAT (5X,I5,6A,I10)                                                     JOID1100
  310 FORMAT ('1 STRUCTURE.. .. ',A,/'  SOLUTION. .. . ',A)                    JOID1110
  315 FORMAT (//2X, 'GCS ',A,',  LOADING #',I5,5X,A, /                         JOID1120
     &        1X,'------',A//6X,'NODE',7X,'DX',13X,'DY',13X,'DZ',              JOID1130
     &        13X,'RX',13X,'RY',13X,'RZ'/)                                     JOID1140
  316 FORMAT (//2X, 'JCS ',A,'  LOADING #',I5,5X,A, /1X,'------',A            JOID1150
     &        //6X,'NODE',7X,'DX',13X,'DY',13X,'DZ',13X,'RX',                  JOID1160
     &        13X,'RY',13X,'RZ',6X,'  COSINE #'/)                              JOID1170
  415 FORMAT (//2X, 'GCS ',A/1X,'------',A/25X,                                JOID1180
     &      'NOTE: MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'//6X,           JOID1190
     &  'NODE',7X,'DX',13X,'DY',13X,'DZ',13X,'RX',13X,'RY',13X,'RZ'/)           JOID1200
  416 FORMAT (//2X, 'JCS ',A/1X,'------',A/25X,                                JOID1210
     &      'NOTE: MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'//6X,           JOID1220
     &  'NODE',7X,'DX',13X,'DY',13X,'DZ',13X,'RX',13X,'RY',13X,'RZ',           JOID1230
     &  6X,'  COSINE #'/)                                                       JOID1240
  320 FORMAT (4X,'NOTE: * DENOTES A RESTRAINED DEGREE OF FREEDOM')              JOID1250
C                                                                               JOID1260
      RETURN                                                                    JOID1270
      END                                                                       JOID1280
C-----------------------------------------------------------------------------SOL00010
      SUBROUTINE SOL03A                                                         SOL00020
     & (L1,L2,L3,L4,L5,L6,  IMD,IMDMS,IMDKG,                                    SOL00030
     &  KGCOND   ,MCOND    , KGDATA   ,KGFORM    ,                            SOL00040
     &  NDOF     , NCOND    , NFREE    , TITLE    , STEPID   ,                 SOL00050
     &  STIFF    , MASS     , KG       , A        , B        ,                 SOL00060
     &  MD       , MDMS     , MDKG     , WORK     , TMP      ,                 SOL00070
     &  NNODE    , NCOS     , MAXNOD   , IBUG     , TMP2     ,                 SOL00080
     &  ID       , IDOF     , CNST     , JTFLG    , COORD    ,                 SOL00090
     &  COSINE   , JTCOS    , EVAL     , EVEC     )                           SOL00100
      LOGICAL TEST,BTEST,MCOND,KGCOND,MCOND1,KCOND1,PRINS                       SOL00110
      REAL MASS, KG, KGDATA                                                     SOL00120
      CHARACTER*40 OPTION                                                       SOL00130
      CHARACTER*(*) TITLE(2),STEPID                                             SOL00140
      DIMENSION STIFF(IMD),MASS(IMDMS),KG(IMDKG),G(6),D(6)                      SOL00150
      DIMENSION A(NFREE,NFREE),B(NFREE,NFREE),CNST(MAXNOD,6)                    SOL00160
      DIMENSION MD(L6+1),MDMS(L6+1),MDKG(L6+1),WORK(L6),TMP(L6)                 SOL00170
      DIMENSION EVAL(NFREE),EVEC(NFREE,NFREE),SUMRCT(6),TMP2(NFREE)             SOL00180
      DIMENSION COORD(MAXNOD,3),ID(MAXNOD),IDOF(MAXNOD,6),                      SOL00190
     &          COSINE(3,3,NCOS),JTFLG(MAXNOD),JTCOS(MAXNOD)                    SOL00200
C                                                                               SOL00210
C---------------                                                                SOL00220
C- INPUT DATA ---                                                               SOL00230
C---------------                                                                SOL00240
      READ(105,*) OPTION,IPRT,IPRT2                                             SOL00250
      PRINS=BTEST(IBUG,5)                                                       SOL00260
C                                                                               SOL00270
      IF (TEST(OPTION,'FREQ',.FALSE.)) THEN                                     SOL00280
        IOPT=1                                                                  SOL00290
        OPTION='NATURAL FREQUENCIES'                                            SOL00300
      ELSE IF (TEST(OPTION,'BUCK',.FALSE.)) THEN                                SOL00310
        IOPT=2                                                                  SOL00320
        OPTION='ELASTIC BUCKLING LOAD'                                          SOL00330
      ELSE                                                                      SOL00340
        WRITE(108,*) 'SOL-03 INVALID OPTION: ',OPTION                           SOL00350
        RETURN                                                                  SOL00360
      ENDIF                                                                     SOL00370
C                                                                               SOL00380
      IF (IPRT.LE.0) IPRT=NFREE                                                 SOL00390
      IPRT=MIN(NFREE,IPRT)                                                      SOL00400
      IPRT2=MIN(IPRT2,IPRT)                                                     SOL00410
C                                                                               SOL00420
      WRITE(108,10) TITLE(1),TITLE(2),OPTION,IPRT                               SOL00430
   10 FORMAT ('1 STRUCTURE..... ',A,/'  SOLUTION.....  ',A,//                   SOL00440
     &  1X,' SOLUTION #3, DETERMINE EIGENVALUES AND EIGENVECTORS ',/            SOL00450
     &  1X,'-----------------------------------------------------'//            SOL00460
     &  1X,' OPTION:.....',A/                                                   SOL00470
     &  1X,' NUMBER OF EIGENVALUES:.....',I5)                                   SOL00480
C                                                                               SOL00490
C-------------------------                                                      SOL00500
C- FORMULATE STIFFNESS ----                                                     SOL00510
C-------------------------                                                      SOL00520
      I1=1                                                                      SOL00530
      CALL FORM(I1)                                                             SOL00540
      IF (PRINS) CALL LMATRX(STIFF,MD,NDOF ,IMD,                                SOL00550
     &                       'GLOBAL STIFFNESS MATRIX')                         SOL00560
C                                                                               SOL00570
C-----------------------------                                                  SOL00580
C- FORM GEOMETRIC STIFFNESS ---                                                 SOL00590
C-----------------------------                                                  SOL00600
C                                                                               SOL00610
C     FORMULATE SEPERATE GEOMETRIC STIFFNESS MATRIX                             SOL00620
      IF (IOPT.EQ.2) THEN                                                       SOL00630
        KGFORM=2                                                                SOL00640
        KGDATA=0                                                                SOL00650
        I2=2                                                                    SOL00660
        CALL FORM(I2)                                                           SOL00670
        IF (PRINS) CALL LMATRX(KG,MDKG,NDOF,IMDKG,                              SOL00680
     &                         'GLOBAL GEOMETRIC STIFFNESS MATRIX')             SOL00690
      ELSE                                                                      SOL00700
        IF (PRINS) CALL LMATRX(MASS,MDMS,NDOF,IMDMS,                            SOL00710
     &                         'GLOBAL MASS MATRIX')                            SOL00720
      ENDIF                                                                     SOL00730
C                                                                               SOL00740
C-----------------------------                                                  SOL00750
C- CONDENSE OUT NON-FREE DOF ---                                                SOL00760
C-----------------------------                                                  SOL00770
C. . IF MCOND IS TRUE THE MASS MATRIX IS ALSO REDUCED BY GUYAN REDUCTION        SOL00780
C .. IF KGCOND IS TRUE THEN KG IS ALSO REDUCED                                  SOL00790
      IF (NCOND.GT.0) THEN                                                      SOL00800
        I1=1                                                                    SOL00810
        IF (IOPT.EQ.2) THEN                                                     SOL00820
          MCOND1 =.FALSE.                                                       SOL00830
          KCOND1 =KGCOND                                                        SOL00840
        ELSE IF (IOPT.EQ.1) THEN                                                SOL00850
          XCOND1 =.FALSE.                                                       SOL00860
          MCOND1 =MCOND                                                         SOL00870
        ENDIF                                                                   SOL00880
C SET UP DUMMY ZERO LOAD VECTOR                                                 SOL00890
        DO 28 I=1,NDOF                                                          SOL00900
   28     TMP(I)=0                                                             SOL00910
        CALL MSOLL(1,BTEST(IBUG,6),L4,IMD,NDOF,1,L1,L2,MD,STIFF,TMP,            SOL00920
     &             WORK,MCOND1,MASS,MDMS,IMDMS,KCOND1,KG,MDKG,IMDKG)            SOL00930
      ENDIF                                                                     SOL00940
C                                                                               SOL00950
C-----------------------------------------                                      SOL00960
C- TRANSFER STIFFNESS INTO A MATRIX ---                                         SOL00970
C-----------------------------------------                                      SOL00980
```

```
C  NOTE. IMSL ROUTINE GVCSP IS USED TO SOLVE FOR EIGENVALUES,
C       THIS ROUTINE USES A 'FULL SYMMETRIC MATRIX'.
C
C  NOTE. THE STORAGE METHOD USED FOR STIFFNESS, MASS AND
C       GEOMETRIC STIFFNESS ARE DIFFERENT
C
      DO 600 J=L3,L4,-1
        DO 600 I=J,L3,-1
           IJ=MD(J)+J-I
           IF(IJ .LT. MD(J+1) .AND. IJ .GE. MD(J) ) THEN
              A(I-L3+1,J-L3+1)=STIFF(IJ)
              A(J-L3+1,I-L3+1)=A(I-L3+1,J-L3+1)
           ELSE IF( IJ .GE. MD(J+1) ) THEN
              A(I-L3+1,J-L3+1)=0
              A(J-L3+1,I-L3+1)=0
           ENDIF
C
        IF (PRIN5) WRITE(108,29) I,J,A(I,J)
   29   FORMAT (' I:',I5,' J:',I5,' A(I,J)',1P,G15 5)
C
C--- ... . CHECK FOR ZERO'S ON MAIN DIAGONAL
        IF (I.EQ.J .AND. A(I-L3+1,J-L3+1).LE.0 ) THEN
           WRITE(108,32) I,A(I-L3+1,J-L3+1)
           STOP
        ENDIF
C
  600  CONTINUE
   32 FORMAT (5X,'DOF #',I5,' IS .LE  0 IN THE STIFFNESS MATRIX. .',
     &        5X,'K(I,I)=',1P,G12 4,'  SOLN IS ABORTED')
C
C-------------------------------------------------------
C- TRANSFER MASS OR GEOMETRIC STIFFNESS INTO B MATRIX ---
C-------------------------------------------------------
      IF (IOPT EQ 1) THEN
      DO 700 J=L3,L4,-1
        DO 700 I=J,L3,-1
           IJ=MDMS(J)+J-I
           IF(IJ .LT. MDMS(J+1) .AND. IJ .GE. MDMS(J) ) THEN
              B(I-L3+1,J-L3+1)=MASS(IJ)
              B(J-L3+1,I-L3+1)=B(I-L3+1,J-L3+1)
           ELSE IF( IJ .GE. MDMS(J+1) ) THEN
              B(I-L3+1,J-L3+1)=0
              B(J-L3+1,I-L3+1)=0
           ENDIF
        IF (PRIN5) WRITE(108,39) I,J,B(I,J)
   39   FORMAT (' I:',I5,' J:',I5,' B(I,J) ', 1P,G15.5)
C
C--------- CHECK FOR ZERO'S ON MAIN DIAGONAL
        IF (I EQ J .AND. B(I-L3+1,J-L3+1).LE.0 ) THEN
           WRITE(108,42) I,B(I-L3+1,J-L3+1)
           STOP
        ENDIF
C
  700 CONTINUE
   42 FORMAT (5X,'DOF #',I5,' IS .LE  0 IN THE MASS MATRIX...',
     &        5X,'M(I,I)=',1P,G12.4,'  SOLN IS ABORTED')
C
      ELSE
      DO 800 J=L3,L4,-1
        DO 800 I=J,L3,-1
           IJ=MDKG(J)+J-I
           IF(IJ .LT. MDKG(J+1) AND. IJ .GE. MDKG(J) ) THEN
              B(I-L3+1,J-L3+1)=KG(IJ)
              B(J-L3+1,I-L3+1)=B(I-L3+1,J-L3+1)
           ELSE IF( IJ .GE. MDKG(J+1) ) THEN
              B(I-L3+1,J-L3+1)=0
              B(J-L3+1,I-L3+1)=0
           ENDIF
        IF (PRIN5) WRITE(108,49) I,J,B(I,J)
   49   FORMAT (' I ',I5,' J:',I5,' B(I,J):', 1P,G15.5)
C
C--------- CHECK FOR ZERO'S ON MAIN DIAGONAL
        IF (I.EQ.J .AND. B(I-L3+1,J-L3+1).LE.0 ) THEN
           WRITE(108,42) I,B(I-L3+1,J-L3+1)
           STOP
        ENDIF
C
  800 CONTINUE
   52 FORMAT (5X,'DOF #',I5,
     &        ' IS .LE  0 IN THE GEOMETRIC STIFFNESS MATRIX.. ',
     &        5X,'KG(I,I)=',1P,G12.4,'  SOLN IS ABORTED')
C
      ENDIF
C
C-----------------------------------------
C- SOLVE FOR EIGENVALUES AND EIGENVECTORS ---
C-----------------------------------------
C---- ...  IMSL SU_BROUTINE GVCSP ------------------------------------ SOL01860
C  EIGENVALUES AND EIGENVECTORS OF A*X-LAMBDA*B*X=0                    SOL01870
C  A  = STIFFNESS - REAL, FULL SYMMETRIC MATRIX                        SOL01880
C  B  = MASS OR GEOMETRIC STIFFNESS - REAL, FULL SYMMETRIC MATRIX      SOL01890
C  N  = NUMBER OF DOF OF A AND B                                       SOL01900
C  EVAL= EIGENVALUES (LAMBDA), N BY 1                                  SOL01910
C  EVEC= EIGENVECTORS, N BY N                                          SOL01920
C-------------------------------------------------------------- ------ SOL01930
C                                                                      SOL01940
C                                                                      SOL01950
      CALL GVCSP(NFREE,A,B,EVAL,EVEC,WORK,TMP2,TMP)                    SOL01960
C-----------------------------------------                             SOL01970
C- NORMALIZE MODE SHAPE TO MAX VALUE OF ONE ---                        SOL01980
C-----------------------------------------                             SOL01990
C                                                                      SOL02000
C NOTE: THE EIGENVECTOR IN IMSL GVCSP HAS BEEN NORMALIZED              SOL02010
C                                                                      SOL02020
C-------------------------------------                                 SOL02030
C- PRINT EIGENVALUES AND EIGENVECTORS ---                              SOL02040
C-------------------------------------                                 SOL02050
      IST =1                                                           SOL02060
  100 IEND=MIN(IST+6,IPRT)                                             SOL02070
C                                                                      SOL02080
      WRITE(108,110) (I,I=IST,IEND)                                    SOL02090
      IF (IOPT.EQ 2) THEN                                              SOL02100
         WRITE(108,120) (EVAL(I),I=IST,IEND)                           SOL02110
      ELSE IF (IOPT.EQ.1) THEN                                         SOL02120
         DO 125 I=IST,IEND                                             SOL02130
  125      EVAL(I)=SQRT( EVAL(I) )                                     SOL02140
         WRITE(108,126) (EVAL(I),I=IST,IEND)                           SOL02150
         DO 127 I=IST,IEND                                             SOL02160
  127      EVAL(I)= EVAL(I)  / 6.2831853                               SOL02170
         WRITE(108,128) (EVAL(I),I=IST,IEND)                           SOL02180
         DO 129 I=IST,IEND                                             SOL02190
           IF (EVAL(I).EQ.0) EVAL(I)=1.E-15                            SOL02200
  129      EVAL(I)= 1.00 / EVAL(I)                                     SOL02210
         WRITE(108,130) (EVAL(I),I=IST,IEND)                           SOL02220
      ENDIF                                                            SOL02230
      WRITE(108,135)                                                   SOL02240
  110 FORMAT('          MODE:',7(I10,5X))                              SOL02250
  120 FORMAT('      EIGENVALUE :',1P,7(G14 5,1X) )                     SOL02260
  126 FORMAT(' FREQ (RAD/SEC):',1P,7(G14 5,1X) )                       SOL02270
  128 FORMAT(' FREQUENCY (HZ):',1P,7(G14.5,1X) )                       SOL02280
  130 FORMAT('   PERIOD (SEC):',1P,7(G14.5,1X) )                       SOL02290
  135 FORMAT('   EIGENVECTORS:')                                       SOL02300
C                                                                      SOL02310
C-------- EIGENVECTORS                                                 SOL02320
      DO 140 I=1,NFREE                                                 SOL02330
        II=I-NCOND                                                     SOL02340
  140   WRITE(108,150) II,(EVEC(I,J),J=IST,IEND)                       SOL02350
  150 FORMAT('      DOF(',I6,')',1P,7(G14.5,1X) )                      SOL02360
C                                                                      SOL02370
      IF (IEND.LT.IPRT ) THEN                                          SOL02380
        WRITE(108,10) TITLE(1),TITLE(2),OPTION,IPRT                    SOL02390
        IST=IEND+1                                                     SOL02400
        GO TO 100                                                      SOL02410
      ENDIF                                                            SOL02420
C                                                                      SOL02430
C--------------------------------------                                SOL02440
```

```
C- CALC EIGEN VECTORS FOR NONDYNAMIC DOF ---                           SOL02450
C-------------------------------------------                           SOL02460
C                                                                      SOL02470
C------ TRANSFER EIGENVECTORS TO TEMP STORAGE                          SOL02480
C                                                                      SOL02490
      DO 500 IFREQ=1,IPRT2                                             SOL02500
C                                                                      SOL02510
C------ TRANSFER EIGENVECTORS TO TEMP STORAGE                          SOL02520
        DO 200 J=L1,L2                                                 SOL02530
  200     WORK(J)=0                                                    SOL02540
        DO 210 J=L3,L4                                                 SOL02550
  210     WORK(J)=EVEC(J-L2,IFREQ)                                     SOL02560
        DO 220 J=L5,L6                                                 SOL02570
  220     WORK(J)=0                                                    SOL02580
C                                                                      SOL02590
C--- -- CALC EIGENVECTORS FOR CONDENSED OUT DOF                        SOL02600
        CALL REACTN(I1,BTEST(IBUG,6),NNODE,L1,L2,L3,L4,TMP,WORK,MD,    SOL02610
     &              STIFF,IMD,NDOF,1,1.,MAXNOD,IDOF,COORD,ID,TITLE,    SOL02620
     &              STEPID,.FALSE.,NCOS,COSINE,JTCOS,JTPLO,SUMRCT)     SOL02630
        CALL MSOLL(3,BTEST(IBUG,6),L4,IMD,NDOF,I1,L1,L2,MD,STIFF,TMP,  SOL02640
     &             WORK, FALSE.,MASS,MDMS,1,.FALSE.,KG  ,MDKG,1)       SOL02650
C                                                                      SOL02660
C------- NORMALIZE MODE SHAPE TO MAX VALUE OF ONE                      SOL02670
        AMIN=0                                                         SOL02680
        AMAX=0                                                         SOL02690
        DO 230 I=1,L6                                                  SOL02700
          AMIN=MIN(AMIN,WORK(I))                                       SOL02710
  230     AMAX=MAX(AMAX,WORK(I))                                       SOL02720
        IF (-AMIN.GT AMAX) AMAX=-AMIN                                  SOL02730
C                                                                      SOL02740
        DO 240 I=1,L6                                                  SOL02750
  240     WORK(I)=WORK(I)/AMAX                                         SOL02760
C                                                                      SOL02770
C------- PRINT EIGENVALUE AT EACH NODE                                 SOL02780
        IF (IOPT.EQ.2) WRITE(STEPID,250) IFREQ,EVAL(IFREQ)             SOL02790
          IF (EVAL(IFREQ) EQ.0) EVAL(IFREQ)=1 E-15                     SOL02800
        FREQ=1.00/EVAL(IFREQ)                                          SOL02810
        IF (IOPT EQ.1) WRITE(STEPID,260) IFREQ,FREQ                    SOL02820
C                                                                      SOL02830
        CALL JOIDSP(1,MAXNOD,NDOF,NNODE,1,L4,ID,IDOF,CNST,WORK,        SOL02840
     &              TITLE,'EIGENVECTORS',STEPID,.TRUE.,NCOS,           SOL02850
     &              COSINE,JTCOS)                                      SOL02860
  250   FORMAT ('MODE #',I3,'   EIGENVALUE:',1P,G15.6)                 SOL02870
  260   FORMAT ('MODE #',I3,'   FREQUENCY:',1P,G15.6)                  SOL02880
C                                                                      SOL02890
C------- PRINT EIGENVALUE TO DISK FOR PLOTTING                         SOL02900
C                                                                      SOL02910
      IF(BTEST(IBUG,2)) THEN                                           SOL02920
      DO 930 I=1,NNODE                                                 SOL02930
C. ........... TRANSFER EIGENVECTOR TO LOCAL VECTOR  ..... .          SOL02940
        DO 920 J=1,6                                                   SOL02950
          IJ=IDOF(I,J)                                                 SOL02960
  920     D(J)=WORK(IJ)                                                SOL02970
C                                                                      SOL02980
C.. ......... ADD ROTATION EIGENVECTORS TO DISPL FOR CONSTRAINTS       SOL02990
C.. .... .... IF UNCONSTRAINED, CNST(I,J)=0                            SOL03000
        D(1) = D(1) + D(5)*CNST(I,3) + D(6)*CNST(I,5)                  SOL03010
        D(2) = D(2) + D(4)*CNST(I,1) + D(6)*CNST(I,6)                  SOL03020
        D(3) = D(3) + D(4)*CNST(I,2) + D(5)*CNST(I,4)                  SOL03030
C                                                                      SOL03040
C ... ... ... TRANSFER TO GLOBAL DISPLACEMENTS .....                  SOL03050
        JCOS=JTCOS(I)                                                  SOL03060
        DO 925 I1=1,3                                                  SOL03070
          I2=I1+3                                                      SOL03080
          G(I1)=0                                                      SOL03090
          G(I2)=0                                                      SOL03100
          DO 925 J1=1,3                                                SOL03110
            J2=J1+3                                                    SOL03120
            G(I1)=G(I1)+COSINE(J1,I1,JCOS)*D(J1)                       SOL03130
  925       G(I2)=G(I2)+COSINE(J1,I1,JCOS)*D(J2)                       SOL03140
C                                                                      SOL03150
C. ........... WRITE DISPL                                            SOL03160
        WRITE(10+IFREQ,926) I, (G(K),K=1,6)                            SOL03170
  926   FORMAT (I7,1P,6G12.4)                                          SOL03180
  930 CONTINUE                                                         SOL03190
      ENDIF                                                            SOL03200
C                                                                      SOL03210
  500 CONTINUE                                                         SOL03220
C                                                                      SOL03230
      RETURN                                                           SOL03240
      END                                                              SOL03250
C                                                                      GVCS0010
C  ..................................... ...............              GVCS0020
C                                                                      GVCS0030
      SUBROUTINE GVCSP(N,A,B,D,X,DL,C1,C2)                             GVCS0040
      DIMENSION A(N,N),B(N,N),D(N),X(N,N),DL(N),C1(N),C2(N)           GVCS0050
      CALL REDUC1(N,A,B,DL,FAIL)                                       GVCS0060
      DO 20 I=2,N                                                      GVCS0070
      DO 20 J=1,I-1                                                    GVCS0080
20    A(J,I)=A(I,J)                                                    GVCS0090
      CALL JACOBI(N,A,D,X,C1,C2)                                       GVCS0100
      CALL REBAC1(N,B,DL,X)                                            GVCS0110
      DO 200 I=1,N                                                     GVCS0120
       IP1=I+1                                                         GVCS0130
       K=I                                                             GVCS0140
       P=D(I)                                                          GVCS0150
       IF (N.GT.I) THEN                                                GVCS0160
        DO 120 J=IP1,N                                                 GVCS0170
         IF (D(J).GE.P) GOTO 120                                       GVCS0180
         K=J                                                           GVCS0190
         P=D(J)                                                        GVCS0200
120     CONTINUE                                                       GVCS0210
       ENDIF                                                           GVCS0220
       IF (K.EQ.I) GOTO 200                                            GVCS0230
       D(K)=D(I)                                                       GVCS0240
       D(I)=P                                                          GVCS0250
       DO 140 J=1,N                                                    GVCS0260
        P=X(J,I)                                                       GVCS0270
        X(J,I)=X(J,K)                                                  GVCS0280
        X(J,K)=P                                                       GVCS0290
140    CONTINUE                                                        GVCS0300
200   CONTINUE                                                         GVCS0310
      RETURN                                                           GVCS0320
      END                                                              GVCS0330
C                                                                      REDU0010
C -------- ---------------------------------------                     REDU0020
C                                                                      REDU0030
      SUBROUTINE REDUC1(N,A,B,DL,FAIL)                                 REDU0040
      DIMENSION A(N,N),B(N,N),DL(N)                                    REDU0050
      DO 110 I=1,N                                                     REDU0060
      DO 110 J=I,N                                                     REDU0070
       X=B(I,J)                                                        REDU0080
       I1=I-1                                                          REDU0090
       IF (I1) 120,120,130                                             REDU0100
130    DO 140 K=1,I1                                                   REDU0110
140    X=X-B(I,K)*B(J,K)                                               REDU0120
120    IF (I-J) 150,160,150                                            REDU0130
160    FAIL=X                                                          REDU0140
       IF (X) 400,400,170                                              REDU0150
170    DL(I)=SQRT(X)                                                   REDU0160
       GOTO 110                                                        REDU0170
150    B(J,I)=X/DL(I)                                                  REDU0180
110   CONTINUE                                                         REDU0190
200   DO 210 I=1,N                                                     REDU0200
      DO 210 J=I,N                                                     REDU0210
       X=A(I,J)                                                        REDU0220
       I1=I-1                                                          REDU0230
       IF (I1) 220,220,230                                             REDU0240
230    DO 240 K=1,I1                                                   REDU0250
240    X=X-B(I,K)*A(J,K)                                               REDU0260
220    A(J,I)=X/DL(I)                                                  REDU0270
210   CONTINUE                                                         REDU0280
300   DO 310 J=1,N                                                     REDU0290
      DO 310 I=J,N                                                     REDU0300
       X=A(I,J)                                                        REDU0310
       I1=I-1                                                          REDU0320
```

```
      IF (I1-J) 320,330,330                                             REDU0330
330   DO 340 K=J,I1                                                     REDU0340
340   X=X-A(K,J)*B(I,K)                                                 REDU0350
320   J1=J-1                                                            REDU0360
      IF (J1) 350,350,360                                               REDU0370
360   DO 370 K=1,J1                                                     REDU0380
370   X=X-A(J,K)*B(I,K)                                                 REDU0390
350   A(I,J)=X/DL(I)                                                    REDU0400
310   CONTINUE                                                          REDU0410
400   RETURN                                                            REDU0420
      END                                                               REDU0430
C                                                                       REBA0010
C ------------- ------- ---------------- ---                            REBA0020
C                                                                       REBA0030
      SUBROUTINE REBAC1(N,B,DL,Z)                                       REBA0040
      DIMENSION B(N,N),DL(N),Z(N,N)                                     REBA0050
      DO 100 J=1,N                                                      REBA0060
       P=1.E-15                                                         REBA0070
       DO 60 I=N,1,-1                                                   REBA0080
        X=Z(I,J)                                                        REBA0090
        I1=I+1                                                          REBA0100
        IF ((I1-N).GT.0) GOTO 50                                        REBA0110
        DO 40 K=I1,N                                                    REBA0120
40      X=X-B(K,I)*Z(K,J)                                               REBA0130
50      Z(I,J)=X/DL(I)                                                  REBA0140
        IF (ABS(Z(I,J)).GT.P) P=ABS(Z(I,J))                             REBA0150
60     CONTINUE                                                         REBA0160
       DO 80 I=1,N                                                      REBA0170
80     Z(I,J)=Z(I,J)/P                                                  REBA0180
100   CONTINUE                                                          REBA0190
      RETURN                                                            REBA0200
      END                                                               REBA0210
C                                                                       JACO0010
C ----------------------------------------                              JACO0020
C                                                                       JACO0030
      SUBROUTINE JACOBI(N,A,D,V,B,Z)                                    JACO0040
      DIMENSION A(N,N),D(N),V(N,N),B(N),Z(N)                            JACO0050
C ---------- --- ---------------------------------- ---                 JACO0060
20    DO 21 I=1,N                                                       JACO0070
      DO 22 J=1,N                                                       JACO0080
22    V(I,J)=0.0                                                        JACO0090
21    V(I,I)=1.                                                         JACO0100
10    DO 30 K=1,N                                                       JACO0110
      D(K)=A(K,K)                                                       JACO0120
      B(K)=D(K)                                                         JACO0130
      Z(K)=0.0                                                          JACO0140
30    CONTINUE                                                          JACO0150
      DO 40 I=1,500                                                     JACO0160
      SM=0.0                                                            JACO0170
      N1=N-1                                                            JACO0180
      DO 50 IP=1,N1                                                     JACO0190
      IP1=IP+1                                                          JACO0200
      DO 50 IQ=IP1,N                                                    JACO0210
50    SM=SM+ABS(A(IP,IQ))                                               JACO0220
      IF (SM) 60,70,60                                                  JACO0230
60    IF (I-4) 80,90,90                                                 JACO0240
80    TRESH=0.2*SM/FLOAT(N)**2                                          JACO0250
      GOTO 100                                                          JACO0260
90    TRESH=0.0                                                         JACO0270
100   DO 110 IP=1,N1                                                    JACO0280
      IP1=IP+1                                                          JACO0290
      DO 110 IQ=IP1,N                                                   JACO0300
      G=100.0*ABS(A(IP,IQ))                                             JACO0310
      IF (G.LE.1.E-20) G=0.0                                            JACO0320
      IF (I-4) 120,120,130                                              JACO0330
130   IF (ABS(D(IP))+G-ABS(D(IP))) 120,140,120                          JACO0340
140   IF (ABS(D(IQ))+G-ABS(D(IQ))) 120,150,120                          JACO0350
150   A(IP,IQ)=0.0                                                      JACO0360
      GOTO 110                                                          JACO0370
120   IF (ABS(A(IP,IQ))-TRESH) 110,110,160                              JACO0380
160   H=D(IQ)-D(IP)                                                     JACO0390
      IF (ABS(H)+G-ABS(H)) 170,180,170                                  JACO0400
180   T=A(IP,IQ)/H                                                      JACO0410
      GOTO 190                                                          JACO0420
170   THETA=0.5*H/A(IP,IQ)                                              JACO0430
      T=SIGN(1.0,THETA)/(ABS(THETA)+SQRT(1.+THETA*THETA))               JACO0440
190   C=1./SQRT(1.0+T*T)                                                JACO0450
      S=T*C                                                             JACO0460
      TAU=S/(1.+C)                                                      JACO0470
      H=T*A(IP,IQ)                                                      JACO0480
      Z(IP)=Z(IP)-H                                                     JACO0490
      Z(IQ)=Z(IQ)+H                                                     JACO0500
      D(IP)=D(IP)-H                                                     JACO0510
      D(IQ)=D(IQ)+H                                                     JACO0520
      A(IP,IQ)=0.0                                                      JACO0530
200   IP2=IP-1                                                          JACO0540
      IF (IP2) 210,210,201                                              JACO0550
201   DO 205 K=1,IP2                                                    JACO0560
      G=A(K,IP)                                                         JACO0570
      H=A(K,IQ)                                                         JACO0580
      A(K,IP)=G-S*(H+G*TAU)                                             JACO0590
      A(K,IQ)=H+S*(G-H*TAU)                                             JACO0600
205   CONTINUE                                                          JACO0610
210   IQ1=IQ-1                                                          JACO0620
      IF (IP1-IQ1) 211,211,220                                          JACO0630
211   DO 215 K=IP1,IQ1                                                  JACO0640
      G=A(IP,K)                                                         JACO0650
      H=A(K,IQ)                                                         JACO0660
      A(IP,K)=G-S*(H+G*TAU)                                             JACO0670
      A(K,IQ)=H+S*(G-H*TAU)                                             JACO0680
215   CONTINUE                                                          JACO0690
220   IQ1=IQ+1                                                          JACO0700
      IF (IQ1-N) 221,221,230                                            JACO0710
221   DO 225 K=IQ1,N                                                    JACO0720
      G=A(IP,K)                                                         JACO0730
      H=A(IQ,K)                                                         JACO0740
      A(IP,K)=G-S*(H+G*TAU)                                             JACO0750
      A(IQ,K)=H+S*(G-H*TAU)                                             JACO0760
225   CONTINUE                                                          JACO0770
230   DO 235 K=1,N                                                      JACO0780
      G=V(K,IP)                                                         JACO0790
      H=V(K,IQ)                                                         JACO0800
      V(K,IP)=G-S*(H+G*TAU)                                             JACO0810
      V(K,IQ)=H+S*(G-H*TAU)                                             JACO0820
235   CONTINUE                                                          JACO0830
110   CONTINUE                                                          JACO0840
      DO 240 K=1,N                                                      JACO0850
      B(K)=B(K)+Z(K)                                                    JACO0860
      D(K)=B(K)                                                         JACO0870
      Z(K)=0.0                                                          JACO0880
240   CONTINUE                                                          JACO0890
40    CONTINUE                                                          JACO0900
70    RETURN                                                            JACO0910
      END                                                               JACO0920
      SUBROUTINE SOL04                                                  SOL00010
      LOGICAL BTEST,UNBAL                                               SOL00020
$INCLUDE:''ZCOMN'                                                       SOL00030
C                                                                       SOL00040
C---------------------------------                                      SOL00050
C- SET GEOMETRIC STIFFNESS FLAGS ---                                    SOL00060
C---------------------------------                                      SOL00070
      Z(IZKGDT)=0.00                                                    SOL00080
      KGLOAD=NZ(IZKGDT+1)                                               SOL00090
      KGTYPE=NZ(IZKGDT+2)                                               SOL00100
      KGFORM=NZ(IZKGDT+3)                                               SOL00110
C                                                                       SOL00120
C-----------------------------                                          SOL00130
C- READ INFO AND SET OPTIONS ---                                        SOL00140
C-----------------------------                                          SOL00150
      READ(105,*) MAXELD,IPRNT,IWRITE,UNBAL                             SOL00160
      NLOAD=1                                                           SOL00170
      IPRNT=MAX(IPRNT,1)                                                SOL00180
C                                                                       SOL00190
C-----------------                                                      SOL00200
C- PRINT HEADER ---                                                     SOL00210
C-----------------                                                      SOL00220
      WRITE(108,101) TITLE(1),TITLE(2),IPRNT,IWRITE                     SOL00230
  101 FORMAT ('1 STRUCTURE . . . ',A,/' SOLUTION..... ',A,/             SOL00240
     &        1X,' SOLUTION #4, STATIC NONLINEAR SOLUTION '/            SOL00250
     &        1X,'=======================================',/           SOL00260
     &        1X,' INTERVAL FOR PRINTING DATA..... ... ',I5/           SOL00270
     &        1X,' INTERVAL FOR WRITING DATA TO FILE.. ...',I5/)        SOL00280
C                                                                       SOL00290
      IF (KGFORM.EQ.2) WRITE(108,11)                                    SOL00300
   11 FORMAT (1X,' GEOMETRIC STIFFNESS IS NOT INCLUDED, ',              SOL00310
     &        ' USE KGDATA(3)=1 TO INCLUDE GEOMETRIC STIFFNESS.')       SOL00320
      IF (     UNBAL) WRITE(108,13)  ' ARE '                            SOL00330
      IF ( NOT.UNBAL) WRITE(108,13)  ' ARE NOT '                        SOL00340
   13 FORMAT('  UNBALANCED JOINT FORCES',A,'ADDED TO THE NEXT CYCLE')    SOL00350
C                                                                       SOL00360
C-----------------------------------                                    SOL00370
C-INPUT FILE OUTPUT CONTROLL DATA ---                                   SOL00380
C-----------------------------------                                    SOL00390
      IF (IWRITE.GT.0) THEN                                             SOL00400
        IOPT=1                                                          SOL00410
        TO=0                                                            SOL00420
        DT=0                                                            SOL00430
        CALL DMPDAT(IOPT,IWRITE,TO,DT)                                  SOL00440
      ENDIF                                                             SOL00450
C                                                                       SOL00460
C-------------------------------------------                            SOL00470
C- INITILIZE STORAGE REQ'D FOR THIS SOLN ---                            SOL00480
C-------------------------------------------                            SOL00490
C     Z(IZLOAD) - LOCATION OF THE TOTAL LOAD MATRIX                     SOL00500
C     Z(IZDISP) - LOCATION OF THE TOTAL DISPLACEMENT MATRIX             SOL00510
C     Z(IZVEL ) - LOCATION OF THE TOTAL VELOCITY MATRIX                 SOL00520
C     Z(IZDLOA) - LOCATION OF THE INCRIMENTAL LOAD MATRIX               SOL00530
C     Z(IZDDSP) - LOCATION OF THE INCRIMENTAL DISPLACEMENT MATRIX       SOL00540
C     Z(IZDVEL) - LOCATION OF THE INCRIMENTAL VELOCITY MATRIX           SOL00550
C     Z(IZQ   ) - LOCATION OF THE JOINT LOADS                           SOL00560
C     Z(IZR   ) - LOCATION OF THE ELEMENT LOADS                         SOL00570
C                                                                       SOL00580
      IF (IZLOAD.EQ.0) THEN                                             SOL00590
        IZLOAD = IZ                                                     SOL00600
        IZ     = IZ+NDOF                                                SOL00610
        DO 1 I=1,NDOF                                                   SOL00620
    1   Z(I+IZLOAD-1) =0                                                SOL00630
        ENDIF                                                           SOL00640
C                                                                       SOL00650
      IF (IZDISP.EQ.0) THEN                                             SOL00660
        IZDISP = IZ                                                     SOL00670
        IZ     = IZ+NDOF                                                SOL00680
        DO 2 I=1,NDOF                                                   SOL00690
    2   Z(I+IZDISP-1) =0                                                SOL00700
        ENDIF                                                           SOL00710
C                                                                       SOL00720
      IF (IZVEL .EQ.0) THEN                                             SOL00730
        IZVEL  = IZ                                                     SOL00740
        IZ     = IZ+NDOF                                                SOL00750
        DO 3 I=1,NDOF                                                   SOL00760
    3   Z(I+IZVEL -1) =0                                                SOL00770
        ENDIF                                                           SOL00780
C                                                                       SOL00790
      IF (IZDLOA.EQ.0) THEN                                             SOL00800
        IZDLOA = IZ                                                     SOL00810
        IZ     = IZ+NDOF                                                SOL00820
        DO 5 I=1,NDOF                                                   SOL00830
    5   Z(I+IZDLOA-1) =0                                                SOL00840
        ENDIF                                                           SOL00850
C                                                                       SOL00860
      IF (IZDDSP.EQ.0) THEN                                             SOL00870
        IZDDSP = IZ                                                     SOL00880
        IZ     = IZ+NDOF                                                SOL00890
        DO 6 I=1,NDOF                                                   SOL00900
    6   Z(I+IZDDSP-1) =0                                                SOL00910
        ENDIF                                                           SOL00920
C                                                                       SOL00930
      IF (IZDVEL.EQ.0) THEN                                             SOL00940
        IZDVEL = IZ                                                     SOL00950
        IZ     = IZ+NDOF                                                SOL00960
        DO 7 I=1,NDOF                                                   SOL00970
    7   Z(I+IZDVEL-1) =0                                                SOL00980
        ENDIF                                                           SOL00990
C                                                                       SOL01000
        IZQ    = IZ                                                     SOL01010
        IZ     = IZ+NDOF                                                SOL01020
        DO 8 I=1,NDOF                                                   SOL01030
    8   Z(I+IZQ-1) =0                                                   SOL01040
C                                                                       SOL01050
        IZR    = IZ                                                     SOL01060
        IZ     = IZ+NDOF                                                SOL01070
        DO 9 I=1,NDOF                                                   SOL01080
    9   Z(I+IZR-1) =0                                                   SOL01090
C                                                                       SOL01100
      IF (IZFUB  EQ.0) THEN                                             SOL01110
        IZFUB  = IZ                                                     SOL01120
        IZ     = IZ+NDOF                                                SOL01130
        ENDIF                                                           SOL01140
        DO 10 I=1,NDOF                                                  SOL01150
   10   Z(I+IZFUB-1) =0                                                 SOL01160
C                                                                       SOL01170
        IZUBLD = IZ                                                     SOL01180
        IZ     = IZ+NDOF                                                SOL01190
        IZUBDP = IZ                                                     SOL01200
        IZ     = IZ+NDOF                                                SOL01210
        DO 12 I=1,NDOF                                                  SOL01220
        Z(I+IZUBLD-1) =0                                                SOL01230
   12   Z(I+IZUBDP-1) =0                                                SOL01240
C                                                                       SOL01250
      IW=IZ+1                                                           SOL01260
C                                                                       SOL01270
C---- DETERMINE THE LENGTH OF THE STIFFNESS MATRIX                      SOL01280
        IMD=NZ(IZMD+NDOF) - NZ(IZMD) +1                                 SOL01290
C                                                                       SOL01300
C      ---- SET UP CONSTANTS -----------------                          SOL01310
      L1=1                                                              SOL01320
      L2=NCOND                                                          SOL01330
      L3=L2+1                                                           SOL01340
      L4=NCOND +NFREE                                                   SOL01350
      L5=L4+1                                                           SOL01360
      L6=NDOF                                                           SOL01370
C                                                                       SOL01380
      NLOAD=1                                                           SOL01390
C                                                                       SOL01400
      ELSTIC= FALSE.                                                    SOL01410
C                                                                       SOL01420
      IF (MAXELD.GT.0) THEN                                             SOL01430
          IZIELD=IZ                                                     SOL01440
          IZ=IZ + MAXELD*5                                              SOL01450
          IZELD=IZ                                                      SOL01460
          IZ=IZ + MAXELD*12                                             SOL01470
      ENDIF                                                             SOL01480
C                                                                       SOL01490
C-------------------------------------                                  SOL01500
      IZPG  = IZ                                                        SOL01510
      IZ    = IZ+NDOF                                                   SOL01520
      IZDPG = IZ                                                        SOL01530
      IZ    = IZ+NDOF                                                   SOL01540
      IZDMAX= IZ                                                        SOL01550
      IZ    = IZ+NDOF                                                   SOL01560
      IZRMAX= IZ                                                        SOL01570
      IZ    = IZ+NDOF                                                   SOL01580
      IZDRLO= IZ                                                        SOL01590
      IZ    = IZ+NDOF                                                   SOL01600
      IZRLCA= IZ                                                        SOL01610
      IZ    = IZ+NDOF                                                   SOL01620
      DO 27 I=0,NDOF-1                                                  SOL01630
        Z(IZPG  +I)=0                                                   SOL01640
        Z(IZDPG +I)=0                                                   SOL01650
        Z(IZDMAX+I)=0                                                   SOL01660
        Z(IZRMAX+I)=0                                                   SOL01670
        Z(IZRLCA+I)=0                                                   SOL01680
```

```
   27   Z(IZDRLO-I)=0                                                    SOL01690
C                                                                       SOL01700
C-------------------------------------                                  SOL01710
C                                                                       SOL01720
C------ CHECK MEMORY                                                   SOL01730
      CALL CKSTOR(NDOF,IW)                                              SOL01740
C                                                                       SOL01750
C------ SET UP CONSTANTS                                               SOL01760
      IMD   =NZ(IZMD  )+NDOF                                            SOL01770
      IMDMS =NZ(IZMDMS)+NDOF                                            SOL01780
      IMDXG =NZ(IZMDXG)+NDOF                                            SOL01790
      IMDMSL=1                                                          SOL01800
      IMDS  =1                                                          SOL01810
C                                                                       SOL01820
C------------------------                                               SOL01830
C- ZERO GROUND MOTION ---                                              SOL01840
C------------------------                                               SOL01850
      DO 18 I=1,6                                                       SOL01860
   18   SUMRCT(I)=0                                                     SOL01870
C                                                                       SOL01880
C------------------------                                               SOL01890
C- ZERO GROUND MOTION ---                                              SOL01900
C------------------------                                               SOL01910
      DO 19 I=1,9                                                       SOL01920
   19   GROUND(I)=0.                                                    SOL01930
C                                                                       SOL01940
C------------------------                                               SOL01950
C- CALC PRINTING INFO ---                                              SOL01960
C------------------------                                               SOL01970
      IPDISP=6*NNODE                                                    SOL01980
      IPRCTN=7                                                          SOL01990
      DO 20 I=1,NNODE                                                   SOL02000
        DO 15 J=0,5                                                     SOL02010
        IF (BTEST(NZ(IZJPLG-1+I),J)) THEN                               SOL02020
          IPRCTN=IPRCTN+1                                               SOL02030
          GO TO 20                                                      SOL02040
        ENDIF                                                           SOL02050
   15   CONTINUE                                                        SOL02060
   20 CONTINUE                                                          SOL02070
      IPELEM=NELMT                                                      SOL02080
      LSTTYP=0                                                          SOL02090
      DO 26 I=1,NELMT                                                   SOL02100
        J=NZ(IZELE+I)                                                   SOL02110
        IF (NZ(J) NE.LSTTYP) IPELEM=IPELEM+4                            SOL02120
   26   LSTTYP=NZ(J)                                                    SOL02130
  111 FORMAT (9P,' ZMATRX(',I6,')=',I15,1P,G20.10)                      SOL02140
C                                                                       SOL02150
      CALL SOL06A(L1,L2,L3,L4,L5,L6,IMD,IMDMS,IMDXG,                    SOL02160
     &  MAXELD   , IPRNT    , IWRITE   , UNBAL    ,                     SOL02170
     &  NLOAD    , NNODE    , NCOS     , NELMT    ,                     SOL02180
     &  IZDDSP   , IZDLOA   , IBUG     , MAXNOD   ,                     SOL02190
     &  TITLE    , STEPID   , NDOF     , Z(IZRLOA),                     SOL02200
     &  KGFORM   , KGLOAD   , KGTYPE   , Z(IZDRLO),                     SOL02210
     &  NEWKG    , NEWX     , SUMRCT   ,                               SOL02220
     &  IPDISP   , IPRCTN   , IPELEM   , Z(IZDMAX), Z(IZRMAX),         SOL02230
     &  Z(IZDISP), Z(IZVEL ), Z(IZLOAD), Z(IZQ   ), Z(IZPUB ),         SOL02240
     &  Z(IZDDSP), Z(IZDVEL), Z(IZDLOA), Z(IZR   ), Z(IW    ),         SOL02250
     &  Z(IZSTIP), Z(IZMASS), Z(IZKG  ), Z(IZELD ), Z(IZUBLD),         SOL02260
     & NZ(IZMD  ),NZ(IZMDMS),NZ(IZMDXG),NZ(IZIELD), Z(IZUBDP),         SOL02270
     & NZ(IZID  ),NZ(IZIDOF), Z(IZCNST),NZ(IZJPLG), Z(IZCORD),         SOL02280
     &  Z(IZCOS ),NZ(IZJCOS), EIE,ESE,PSE,EXE,EDD ,                     SOL02290
     &  Z(IZPG  ), Z(IZDPG ))                                           SOL02300
C                                                                       SOL02310
      RETURN                                                            SOL02320
      END                                                               SOL02330
C                                                                       JOIL0010
C-----------------------------------------------------------------------JOIL0020
C                                                                       JOIL0040
      SUBROUTINE JOILOA(BUG,MAXNOD,NDOF,NNODE,NLOAD,NF,ID,IDOF,CONST,   JOIL0050
     &                  FORCE,DSPPLG,JTFLG,TITLE,HEAD)                  JOIL0060
      CHARACTER*(*) TITLE(2)                                            JOIL0070
      CHARACTER  DIR*4,SUPT*25,DOF*18                                   JOIL0080
      CHARACTER*1  CX(6)                                                JOIL0090
      CHARACTER*14 F(6)                                                 JOIL0100
      LOGICAL DSPPLG,FIRST,BUG,HEAD,BTEST                               JOIL0110
      DIMENSION IDOF(MAXNOD,6),CONST(MAXNOD,6),FORCE(NDOF,NLOAD)        JOIL0120
      DIMENSION ID(MAXNOD),JTFLG(MAXNOD)                                JOIL0130
C                                                                       JOIL0140
      FIRST = .TRUE.                                                    JOIL0150
      DSPPLG=.FALSE.                                                    JOIL0160
    5 FORMAT('1 STRUCTURE. ...: ',A,/'  SOLUTION.....: ',A,//)          JOIL0170
    6 FORMAT('    APPLIED JOINT LOADS',/'    ---------------------',/) JOIL0180
C                                                                       JOIL0190
C----------------------                                                 JOIL0200
C- INPUT LOADING DATA --                                                JOIL0210
C----------------------                                                 JOIL0220
   10 READ(105,*) LOAD,JOINT,JTGEN,JTINC,DIR,VALUE                      JOIL0230
      IF (INDEX(DIR,'END').NE.0) GO TO 100                              JOIL0240
   15 DOF=' '                                                           JOIL0250
      IF (FIRST) THEN                                                   JOIL0260
            IF (HEAD) WRITE(108,5) TITLE(1),TITLE(2)                    JOIL0270
            WRITE(108,6)                                                JOIL0280
            FIRST=.FALSE.                                               JOIL0290
      ENDIF                                                             JOIL0300
C                                                                       JOIL0310
C------- CHECK FOR A VALID JOINT NUMBER..                               JOIL0320
      IF (JOINT LT.ID(1) .OR. JOINT GT.ID(NNODE) ) THEN                 JOIL0330
         WRITE(108,21) LOAD                                             JOIL0340
         GO TO 10                                                       JOIL0350
      ENDIF                                                             JOIL0360
C                                                                       JOIL0370
       J=IQUICK(JOINT,ID,NNODE)                                         JOIL0380
       SUPT=' '                                                         JOIL0390
C                                                                       JOIL0400
C---------------------------------------------------------- ---- ------JOIL0410
C------  PUT A LOAD ON A JOINT                                          JOIL0420
C NOTES:  - IX, IY, IZ, MX, MY AND Z ARE THE DOF THAT THE LOADS ARE     JOIL0430
C           APPLIED TO.                                                 JOIL0440
C         - THE MOMENTS RESULTING FROM FX, FY AND FZ ARE DUE TO         JOIL0450
C           CONSTRAINTS.  IF A NODE IS NOT CONSTRAINED, THEN THE VALUE  JOIL0460
C           OF CONST IS ZERO..                                          JOIL0470
C         - LOADS APPLIED TO RESTRAINED DOF ARE INTERPERATED AS         JOIL0480
C           SUPPORT DISPLACEMENTS  IE: SUPPORT SETTELMENT OR            JOIL0490
C           DISPLACEMENT CONTROLL. . .                                  JOIL0500
C-----------------------------------------------------------------------JOIL0510
      IF (INDEX(DIR,'FX').NE.0) THEN                                    JOIL0520
         IX=IDOF(J,1)                                                   JOIL0530
         FORCE(IX,LOAD)=FORCE(IX,LOAD)+VALUE                            JOIL0540
         IF ( BTEST(JTFLG(J),12) ) THEN                                 JOIL0550
           MY=IDOF(J,5)                                                 JOIL0560
           FORCE(MY,LOAD)=FORCE(MY,LOAD)+VALUE*CONST(J,3)               JOIL0570
           MZ=IDOF(J,6)                                                 JOIL0580
           FORCE(MZ,LOAD)=FORCE(MZ,LOAD)+VALUE*CONST(J,5)               JOIL0590
           WRITE(DOF,35) IX,MY,MZ                                       JOIL0600
         ELSE                                                           JOIL0610
           WRITE(DOF,35) IX                                             JOIL0620
         ENDIF                                                          JOIL0630
         IF (IX GT.NF) THEN                                             JOIL0640
           DSPFLG= TRUE.                                                JOIL0650
           SUPT='...JOINT DISPLACEMENT ..'                              JOIL0660
         ENDIF                                                          JOIL0670
      ELSE IF (INDEX(DIR,'FY') NE.0) THEN                               JOIL0680
         IY=IDOF(J,2)                                                   JOIL0690
         FORCE(IY,LOAD)=FORCE(IY,LOAD)+VALUE                            JOIL0700
         IF (BTEST(JTFLG(J),13)) THEN                                   JOIL0710
           MX=IDOF(J,4)                                                 JOIL0720
           FORCE(MX,LOAD)=FORCE(MX,LOAD)+VALUE*CONST(J,1)               JOIL0730
           MZ=IDOF(J,6)                                                 JOIL0740
           FORCE(MZ,LOAD)=FORCE(MZ,LOAD)+VALUE*CONST(J,6)               JOIL0750
           WRITE(DOF,35) IY,MX,MZ                                       JOIL0760
         ELSE                                                           JOIL0770
           WRITE(DOF,35) IY                                             JOIL0780
         ENDIF                                                          JOIL0790
         IF (IY GT.NF) THEN                                             JOIL0800
           DSPFLG=.TRUE.                                                JOIL0810
           SUPT='...JOINT DISPLACEMENT.  '                              JOIL0820
         ENDIF                                                          JOIL0830
      ELSE IF (INDEX(DIR,'FZ') NE.0) THEN                               JOIL0840
         JZ=IDOF(J,3)                                                   JOIL0850
         FORCE(JZ,LOAD)=FORCE(JZ,LOAD)+VALUE                            JOIL0860
         IF (BTEST(JTFLG(J),14)) THEN                                   JOIL0870
           MX=IDOF(J,4)                                                 JOIL0880
           FORCE(MX,LOAD)=FORCE(MX,LOAD)+VALUE*CONST(J,2)               JOIL0890
           MY=IDOF(J,5)                                                 JOIL0900
           FORCE(MY,LOAD)=FORCE(MY,LOAD)+VALUE*CONST(J,4)               JOIL0910
           WRITE(DOF,35) JZ,MX,MY                                       JOIL0920
         ELSE                                                           JOIL0930
           WRITE(DOF,35) JZ                                             JOIL0940
         ENDIF                                                          JOIL0950
         IF (JZ.GT.NF) THEN                                             JOIL0960
           DSPFLG=.TRUE.                                                JOIL0970
           SUPT='...JOINT DISPLACEMENT ..'                              JOIL0980
         ENDIF                                                          JOIL0990
      ELSE IF (INDEX(DIR,'MX') NE.0) THEN                               JOIL1000
         MX=IDOF(J,4)                                                   JOIL1010
         FORCE(MX,LOAD)=FORCE(MX,LOAD)+VALUE                            JOIL1020
         WRITE(DOF,35) MX                                               JOIL1030
         IF (MX.GT.NF) THEN                                             JOIL1040
           DSPFLG= TRUE.                                                JOIL1050
           SUPT='...JOINT DISPLACEMENT...'                              JOIL1060
         ENDIF                                                          JOIL1070
      ELSE IF (INDEX(DIR,'MY') NE.0) THEN                               JOIL1080
         MY=IDOF(J,5)                                                   JOIL1090
         FORCE(MY,LOAD)=FORCE(MY,LOAD)+VALUE                            JOIL1100
         WRITE(DOF,35) MY                                               JOIL1110
         IF (MY.GT.NF) THEN                                             JOIL1120
           DSPFLG=.TRUE.                                                JOIL1130
           SUPT='... JOINT DISPLACEMENT...'                             JOIL1140
         ENDIF                                                          JOIL1150
      ELSE IF (INDEX(DIR,'MZ') NE.0) THEN                               JOIL1160
         MZ=IDOF(J,6)                                                   JOIL1170
         FORCE(MZ,LOAD)=FORCE(MZ,LOAD)+VALUE                            JOIL1180
         WRITE(DOF,35) MZ                                               JOIL1190
         IF (MZ.GT.NF) THEN                                             JOIL1200
           DSPFLG= TRUE.                                                JOIL1210
           SUPT='  ..JOINT DISPLACEMENT ..'                            JOIL1220
         ENDIF                                                          JOIL1230
      ELSE                                                              JOIL1240
         WRITE(108,20) JOINT,DIR,VALUE                                  JOIL1250
      ENDIF                                                             JOIL1260
      WRITE(108,30) LOAD,JOINT,DIR,DOF,VALUE,SUPT                       JOIL1270
      IF (JTGEN.GT.0) THEN                                              JOIL1280
         JTGEN=JTGEN-1                                                  JOIL1290
         JOINT=JOINT+JTINC                                              JOIL1300
         GO TO 15                                                       JOIL1310
      ENDIF                                                             JOIL1320
      GO TO 10                                                          JOIL1330
C                                                                       JOIL1340
   20 FORMAT ('  INVALID JOINT LOAD... ',I6,2X,A,2X,1P,G14.6,          JOIL1350
     &        ' --- LOAD IS SKIPPED')                                   JOIL1360
   21 FORMAT ('  INVALID LOAD CASE ... ',I6,' --- LOAD IS SKIPPED')    JOIL1370
   30 FORMAT (' LOAD CASE:',I3,'  JOINT:',I6,'  DIRECTION: ',A,         JOIL1380
     &        ' DOF(S)',A,'  MAGNITUDE:',1P,G14.6,2X,A)                JOIL1390
   35 FORMAT (I5,:,',',I5, ,',',I5)                                     JOIL1400
C                                                                       JOIL1410
C------- RETURN IF NO JOINT LOADS WERE INPUT                            JOIL1420
  100 IF (FIRST) RETURN                                                 JOIL1430
C                                                                       JOIL1440
      IF (.NOT.BUG) RETURN                                              JOIL1450
C------- PRINT JOINT LOADS ON EACH DOF                                  JOIL1460
      N5=NDOF+5                                                         JOIL1470
      DO 120 L=1,NLOAD                                                  JOIL1480
        IF (HEAD) WRITE(108,5  ) TITLE(1),TITLE(2)                      JOIL1490
        WRITE(108,130) L                                                JOIL1500
        DO 120 I=1,N5,5                                                 JOIL1510
        I5=MIN((I+4),NDOF)                                              JOIL1520
        DO 110 J=I,I5                                                   JOIL1530
          K=J-I+1                                                       JOIL1540
          CX(K)=' '                                                     JOIL1550
  110     IF (J.GT.NF) CX(K)='*'                                        JOIL1560
          WRITE(108,140) (J,FORCE(J,L),CX(J-I+1),J=I,I5)                JOIL1570
  120 CONTINUE                                                          JOIL1580
C                                                                       JOIL1590
  130 FORMAT ('  APPLIED JOINT LOADS,   LOADING #',I5,10X,             JOIL1600
     &        '  NOTE: * DENOTES SUPPORT DISPLACEMENT'/                 JOIL1610
     &        ' ---------------------- '//                             JOIL1620
     &      5X,5('  DOF',7X,'LOAD    ')/)                             JOIL1630
  140   FORMAT (5X,5(0P,I5,1P,G14.6,A) )                                JOIL1640
C                                                                       JOIL1650
      DO 220 L=1,NLOAD                                                  JOIL1660
        IF (HEAD) WRITE(108,5  ) TITLE(1),TITLE(2)                      JOIL1670
        WRITE(108,200) L                                                JOIL1680
        DO 220 I=1,NNODE                                                JOIL1690
        DO 210 J=1,6                                                    JOIL1700
          CX(J)=' '                                                     JOIL1710
          IF (BTEST(JTFLG(I),J+11)) THEN                                JOIL1720
            F(J)=' '                                                    JOIL1730
          ELSE                                                          JOIL1740
            K=IDOF(I,J)                                                 JOIL1750
            WRITE(F(J),240) FORCE(K,L)                                  JOIL1760
            IF (K.GT.NF) CX(J)='*'                                      JOIL1770
          ENDIF                                                         JOIL1780
  210     CONTINUE                                                      JOIL1790
          WRITE(108,230) ID(I),(F(J),CX(J),J=1,6)                       JOIL1800
  220 CONTINUE                                                          JOIL1810
C                                                                       JOIL1820
  200 FORMAT ('  APPLIED JOINT LOADS,   LOADING #',I5,10X,             JOIL1830
     &        '  NOTE: * DENOTES SUPPORT DISPLACEMENT'/                 JOIL1840
     &        ' ---------------------- '//6X,'NODE',7X,'FX',           JOIL1850
     &      13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ'/)              JOIL1860
  230   FORMAT (5X,I5,12A )                                             JOIL1870
  240   FORMAT (1P,G14.6)                                               JOIL1880
C                                                                       JOIL1890
      RETURN                                                            JOIL1900
      END                                                               JOIL1910
C                                                                       JOID0010
C-----------------------------------------------------------------------JOID0020
C                                                                       JOID0030
      SUBROUTINE JOIDSP(IOPT,MAXNOD,NDOF,NNODE,NLOAD,NF,ID,IDOF,        JOID0040
     &            CONST,DISP,TITLE,NAME,STEPID,HEAD,NCOS,COSINE,JTCOS)  JOID0050
      CHARACTER*(*) TITLE(2),STEPID,NAME                                JOID0060
      CHARACTER*15 CD(6),EQUAL*60                                       JOID0070
      LOGICAL HEAD                                                      JOID0080
      DIMENSION IDOF(MAXNOD,6),CONST(MAXNOD,6)                          JOID0090
      DIMENSION DISP(NDOF,NLOAD),ID(MAXNOD)                             JOID0100
      DIMENSION D(6),Q(6),COSINE(3,3,NCOS),JTCOS(MAXNOD)                JOID0110
      DATA EQUAL                                                        JOID0120
     & /'------------------------------------------------------------'/ JOID0130
C                                                                       JOID0140
      NLEN=MIN(LEN(NAME)+1,60)                                          JOID0150
      NC=1                                                              JOID0160
C                                                                       JOID0170
C-----------------------------                                          JOID0180
C- LOOP FOR EACH LOAD CASE  ---                                         JOID0190
C-----------------------------                                          JOID0200
C                                                                       JOID0210
      DO 300 L=1,NLOAD                                                  JOID0220
        NDSPLS=0                                                        JOID0230
        IF (.NOT (NCOS EQ 1  AND  COSINE(1,1,1) EQ.1  AND.              JOID0240
     &       COSINE(2,2,1).EQ 1  .AND. COSINE(3,3,1).EQ.1 )             JOID0250
     &       AND. IOPT.EQ.2 ) GO TO 210                                 JOID0260
C                                                                       JOID0270
C-----------------------------                                          JOID0280
C- PRINT GLOBAL DISPLACEMENTS ---                                       JOID0290
C-----------------------------                                          JOID0300
C  NOTE:DISP MAY CONTAIN DISPLACEMENTS, VELOCITIES, ACCELERATIONS OR    JOID0310
C       EIGENVALUES...                                                  JOID0320
C       ALL OF WHICH ARE PRINTED OUT AT EACH NODE. . .                  JOID0330
C                                                                       JOID0340
        IF (HEAD) WRITE(108,310) TITLE(1),TITLE(2)                      JOID0350
        IF (IOPT.NE.2) WRITE(108,315) NAME,L,STEPID,EQUAL(1:NLEN)       JOID0360
```

```
      IF (IOPT.EQ.2) WRITE(108,415) NAME,          EQUAL(1:NLEN)       JOID0370
      DO 100 I=1,NNODE                                                 JOID0380
C                                                                      JOID0390
C. .  ..... TRANSFER DISPL TO LOCAL VECTOR .. ... ... .                JOID0400
        DO 20 J=1,6                                                    JOID0410
          IJ=IDOF(I,J)                                                 JOID0420
   20     D(J)=DISP(IJ,L)                                             JOID0430
C                                                                      JOID0440
C . ... ..  ADD ROTATIONS TO DISPL FOR CONSTRAINTS                     JOID0450
C... ... ..  IF UNCONSTRAINED, CONST(I,J)=0                            JOID0460
        D(1) = D(1) + D(5)*CONST(I,3) - D(6)*CCNST(I,5)                JOID0470
        D(2) = D(2) + D(4)*CONST(I,1) - D(6)*CCNST(I,6)                JOID0480
        D(3) = D(3) + D(4)*CONST(I,2) + D(5)*CCNST(I,4)                JOID0490
C                                                                      JOID0500
C. .... ....  TRANSFER TO GLOBAL DISPLACEMENTS .....                  JOID0510
        JCOS=JTCOS(I)                                                  JOID0520
        DO 25 I1=1,3                                                   JOID0530
          I2=I1-3                                                      JOID0540
          G(I1)=0                                                      JOID0550
          G(I2)=0                                                      JOID0560
          DO 25 J1=1,3                                                 JOID0570
            J2=J1-3                                                    JOID0580
            G(I1)=G(I1)+COSINE(J1,I1,JCOS)*D(J1)                       JOID0590
   25       G(I2)=G(I2)+COSINE(J1,I1,JCOS)*D(J2)                       JOID0600
C                                                                      JOID0610
C........ ... WRITE DISPL                                              JOID0620
        DO 30 J=1,6                                                    JOID0630
   30   WRITE(CD(J),9) G(J)                                           JOID0640
    9   FORMAT(1P,G14.6,' ')                                          JOID0650
        WRITE(108,220) ID(I),(CD(K),K=1,6)                             JOID0660
C                                                                      JOID0670
        NDSPLS=NDSPLS+1                                                JOID0680
  100 CONTINUE                                                         JOID0690
C                                                                      JOID0700
      IF (NCOS.EQ.1 .AND. COSINE(1,1,1).EQ.1 .AND. COSINE(2,2,1).EQ.1  JOID0710
     &              .AND. COSINE(3,3,1).EQ.1 ) GO TO 300               JOID0720
C                                                                      JOID0730
C-------------------------------                                       JOID0740
C- PRINT JOINT DISPLACEMENTS ---                                       JOID0750
C-------------------------------                                       JOID0760
C  NOTE DISP MAY CONTAIN DISPLACEMENTS, VELOCITIES, ACCELERATIONS OR   JOID0770
C       EIGENVALUES...                                                 JOID0780
C       ALL OF WHICH ARE PRINTED OUT AT EACH NODE....                  JOID0790
C                                                                      JOID0800
  210   IF (HEAD.AND. NDSPLS.GT.20) WRITE(108,310) TITLE(1),TITLE(2)   JOID0810
        IF (IOPT.NE.2) WRITE(108,316) NAME,L,STEPID,EQUAL(1:NLEN)      JOID0820
        IF (IOPT.EQ.2) WRITE(108,416) NAME,          EQUAL(1:NLEN)     JOID0830
        DO 200 I=1,NNODE                                               JOID0840
C                                                                      JOID0850
C.....  ....  TRANSFER GLOBAL DISPL TO LOCAL VECTOR ... ... ... .     JOID0860
        DO 219 J=1,6                                                   JOID0870
          IJ=IDOF(I,J)                                                 JOID0880
  219     D(J)=DISP(IJ,L)                                              JOID0890
C                                                                      JOID0900
C.... .  . ADD ROTATIONS TO DISPL FOR CONSTRAINTS                      JOID0910
C. ...  .... IF UNCONSTRAINED, CONST(I,J)=0                            JOID0920
        D(1) = D(1) + D(5)*CONST(I,3) - D(6)*CONST(I,5)                JOID0930
        D(2) = D(2) + D(4)*CONST(I,1) - D(6)*CONST(I,6)                JOID0940
        D(3) = D(3) + D(4)*CONST(I,2) + D(5)*CONST(I,4)                JOID0950
C                                                                      JOID0960
C. ..  ..... WRITE DISPL                                               JOID0970
        DO 230 J=1,6                                                   JOID0980
          WRITE(CD(J),9) D(J)                                          JOID0990
          IJ=IDOF(I,J)                                                 JOID1000
  230     IF (IJ.GT.NF) CD(J)(15:15)='*'                               JOID1010
        WRITE(108,221) ID(I),(CD(K),K=1,6),JTCOS(I)                    JOID1020
C                                                                      JOID1030
  200 CONTINUE                                                         JOID1040
      WRITE(108,320)                                                   JOID1050
  300 CONTINUE                                                         JOID1060
C                                                                      JOID1070
   10 FORMAT (1P,G15.7)                                                JOID1080
  320 FORMAT (5X,I5,6A)                                                JOID1090
  221 FORMAT (5X,I5,6A,I10)                                            JOID1100
  310 FORMAT ('1 STRUCTURE.....: ',A,/'  SOLUTION.....: ',A)           JOID1110
  315 FORMAT (//2X, 'GCS ',A,',  LOADING #',I5,5X,A, /                 JOID1120
     &        1X,'------',A//6X,'NODE',7X,'DX',13X,'DY',13X,'DZ',      JOID1130
     &        13X,'RX',13X,'RY',13X,'RZ'/)                            JOID1140
  316 FORMAT (//2X, 'JCS ',A,',  LOADING #',I5,5X,A, /1X,'------',A     JOID1150
     &        //6X,'NODE',7X,'DX',13X,'DY',13X,'DZ',13X,'RX',         JOID1160
     &        13X,'RY',13X,'RZ',6X,'  COSINE #'/)                     JOID1170
  415 FORMAT (//2X, 'GCS ',A/1X,'------',A/25X,                       JOID1180
     &       'NOTE: MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'//6X,  JOID1190
     &  'NODE',7X,'DX',13X,'DY',13X,'DZ',13X,'RX',13X,'RY',13X,'RZ'/)  JOID1200
  416 FORMAT (//2X, 'JCS ',A/1X,'------',A/25X,                       JOID1210
     &       'NOTE: MAXIMUM VALUES MAY NOT OCCUR SIMULTANEOUSLY'//6X,  JOID1220
     &  'NODE',7X,'DX',13X,'DY',13X,'DZ',13X,'RX',13X,'RY',13X,'RZ',   JOID1230
     &  6X,'  COSINE #'/)                                              JOID1240
  320 FORMAT (4X,'NOTE: * DENOTES A RESTRAINED DEGREE OF FREEDOM')     JOID1250
C                                                                      JOID1260
      RETURN                                                           JOID1270
      END                                                              JOID1280
C                                                                      ELEL0010
C-----------------------------------------------------------------------ELEL0020
C                                                                      ELEL0030
      SUBROUTINE ELELOA(BUG,NF,NELMT,MAXELD,NELD,NDOF,NLOAD,FORCE,     ELEL0040
     &                  IELD,ELD,IDISP,ILOAD ,NAME,TITLE,HEAD)         ELEL0050
C                                                                      ELEL0060
C NOTE ON ARGUMENTS. THE LOAD MATRIX FORCE IS PASSED FROM THE CALLING  ELEL0070
C  ROUTINE.  FORCE IS EQUIVALENCED TO THE Z MATRIX BY THE CALLING ROUTINELEL0080
C  IE. Z(ILOAD)=FORCE(1,1) WHERE ILOAD IS THE OFFSET IN THE Z MATRIX.  ELEL0090
C  BOTH THE CALLING ROUTINE AND SUBROUTINE ELE_LIB USE THE Z MATRIX    ELEL0100
C                                                                      ELEL0110
      LOGICAL FIRST,HEAD,FALSE,AXIAL,BUG                               ELEL0120
      CHARACTER*(*) TITLE(2),NAME                                      ELEL0130
      CHARACTER*20 TYPE,DIRI,DIR*40                                    ELEL0140
      CHARACTER*1 CX(6)                                                ELEL0150
      DIMENSION FORCE(NDOF,NLOAD),IELD(5,MAXELD),ELD(12,MAXELD)        ELEL0160
      DIMENSION RINPUT(100)                                            ELEL0170
C                                                                      ELEL0180
      FIRST=.TRUE.                                                     ELEL0190
    5 FORMAT ('1 STRUCTURE.....: ',A,/'  SOLUTION ... : ',A,/)         ELEL0200
    6 FORMAT (/'     ELEMENT LOADS  ',/'  ---------- ',//              ELEL0210
     & 1X,' LOAD ELEM  GEN  INC  GROUP & DIRECTION    VALUES ... '/)  ELEL0220
C----------------------                                                ELEL0230
C- INPUT LOADING DATA --                                               ELEL0240
C----------------------                                                ELEL0250
      N=0                                                              ELEL0260
   10 N=N+1                                                            ELEL0270
      IF (N.GT.MAXELD) GO TO 100                                       ELEL0280
   20 READ(105,*) (IELD(I,N),I=1,4),TYPE,DIRI,ELD(1,N)                ELEL0290
      DIR-TYPE//DIRI                                                   ELEL0300
      IF (INDEX(DIR,'END').NE.0) GO TO 100                             ELEL0310
      NELD=N                                                           ELEL0320
      IF (FIRST)  THEN                                                 ELEL0330
           IF (HEAD) WRITE(108,5) TITLE(1),TITLE(2)                    ELEL0340
           WRITE(108,6)                                                ELEL0350
           FIRST=.FALSE.                                               ELEL0360
      ENDIF                                                            ELEL0370
C                                                                      ELEL0380
C------- PROCESS LOAD GROUPS AND READ ADDITIONAL INFO  .               ELEL0390
      IF (INDEX(DIR,'CONC').NE.0) THEN                                 ELEL0400
         KODE=100                                                      ELEL0410
         J=2                                                           ELEL0420
         BACKSPACE(105)                                                ELEL0430
         READ(105,*) (IELD(I,N),I=1,4),TYPE,DIRI,(ELD(I,N),I=1,J)      ELEL0440
         DIR-TYPE//DIRI                                                ELEL0450
         CALL CKRNG(ELD(2,N),0.,1.,'DISTANCE')                         ELEL0460
      ELSE IF (INDEX(DIR,'UNIF') NE 0) THEN                            ELEL0470
         J=1                                                           ELEL0480
         KODE=200                                                      ELEL0490
      ELSE IF (INDEX(DIR,'VARR').NE.0) THEN                            ELEL0500
         KODE=300                                                      ELEL0510
         J=4                                                           ELEL0520
         BACKSPACE(105)                                                ELEL0530
         READ(105,*) (IELD(I,N),I=1,4),TYPE,DIRI,(ELD(I,N),I=1,J)      ELEL0540
         DIR-TYPE//DIRI                                                ELEL0550
         CALL CKRNG(ELD(3,N),0.,ELD(4,N),'DISTANCE')                   ELEL0560
         CALL CKRNG(ELD(4,N),ELD(3,N),1.,'DISTANCE')                   ELEL0570
      ELSE IF (INDEX(DIR,'FEM').NE.0) THEN                             ELEL0580
         KODE=400                                                      ELEL0590
         J=12                                                          ELEL0600
         BACKSPACE(105)                                                ELEL0610
         READ(105,*) (IELD(I,N),I=1,4),TYPE,DIRI,(ELD(I,N),I=1,J)      ELEL0620
         DIR-TYPE//DIRI                                                ELEL0630
      ELSE                                                             ELEL0640
         WRITE(108,30) 'INVALID LOAD TYPE',                            ELEL0650
     &                 (IELD(I,N),I=1,4),DIR,ELD(1,N)                  ELEL0660
         GO TO 20                                                      ELEL0670
      ENDIF                                                            ELEL0680
      IF (INDEX(DIR,'FX') NE 0) THEN                                   ELEL0690
         KODE=KODE+1                                                   ELEL0700
      ELSE IF (INDEX(DIR,'FY').NE.0) THEN                              ELEL0710
         KODE=KODE+2                                                   ELEL0720
      ELSE IF (INDEX(DIR,'FZ').NE.0) THEN                              ELEL0730
         KODE=KODE+3                                                   ELEL0740
      ELSE IF (INDEX(DIR,'MX').NE.0) THEN                              ELEL0750
         KODE=KODE+4                                                   ELEL0760
      ELSE IF (INDEX(DIR,'MY').NE.0) THEN                              ELEL0770
         KODE=KODE+5                                                   ELEL0780
      ELSE IF (INDEX(DIR,'MZ').NE.0) THEN                              ELEL0790
         KODE=KODE+6                                                   ELEL0800
      ELSE IF (KODE.NE.400) THEN                                       ELEL0810
         WRITE(108,30) 'INVALID LOAD DIRECTION',                       ELEL0820
     &                 (IELD(I,N),I=1,4),DIR,(ELD(1,N),I=1,J)          ELEL0830
         GO TO 20                                                      ELEL0840
      ENDIF                                                            ELEL0850
      IELD(5,N)=KODE                                                   ELEL0860
      WRITE(108,40) (IELD(I,N),I=1,4),DIR,(ELD(I,N),I=1,J)             ELEL0870
      GO TO 10                                                         ELEL0880
C                                                                      ELEL0890
   30 FORMAT (' ',A, ' --- LOAD IS SKIPPED'/                           ELEL0900
     &        1X,4I5,1X,A20,1P,6G14.6,:/(42X,6G14.6) )                 ELEL0910
   40 FORMAT (1X,4I5,1X,A20,1P,6G14.6,:/(42X,6G14.6) )                 ELEL0920
C                                                                      ELEL0930
C- ----- RETURN IF NO ELEMENT LOADS WERE INPUT                        ELEL0940
  100 IF (FIRST)  RETURN                                               ELEL0950
C                                                                      ELEL0960
C------- LOOP TO GENERATE MEMBER LOADS FOR EACH ELEMENT                ELEL0970
      IOPT=6                                                           ELEL0980
      DO 101 IELNO=1,NELMT                                             ELEL0990
        FALSE= FALSE.                                                  ELEL1000
  101   CALL ELELIB                                                    ELEL1010
     &  (IOPT,LSTTYP,FALSE,IREL,FALSE,NAME,EESE,EPSE,DAMAGE,DUCFAG,    ELEL1020
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IDISP,ILOAD,MSTOR)     ELEL1030
C                                                                      ELEL1040
C------- PRINT LOADS AT EACH DOF                                       ELEL1050
      IF (BUG) THEN                                                    ELEL1060
         N5=NDOF+5                                                     ELEL1070
         DO 120 L=1,NLOAD                                              ELEL1080
           IF (HEAD) WRITE(108,5) TITLE(1),TITLE(2)                    ELEL1090
           WRITE(108,130) L                                            ELEL1100
           DO 120 I=1,N5,5                                             ELEL1110
           I5=MIN((I+4),NDOF)                                          ELEL1120
           DO 110 J=I,I5                                               ELEL1130
             K=J-I+1                                                   ELEL1140
             CX(K)=' '                                                 ELEL1150
  110        IF (J.GT.NF) CX(K)='*'                                    ELEL1160
             WRITE(108,140) (J,FORCE(J,L),CX(J-I+1),J=I,I5)            ELEL1170
  120    CONTINUE                                                      ELEL1180
      ENDIF                                                            ELEL1190
C                                                                      ELEL1200
  130 FORMAT ('  GLOBAL JOINT FORCES DUE TO ELEMENT LOADS ',3X,        ELEL1210
     &        '   LOADING #',I5,10X,                                   ELEL1220
     &        '  NOTE. * DENOTES SUPPORT DISPLACEMENT'/                ELEL1230
     &        ' ----------------------------------------'//           ELEL1240
     &        5X,5('  DOF',7X,'LOAD    ')/)                          ELEL1250
  140   FORMAT (5X,5(0P,I5,1P,G14.6,A) )                               ELEL1260
C                                                                      ELEL1270
      RETURN                                                           ELEL1280
      END                                                              ELEL1290
C                                                                      CKRN0010
C-----------------------------------------------------------------------CKRN0020
C                                                                      CKRN0030
      SUBROUTINE CKRNG(V,A,B,NAME)                                     CKRN0040
      CHARACTER*(*) NAME                                               CKRN0050
      IF (A.LT.V .AND. V.LT.B) RETURN                                  CKRN0060
      WRITE(108,10) NAME,A,V,B                                         CKRN0070
   10 FORMAT(' ****)',A,' IS OUT OF RANGE, SET ',1P,G13.5,' LE INPUT=', CKRN0080
     &       G13.5,' LE ',G13 5)                                       CKRN0090
      V=MIN(V,B)                                                       CKRN0100
      V=MAX(V,A)                                                       CKRN0110
      RETURN                                                           CKRN0120
      END                                                              CKRN0130
C                                                                      ENER0010
C-----------------------------------------------------------------------ENER0020
C  ENERGY BALANCE                                                      ENER0030
C-----------------------------------------------------------------------ENER0040
      SUBROUTINE ENERGY(L1,L2,L3,L4,L6,MAXNOD,NCOS,PT2,                ENER0050
     &          INDMS,MDMS,DYN,WRITO,MASS,EIE,ESE,PSE,KKE,EDD,         ENER0060
     &          ALPHA,BETA,KGFORM,NELMT,NNODE,DT,SUMUB,SUMRC,          ENER0070
     &          DSUMRC,SUMPD,DSUMPD,FDAMP,DFDAMP,GA,GV,GD,GAT,         ENER0080
     &          GVT,GDT,WORK,WORK2,PG,DPG,DACC,DVEL,DDISP,ACC,         ENER0090
     &          VEL,DISP,LOAD,DLOAD,IDOF,JTFLG,COSINE,JTCOS)           ENER0100
C                                                                      ENER0110
      REAL MASS,LOAD                                                   ENER0120
      CHARACTER*80 NAME                                                ENER0130
      LOGICAL BTEST,PT2,AXIAL,DYN,WRITO                                ENER0140
      DIMENSION IDOF(MAXNOD,6) ,COSINE(3,3,NCOS),JTFLG(MAXNOD)         ENER0150
      DIMENSION MASS(INDMS),MDMS(L6+1),JTCOS(MAXNOD)                   ENER0160
      DIMENSION RINPUT(100),FDAMP(L6),DFDAMP(L6),PG(6),DPG(6)          ENER0170
      DIMENSION  SUMRC(6),DSUMRC(6),SUMPD(6),DSUMPD(6),SUMUB(6)        ENER0180
      DIMENSION  GD(3), GV(3), GA(3),GDT(3),GVT(3),GAT(3)              ENER0190
      DIMENSION DDISP(L6),DVEL(L6),DACC(L6)                            ENER0200
      DIMENSION  DISP(L6), VEL(L6), ACC(L6)                            ENER0210
      DIMENSION WORK(L6),WORK2(L6),LOAD(L6),DLOAD(L6)                  ENER0220
C                                                                      ENER0230
C------------------ NOMENCLATURE ------------------  ----------------CENER0240
C  DACC  INCRIMENTAL RELATIVE ACCELERATION                           CENER0250
C  DVEL  INCRIMENTAL RELATIVE VELOCITY                               CENER0260
C  DDISP INCRIMENTAL RELATIVE DISPLACEMENT                           CENER0270
C  ACC   PREVIOUS STEP'S RELATIVE ACCELERATION                       CENER0280
C  VEL   PREVIOUS STEP'S RELATIVE VELOCITY                           CENER0290
C  DISP  PREVIOUS STEP'S RELATIVE DISPLACEMENT                       CENER0300
C  DLOAD INCRIMENTAL APPLIED LOAD, NOT M*AG                          CENER0310
C  LOAD  PREVIOUS STEP'S APPLIED LOAD, NOT M*AG                      CENER0320
C  DFDAMP INCRIMENTAL DAMPING FORCE                                  CENER0330
C  FDAMP PREVIOUS STEP'S DAMPING FORCE                               CENER0340
C                                                                    CENER0350
C  GD    INCRIMENTAL GROUND DISPLACEMENT                             CENER0360
C  GV    INCRIMENTAL GROUND VELOCITY                                 CENER0370
C  GA    INCRIMENTAL GROUND ACCELERATION                             CENER0380
C  GDT   TOTAL GROUND DISPLACEMENT                                   CENER0390
C  GVT   TOTAL GROUND VELOCITY                                       CENER0400
C  GAT   TOTAL GROUND ACCELERATION                                   CENER0410
C  WORK  TEMP FORCE VECTOR                                           CENER0420
C  WORK2 TEMP DISPL OR VELOCITY VECTOR                               CENER0430
C-------- ------------------------------ ---------- -- ---CENER0440
C                                                                    ENER0450
      L41=L4+1                                                       ENER0460
C                                                                    ENER0470
C-------------------------------------                               ENER0480
C---  INTEGRATE GROUND ACCELERATION  ---                             ENER0490
C-------------------------------------                               ENER0500
      IF (DYN) THEN                                                  ENER0510
      DO 10 I=1,3                                                    ENER0520
        GA(I)=GA(I)                                                  ENER0530
        GV(I)=(2*GAT(I)+GA(I)) *DT /2                                ENER0540
        GD(I)=GVT(I)*DT + (3*GAT(I)+GA(I)) *DT*DT /6                 ENER0550
        GAT(I)=GA(I)+GAT(I)                                          ENER0560
        GVT(I)=GV(I)+GVT(I)                                          ENER0570
   10   GDT(I)=GD(I)+GDT(I)                                          ENER0580
```

```
      ENDIF                                                             ENER0590
        IF (PT2) THEN                                                   ENER0600
          IF (DYN) THEN                                                 ENER0610
              WRITE(108,22) (I,ACC(I),VEL(I),DISP(I),LOAD(I),I=1,L4)    ENER0620
              WRITE(108,23) (I,DACC(I),DVEL(I),DDISP(I),DLOAD(I),I=1,L4) ENER0630
              WRITE(108,24) (I,PG(I),DPG(I),FDAMP(I),DFDAMP(I),I=1,L6)  ENER0640
          ELSE                                                          ENER0650
              WRITE(108,22) (I,ACC(I),VEL(I),DISP(I),LOAD(I),I=1,L4)    ENER0660
              WRITE(108,23) (I,DACC(I),DVEL(I),DDISP(I),DLOAD(I),I=1,L6) ENER0670
              WRITE(108,24) (I,PG(I),DPG(I),FDAMP(I),DFDAMP(I),I=1,L6)  ENER0680
          ENDIF                                                         ENER0690
   22   FORMAT(/T10,'TOTAL RELATIVE RESPONSE FOR PREVIOUS TIME STEP'/    ENER0700
     &         (T10,'DOF:',I6,' ACC.',1P,G13.5,' VEL:',G13 5,           ENER0710
     &          ' DSP ',G13.5,'LOAD ',G13.5))                           ENER0720
   23   FORMAT(/T10,'INCRIMENTAL RELATIVE RESPONSE'/                     ENER0730
     &         (T10,'DOF ',I6,' ACC:',1P,G13.5,' VEL:',G13 5,           ENER0740
     &          ' DSP ',G13.5,'LOAD ',G13.5))                           ENER0750
   24   FORMAT(/T10,'SPECIAL FORCES '/(T10,'DOF:',I6,' FD ',1P,G13 5,    ENER0760
     &          ' DFD:',G13 5,' FDAMP.',G13.5,' DFDAMP.',G13.5))        ENER0770
        ENDIF                                                           ENER0780
C                                                                       ENER0790
C---------------------------------------                                ENER0800
C---  CALCULATE INPUT EARTHQUAKE ENERGY  ---                            ENER0810
C---------------------------------------                                ENER0820
C                                                                       ENER0830
      DEIE=0                                                            ENER0840
C                                                                       ENER0850
C..... CALCULATE THE INCR. INPUT ENERGY DUE TO GRND MOTION              ENER0860
      IF (DYN) THEN                                                     ENER0870
          IF (PT2) THEN                                                 ENER0880
           WRITE(108,*) ' SUMRC', SUMRC                                 ENER0890
           WRITE(108,*) 'DSUMRC',DSUMRC                                 ENER0900
           WRITE(108,*) ' SUMFD', SUMFD                                 ENER0910
           WRITE(108,*) 'DSUMFD',DSUMFD                                 ENER0920
           WRITE(108,*) ' SUMUB', SUMUB                                 ENER0930
          ENDIF                                                         ENER0940
          DO 48 I=1,3                                                   ENER0950
            FORCEI= SUMRC(I)+ 0 5*DSUMRC(I) + SUMFD(I) + 0.5*DSUMFD(I)  ENER0960
     &            + SUMUB(I)                                            ENER0970
            ENER=GD(I)*FORCEI                                           ENER0980
            IF (PT2) WRITE(108,*) I,'FORCEI',FORCEI,                    ENER0990
     &                              ' GD',GD(I),'  ENER',ENER           ENER1000
            DEIE=DEIE + ENER                                            ENER1010
   48     ENDIF                                                         ENER1020
C                                                                       ENER1030
C.....  CALCULATE THE INCR. INPUT ENERGY DUE TO APPLIED LOADS...        ENER1040
          N2=L6                                                         ENER1050
          IF (DYN) N2=L4                                                ENER1060
          DO 50 I=L3,N2                                                 ENER1070
            FORCEI=LOAD(I)+DLOAD(I)/2                                   ENER1080
            ENER=FORCEI*DDISP(I)                                        ENER1090
            IF (PT2) WRITE(108,*) I,                                    ENER1100
     &        'FORCEI',FORCEI,'  DDISP',DDISP(I),'  ENER',ENER          ENER1110
   50       DEIE=DEIE + ENER                                            ENER1120
C                                                                       ENER1130
C.  ..  CALCULATE THE ENERGY DUE TO GEOMETRIC STIFFNESS                 ENER1140
        IF (KGFORM.EQ.2 ) THEN                                          ENER1150
C....    GET THE INCRIMENTAL DISPL AND STORE IN VECTOR WORK2. ...       ENER1160
C  .     SAVE INCRIMENTAL IN VECTOR WORK2                               ENER1170
          DO 51 I=L3,L4                                                 ENER1180
   51       WORK(I)=DDISP(I)                                           ENER1190
C .      ADD GROUND RESPONSE IN JOINT CORRDINATES TO RELATIVE RESP      ENER1200
C          MULTIPLY GROUND DISPL * DIRECTION COSINE                     ENER1210
C          OMIT ALL DOF THAT ARE CONSTRAINED TO A MASTER DOF            ENER1220
          IF (GD(1).NE.0 .OR. GD(2).NE.0 OR. GD(3).NE.0) THEN           ENER1230
            DO 52 NODE=1,NNODE                                          ENER1240
              ICOS=JTCOS(NODE)                                          ENER1250
              DO 52 K=1,3                                               ENER1260
              IF (.NOT. BTEST(JTFLG(NODE),K-11)) THEN                   ENER1270
                II=IDOF(NODE,K)                                         ENER1280
                IF (II.LE.L4  AND. II.GE.L3)                            ENER1290
     &          WORK(II)=WORK(II)+ GD(1)*COSINE(1,K,ICOS)               ENER1300
     &                + GD(2)*COSINE(2,K,ICOS) + GD(3)*COSINE(3,K,ICOS) ENER1310
              ENDIF                                                     ENER1320
   52         CONTINUE                                                  ENER1330
        ENDIF                                                           ENER1340
C                                                                       ENER1350
C. .... CALCULATE THE INCRIMENTAL INPUT ENERGY DUE TO XG...             ENER1360
        DIXG=0                                                          ENER1370
          DO 53 I=L3,L4                                                 ENER1380
             ENER= (PG(I)+DPG(I)/2)*WORK(I)                             ENER1390
             IF (PT2) WRITE(108,*) I,                                   ENER1400
     &         'PG,DPG',PG(I),DPG(I),'  DDISP',WORK(I),'  ENER',ENER    ENER1410
   53       DIXG=DIXG+ENER                                             ENER1420
        DEIE=DEIE+DIXG                                                  ENER1430
      ENDIF                                                             ENER1440
C*                                                                      ENER1450
C..... CALCULATE THE TOTAL INPUT ENERGY                                 ENER1460
      EIE=EIE + DEIE                                                    ENER1470
C                                                                       ENER1480
C---------------------------------------                                ENER1490
C---  CALCULATE TOTAL STRAIN ENERGY   ---                               ENER1500
C---------------------------------------                                ENER1510
C..... CALCULATE THE TOTAL STRAIN ENERGY                                ENER1520
      DESE=ESE                                                          ENER1530
      DPSE=PSE                                                          ENER1540
      ESE=0                                                             ENER1550
      PSE=0                                                             ENER1560
C                                                                       ENER1570
C     ..... GET ELASTIC AND PLASTIC STRAIN ENERGY FOR EACH ELEMENT      ENER1580
      LSTTYP=0                                                          ENER1590
      DO 60 IELNO=1,NELMT                                               ENER1600
        CALL ELELIB                                                     ENER1610
     &  (10,LSTTYP,.FALSE.,IREL,.FALSE.,NAME,EESE,EPSE,DAMAGE,DUCFAG,   ENER1620
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDISP,IZLOAD,MSTOR)    ENER1630
        IF (PT2) WRITE(108,55) IELNO,EESE,EPSE                          ENER1640
   55   FORMAT('   ELEMENT#',I5,'     ELASTIC SE=',1P,G13 4,           ENER1650
     &                         '    PLASTIC SE=',1P,G13.4)             ENER1660
        ESE = ESE + EESE                                                ENER1670
   60   PSE = PSE + EPSE                                                ENER1680
C                                                                       ENER1690
C .... CALCULATE THE INCRIMENTAL STRAIN ENERGY                          ENER1700
       DESE=ESE-DESE                                                    ENER1710
       DPSE=PSE-DPSE                                                    ENER1720
C                                                                       ENER1730
C---------------------------------                                      ENER1740
C---  CALCULATE KINETIC ENERGY  ---                                     ENER1750
C---------------------------------                                      ENER1760
      IF (DYN) THEN                                                     ENER1770
C  ..... GET THE ABSOLUTE VELOCITY AND STORE IN VECTOR WORK2... .       ENER1780
C. .     SAVE RELATIVE RESPONSE IN VECTOR WORK2                         ENER1790
         DO 15 I=1,L4                                                   ENER1800
   15      WORK2(I)=VEL(I)+DVEL(I)                                      ENER1810
C  .     ADD GROUND RESPONSE IN JOINT CORRDINATES TO RELATIVE RESP      ENER1820
C          MULTIPLY GROUND VELOCITY * DIRECTION COSINE                  ENER1830
C          OMIT ALL DOF THAT ARE CONSTRAINED TO A MASTER DOF...         ENER1840
         IF (GVT(1) NE.0  OR  GVT(2).NE 0  OR  GVT(3) NE 0) THEN        ENER1850
           DO 20 NODE=1,NNODE                                           ENER1860
             ICOS=JTCOS(NODE)                                           ENER1870
             DO 20 K=1,3                                                ENER1880
             IF (.NOT. BTEST(JTFLG(NODE),K-11)) THEN                    ENER1890
               II=IDOF(NODE,K)                                          ENER1900
               IF (II.LE.L4)                                            ENER1910
     &         WORK2(II)=WORK2(II)        + GVT(1)*COSINE(1,K,ICOS)     ENER1920
     &           + GVT(2)*COSINE(2,K,ICOS) + GVT(3)*COSINE(3,K,ICOS)    ENER1930
             ENDIF                                                      ENER1940
   20        CONTINUE                                                   ENER1950
         ENDIF                                                          ENER1960
C                                                                       ENER1970
C. . .. CALCULATE THE TOTAL KE                                          ENER1980
        DEKE=EKE                                                        ENER1990
        CALL ETMPLY(PT2,L1,L2,L3,L4,L41,L4,                             ENER2000
     &              IMDMS,MDMS,MASS,WORK,WORK2,WORK2,EKE)               ENER2010
        EKE=EKE/2 0                                                     ENER2020
C                                                                       ENER2030
C .  .. CALCULATE THE INCRIMENTAL KE                                    ENER2040
        DEKE=EKE-DEKE                                                   ENER2050
      ELSE                                                              ENER2060
C                                                                       ENER2070
C. .    ZERO KE FOR STATIC CASE.                                        ENER2080
        DEKE=0                                                          ENER2090
        EKE =0                                                          ENER2100
      ENDIF                                                             ENER2110
C                                                                       ENER2120
C-----------------------------------                                    ENER2130
C---  CALCULATE DAMPINED ENERGY   ---                                   ENER2140
C-----------------------------------                                    ENER2150
      IF (DYN) THEN                                                     ENER2160
C                                                                       ENER2170
C..... CALCULATE THE INCRIMENTAL DAMPING ENERGY                         ENER2180
        DEDD =0                                                         ENER2190
        IF (ALPHA NE.0  OR  BETA NE.0) THEN                             ENER2200
          DO 85 I=L3,L4                                                 ENER2210
            FORCEI= FDAMP(I)+DFDAMP(I)/2                                ENER2220
            VMI = VEL(I)+ DVEL(I)/2                                     ENER2230
            ENER=FORCEI*VMI                                             ENER2240
            IF (PT2) WRITE(108,*) I,                                    ENER2250
     &        'FORCEI',FORCEI,'  VM',VMI,'  ENER',ENER                  ENER2260
   85       DEDD=DEDD + ENER                                            ENER2270
          DEDD=DEDD*DT                                                  ENER2280
        ENDIF                                                           ENER2290
C                                                                       ENER2300
C..... CALCULATE THE TOTAL DAMPINED ENERGY                              ENER2310
        EDD=EDD+DEDD                                                    ENER2320
      ELSE                                                              ENER2330
C                                                                       ENER2340
C..... ZERO DE FOR STATIC CASE.                                         ENER2350
        EDD=0                                                           ENER2360
        DEDD=0                                                          ENER2370
      ENDIF                                                             ENER2380
C                                                                       ENER2390
C-------------------------------                                        ENER2400
C---  CHECK ENERGY BALANCE  ---                                         ENER2410
C-------------------------------                                        ENER2420
      DIFF=EIE-ESE-EKE-PSE-EDD                                          ENER2430
      IF (EIE.NE.0) THEN                                                ENER2440
        RE=DIFF/EIE * 100                                               ENER2450
      ELSE                                                              ENER2460
        RE=100.00                                                       ENER2470
      ENDIF                                                             ENER2480
C                                                                       ENER2490
C-------------------                                                    ENER2500
C---  PRINT DATA ---                                                    ENER2510
C-------------------                                                    ENER2520
      IF (WRITO) WRITE(108,100) DEIE,EIE,DESE,ESE,DEKE,EKE,             ENER2530
     &                          DPSE,PSE,DEDD,EDD,RE                    ENER2540
  100 FORMAT (/,T10,' ENERGY BALANCE'/T10,'----------------',            ENER2550
     & T40,'INCRIMENTAL',T58,'TOTAL'/                                   ENER2560
     & T15,'INPUT ENERGY'              ,T40,1P,G12.5,T55,G12 5/         ENER2570
     & T15,'ELASTIC STRAIN ENERGY'     ,T40,1P,G12.5,T55,G12.5/         ENER2580
     & T15,'KINETIC ENERGY '           ,T40,1P,G12.5,T55,G12 5/         ENER2590
     & T15,'PLASTIC STRAIN ENERGY'     ,T40,1P,G12 5,T55,G12.5/         ENER2600
     & T15,'DAMPING ENERGY'            ,T40,1P,G12.5,T55,G12.5/         ENER2610
     & T15,'RELATIVE ER','ROR'         ,                T55,G12.5,'%')  ENER2620
      IF (DYN.AND WRITO) WRITE (6,110) ( GD(I), GV(I), GA(I),I=1,3),    ENER2630
     &                                 (GDT(I),GVT(I),GAT(I),I=1,3)     ENER2640
  110 FORMAT (/,T10,' GROUND RESPONSE' /T10,'----------------'/         ENER2650
     &          T10,'INCRIMENTAL VALUES',T46,'DISPLACEMENT',            ENER2660
     &          T63,'VELOCITY',T76,'ACCELERATION'/                      ENER2670
     &          T15,'GLOBAL X AXIS TRANSLATION...',T45,1P,3G15.5,/      ENER2680
     &          T15,'GLOBAL Y AXIS TRANSLATION...',T45,1P,3G15.5,/      ENER2690
     &          T15,'GLOBAL Z AXIS TRANSLATION...',T45,1P,3G15.5,/      ENER2700
     &          T10,'TOTAL VALUES'/                                     ENER2710
     &          T15,'GLOBAL X AXIS TRANSLATION...',T45,1P,3G15.5,/      ENER2720
     &          T15,'GLOBAL Y AXIS TRANSLATION.. ',T45,1P,3G15.5,/      ENER2730
     &          T15,'GLOBAL Z AXIS TRANSLATION...',T45,1P,3G15.5)       ENER2740
C                                                                       ENER2750
C---------------------------------------                                ENER2760
C---  DUCTILITIES AND EXCURSION RATIOS ---                              ENER2770
C---------------------------------------                                ENER2780
       LSTTYP=0                                                         ENER2790
      DO 200 IELNO=1,NELMT                                              ENER2800
  200   CALL ELELIB                                                     ENER2810
     &  (11,LSTTYP,.FALSE.,IREL,.FALSE.,NAME,EESE,EPSE,DAMAGE,DUCFAG,   ENER2820
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDISP,IZLOAD,MSTOR)    ENER2830
C                                                                       ENER2840
      RETURN                                                            ENER2850
      END                                                               ENER2860
C                                                                       ETMP0010
C-----------------------------------------------------------------------ETMP0020
C   CALC THE ENERGY (E) FOR U(TRANSP) * P = U(TRANSP) * A * X           ETMP0030
C                                                                       ETMP0040
C - MATRIX A IS AN UPPER TRIANGULAR MATRIX, WITH MAIN DIAGONAL ADDRESS  ETMP0050
C    MATRIX MD.                                                         ETMP0060
C    DOF L1 TO L3 HAVE BEEN CONDENSED OUT                               ETMP0070
C    DOF L3 TO L4 HAVE BEEN MODIFIED FOR THE CONDENSATION OF L1 TO L2   ETMP0080
C    DOF L5 TO L6 HAVE NOT BEEN MODIFIED                                ETMP0090
C - X IS A VECTOR.                                                      ETMP0100
C - P IS A DUMMY INPUT VECTOR.                                          ETMP0110
C-----------------------------------------------------------------------ETMP0120
      SUBROUTINE ETMPLY(FRINT,L1,L2,L3,L4,L5,L6,IMD,MD,A,P,X,U,E)       ETMP0130
      DIMENSION A(IMD),X(L6),P(L6),MD(L6+1),U(L6)                       ETMP0140
      LOGICAL PRINT,T,F                                                 ETMP0150
      IJI(I,J)=MD(J)-J+I                                                ETMP0160
C                                                                       ETMP0170
      T=.TRUE.                                                          ETMP0180
      F=.FALSE.                                                         ETMP0190
C                                                                       ETMP0200
      IF (PRINT) THEN                                                   ETMP0210
        WRITE(108,500) 'CALC ENERGY FOR PARTIALLY REDUCED A MATRIX',    ETMP0220
     &    L1,L2,L3,L4,L5,L6                                             ETMP0230
        CALL LMATRX(A,MD,L6,IMD,'MATRIX A ')                            ETMP0240
        DO 515 I=L1,L6                                                  ETMP0250
  515     WRITE(108,520) I,X(I),U(I)                                    ETMP0260
      ENDIF                                                             ETMP0270
C                                                                       ETMP0280
C--- -------- CALC PP" = KFP"*X1                                        ETMP0290
      CALL TMPLY(P,T,IMD,L6,L3,L4,L3,L4,MD,A,P,X)                       ETMP0300
C                                                                       ETMP0310
C------- ---- CALC E1 = XP' * PP"                                       ETMP0320
      E=0                                                               ETMP0330
      DO 10 L=L3,L4                                                     ETMP0340
   10   E = E + U(L)*P(L)                                               ETMP0350
      E1=E                                                              ETMP0360
C                                                                       ETMP0370
C------------ CALC PP  = KFR *XR                                        ETMP0380
      CALL TMPLY(P,T,IMD,L6,L1,L4,L5,L6,MD,A,P,X)                       ETMP0390
C                                                                       ETMP0400
C ------------ CALC PP  = KFR *XR                                       ETMP0410
      CALL TMPLY(P,T,IMD,L6,L5,L6,L1,L6,MD,A,P,X)                       ETMP0420
C                                                                       ETMP0430
C------------ CALC E2 + E3 + E4 = XP * PP + XR * PR                     ETMP0440
      DO 20 L=L1,L6                                                     ETMP0450
   20   E = E + U(L)*P(L)                                               ETMP0460
C                                                                       ETMP0470
C ------------ PRINT ANSWER  . .                                        ETMP0480
      IF (PRINT) WRITE(108,510) E,E1                                    ETMP0490
C                                                                       ETMP0500
  500 FORMAT (/2X,A,/2X,'L1:',I5,'   L2:',I5,'   L3:',I5,'   L4:',I5,  ETMP0510
     &        '   L5:',I5,'   L6:',I5)                                  ETMP0520
  510 FORMAT (/2X,' TOTAL ENERGY:',1P,G13.5,'  E1:',G13.5)              ETMP0530
  520 FORMAT (2X,'DOF:',I6,'  X(DOF)',1P,G14 5,'  U(DOF):',G14 5)       ETMP0540
      RETURN                                                            ETMP0550
      END                                                               ETMP0560
C                                                                       TMPL0010
C-----------------------------------------------------------------------TMPL0020
C   TRIANGLAR MATRIX MULTIPLICATION ....                                TMPL0030
C   MATRIX A IS AN UPPER TRIANGULAR MATRIX, WITH MAIN DIAGONAL ADDRESS  TMPL0040
C   MATRIX MD.  X IS A VECTOR   P=AX                                    TMPL0050
C   THE MULTIPLICATION IS CARRIED OUT FROM ROW IR1 TO IR2  AND FROM     TMPL0060
C   COLUMN JC1 TO JC2.                                                  TMPL0070
C-----------------------------------------------------------------------TMPL0080
```

```
      SUBROUTINE TMPLY(PRINT,ZERO,IMD,N,IR1,IR2,JC1,JC2,MD,A,P,X)       TMPL0090
      DIMENSION A(IMD),X(N),P(N),MD(N+1)                                TMPL0100
      LOGICAL PRINT,ZERO                                                TMPL0110
C                                                                       TMPL0120
      IJI(I,J)=MD(J)+J-I                                                TMPL0130
      IF (PRINT) THEN                                                   TMPL0140
        WRITE(108,500) 'TRIANGULAR MULTIPLACATION',IR1,IR2,JC1,JC2,N    TMPL0150
        CALL LMATRX(A,MD,N,MD(N-1),'MATRIX A ')                         TMPL0160
      ENDIF                                                             TMPL0170
C       ---- ZERO P  ---                                              TMPL0180
      IF (ZERO) THEN                                                    TMPL0190
        DO 320 J=IR1,IR2                                                TMPL0200
  320     P(J)=0                                                        TMPL0210
      ENDIF                                                             TMPL0220
C                                                                       TMPL0230
C-------- MULTIPLY TERMS ABOVE AND INCLUDING THE MAIN DIAGONAL          TMPL0240
      DO 20 J=JC1,JC2                                                   TMPL0250
        IRC=J-MD(J-1)+MD(J)+1                                           TMPL0260
        MDJ=MD(J)-J                                                     TMPL0270
        I1=MAX(IRC,IR1)                                                 TMPL0280
        I2=MIN(J,IR2)                                                   TMPL0290
        IF (I2.GE.I1) THEN                                              TMPL0300
          DO 10 I=I1,I2                                                 TMPL0310
   10       P(I)=P(I)+A(MDJ-I)*X(J)                                    TMPL0320
        ENDIF                                                           TMPL0330
   20 CONTINUE                                                          TMPL0340
C                                                                       TMPL0350
C------ -- MULTIPLY TERMS BELOW THE MAIN DIAGONAL                       TMPL0360
      DO 30 I=IR1,IR2                                                   TMPL0370
        IRC=I MD(I-1)-MD(I)+1                                           TMPL0380
        MDI=MD(I)+I                                                     TMPL0390
        J1=MAX(IRC,JC1)                                                 TMPL0400
        J2=MIN(I-1,JC2)                                                 TMPL0410
        IF (J2.GE.J1) THEN                                              TMPL0420
          DO 40 J=J1,J2                                                 TMPL0430
   40       P(I)=P(I)+A(MDI-J)*X(J)                                    TMPL0440
        ENDIF                                                           TMPL0450
   30 CONTINUE                                                          TMPL0460
  500 FORMAT (/2X,A/,2X,' IR1:',I5,' IR2:',I5,                         TMPL0470
     &                  ' JC1:',I5,' JC2:',I5,' N:',I5)                TMPL0480
      IF (PRINT) THEN                                                   TMPL0490
        DO 515 I=IR1,IR2                                                TMPL0500
  515     WRITE(108,520) I,X(I),P(I)                                    TMPL0510
          WRITE(108,*) ' '                                              TMPL0520
  520   FORMAT (2X,'DOF:',I6,'  X(DOF)',1P,G14.5,' A*X=P(DOF):',G14.5)  TMPL0530
      ENDIF                                                             TMPL0540
      RETURN                                                            TMPL0550
      END                                                               TMPL0560
C-----------------------------------------------------------------------SOL00010
      SUBROUTINE SOL06A                                                 SOL00020
     & (L1,L2,L3,L4,L5,L6,IMD,IMDMS,IMDKG,                              SOL00030
     &  MAXELD   , IPRNT    , IWRITE   , UNBAL    ,                    SOL00040
     &  NLOAD    , NNODE    , NCOS     , NELMT    ,                    SOL00050
     &  IZDISP   , IZDLOA   , IBUG     , MAXNOD   ,                    SOL00060
     &  TITLE    , STEPID   , NDOF     , RLOAD    ,                    SOL00070
     &  KGFORM   , KGLOAD   , KGTYPE   , DRLOA    ,                    SOL00080
     &  NEWKG    , NEWK     , SUMRC    ,                               SOL00090
     &  IPDISP   , IPRCTN   , IPELEM   , DSPMAX   , RCTMAX   ,         SOL00100
     &  DISP     , VEL      , LOAD     , Q        , PUB      ,         SOL00110
     &  DDISP    , DVEL     , DLOAD    , R        , WORK     ,         SOL00120
     &  STIFF    , MASS     , KG       , ELD      , UBLOAD   ,         SOL00130
     &  MD       , MDMS     , MDKG     , IELD     , UBDISP   ,         SOL00140
     &  ID       , IDOF     , CNST     , JTFLG    , COORD    ,         SOL00150
     &  COSINE   , JTCOS    ,  EIE,ESE,PSE,EKE,EDD,                    SOL00160
     &  PG       , DPG      )                                          SOL00170
C                                                                       SOL00180
      COMMON /IDSTEP/ISTEP                                              SOL00190
      LOGICAL PRINT,DSPPLG,HEAD,REPEAT,AXIAL,TEST,BTEST,NEWFAC          SOL00200
      LOGICAL NEWKG,NEWK,UNBAL,WRIT0,BUG3,BUG5,BUG6,BUG11              SOL00210
      CHARACTER*(*) TITLE(2),STEPID                                     SOL00220
      CHARACTER NAME*80,STEP*10,STEP2*10                                SOL00230
      REAL LOAD,MASS,KG                                                 SOL00240
      INTEGER FDAMP                                                     SOL00250
      DIMENSION SUMRC(6),DSUMRC(6),SUMFD(6),DSUMFD(6),SUMRCM(6),SUMUB(6)SOL00260
      DIMENSION DSPMAX(L6),RCTMAX(L6),DRLOA(L6),RLOAD(L6)               SOL00270
      DIMENSION RINPUT(100),WORX(2*L6),ID(MAXNOD),GAT(3),GVT(3),GDT(3)  SOL00280
      DIMENSION STIFF(IMD) ,MASS(IMDMS),  KG(IMDKG)                     SOL00290
      DIMENSION    MD(L6+1),MDMS(L6+1 ),MDKG(L6+1 )                     SOL00300
      DIMENSION  LOAD(L6), DISP(L6), VEL(L6), Q(L6),PG(L6),DPG(L6)      SOL00310
      DIMENSION DLOAD(L6),DDISP(L6),DVEL(L6), R(L6)                     SOL00320
      DIMENSION  PUB(L6),UBLOAD(L6),UBDISP(L6)                          SOL00330
      DIMENSION IELD(1),ELD(1),GA(3),GV(3),GD(3)                        SOL00340
      DIMENSION COORD(MAXNOD,3),JTFLG(MAXNOD),JTCOS(MAXNOD),            SOL00350
     &          IDOF(MAXNOD,6) ,COSINE(3,3,NCOS),CNST(MAXNOD,6)         SOL00360
C                                                                       SOL00370
      DO 5 I=1,6                                                        SOL00380
5     SUMUB(I)=0.                                                       SOL00390
      SUMFCR=0                                                          SOL00400
      SUMMOM=0                                                          SOL00410
      SUMMFO=0.                                                         SOL00420
      SUMMMO=0                                                          SOL00430
C--------------------------------                                       SOL00440
      BUG3 =BTEST(IBUG, 3)                                              SOL00450
      BUG5 =BTEST(IBUG, 5)                                              SOL00460
      BUG6 =BTEST(IBUG, 6)                                              SOL00470
      BUG11=BTEST(IBUG,11)                                              SOL00480
C--------------------------------                                       SOL00490
C                                                                       SOL00500
C--------------------------------                                       SOL00510
C- INITIALIZE ENERGY CONSTANTS ---                                      SOL00520
C--------------------------------                                       SOL00530
      EIE=0                                                             SOL00540
      ESE=0                                                             SOL00550
      PSE=0                                                             SOL00560
      EKE=0                                                             SOL00570
      EDD=0                                                             SOL00580
C                                                                       SOL00590
C--------------------------------                                       SOL00600
C- SET GEOMETRIC STIFFNESS FLAGS ---                                    SOL00610
C--------------------------------                                       SOL00620
      NEWKG=.TRUE.                                                      SOL00630
      IF (KGLOAD.EQ.0 .OR. KGTYPE.EQ.0 .OR. KGFORM.EQ.0) NEWKG=.FALSE.  SOL00640
C                                                                       SOL00650
C------------------------                                               SOL00660
C- SET STIFFNESS FLAGS ---                                              SOL00670
C------------------------                                               SOL00680
      NEWK = .TRUE.                                                     SOL00690
      KSAME= 0                                                          SOL00700
C                                                                       SOL00710
C----------------------                                                 SOL00720
C- INPUT LOADING DATA --                                                SOL00730
C----------------------                                                 SOL00740
C---------- MATRICIES Q AND R ARE ZEROED IN THE CALLING PGM.            SOL00750
C                                                                       SOL00760
C       ----- INPUT JOINT LOADS --------------------                    SOL00770
C       JOINT LOADS AND SUPPORT DISPLACEMENTS ARE STORED IN Q           SOL00780
      CALL JOILOA(BUG3,MAXNOD,NDOF,NNODE,NLOAD,L4,                      SOL00790
     &            ID,IDOF,CNST,Q, DSPPLG,JTFLG,TITLE,.FALSE.)           SOL00800
C                                                                       SOL00810
C       ----- INPUT ELEMENT LOADS ---------------- --                   SOL00820
C       ELEMENT LOADS ARE STORED IN R                                   SOL00830
      IF (MAXELD.GT.0) CALL ELELOA(BUG3,L5,NELMT,MAXELD,NELD,NDOF,      SOL00840
     &           NLOAD,R,IELD,ELD,IZDISP,IZLOAD,NAME,TITLE,.FALSE.)     SOL00850
C                                                                       SOL00860
C--------------------------------                                       SOL00870
C-WRITE INITIAL DATA TO OUTPUT FILE ---                                 SOL00880
C--------------------------------                                       SOL00890
      IF (IWRITE.GT.0) THEN                                             SOL00900
        IOPT=2                                                          SOL00910
        CALL DMPDAT(IOPT,IWRITE,TO,DT)                                  SOL00920
      ENDIF                                                             SOL00930
C                                                                       SOL00940
C-----------------------------------------------------------------------SOL00950
C- LOOP FOR EACH TIME STEP , ISTEP=CURRENT STEP # ---------------------SOL00960
C-----------------------------------------------------------------------SOL00970
C... ISTEP = STEP NUMBER                                                SOL00980
C    FACTOR= CURRENT TARGET LOAD = FACTOR*(INPUT LOAD)                  SOL00990
C    FACTL = PREVIOUS TRAGET LOAD                                       SOL01000
C    FACT2 = NEXT TRAGET LOAD                                           SOL01010
C    DF    = LOAD STEP                                                  SOL01020
C    DFL   = PREVIOUS LOAD STEP                                         SOL01030
C    DFN   = NEXT LOAD STEP                                             SOL01040
C    FMAX  = MAXIMUN ALLOWABLE INCRIMENTAL LOAD / STEP                  SOL01050
C    DMAX  = MAXIMUN ALLOWABLE INCRIMENTAL DISPLACMENT / STEP           SOL01060
      ISTEP=0                                                           SOL01070
      FACTL=0                                                           SOL01080
      DF=0                                                              SOL01090
      READ(105,*) STEP2,FACT2,FMAX2,DMAX2,NUMB2                         SOL01100
 1000 STEP  =STEP2                                                      SOL01110
      IF (TEST(STEP,'END', FALSE.)) GO TO 5000                          SOL01120
      FACTOR=FACT2                                                      SOL01130
      FMAX  =FMAX2                                                      SOL01140
      DMAX  =DMAX2                                                      SOL01150
      NUMB  =NUMB2                                                      SOL01160
      NUMBDF=NUMB2                                                      SOL01170
      READ(105,*,END=1010) STEP2,FACT2,FMAX2,DMAX2,NUMB2                SOL01180
      GO TO 1020                                                        SOL01190
 1010 STEP2='END'                                                       SOL01200
      FACT2=FACTOR                                                      SOL01210
 1020 IF (TEST(STEP,'END',.FALSE.)) GO TO 5000                          SOL01220
      IF (FACTOR.EQ.FACTL) GO TO 1000                                   SOL01230
C                                                                       SOL01240
 1100   ISTEP=ISTEP+1                                                  SOL01250
        WRITE(STEPID,90) STEP,ISTEP,FACTOR                              SOL01260
        IF (BUG3.OR.BUG5.OR.BUG6.OR.BUG11) THEN                         SOL01270
          WRITE(108,91) STEP,ISTEP,FACTOR                               SOL01280
          WRITE(108,*) 'NEWK: ',NEWK,'   NEWKG: ',NEWKG                 SOL01290
        ENDIF                                                           SOL01300
   90   FORMAT (A,' STEP:',I5," FACTOR:",1P,G11.4)                      SOL01310
   91   FORMAT ('1 ',A,' STEP:',I5,' FACTOR:',1P,G11.4)                 SOL01320
C                                                                       SOL01330
C--------------------------                                             SOL01340
C- CALC INCRIMENTAL LOADS  --                                           SOL01350
C--------------------------                                             SOL01360
C 1) ADD ELEMENT LOADS, AND JOINT LOADS ON FREE JOINTS                  SOL01370
C    SAVE IN DLOA FOR CALCULATING UNBALANCED FORCES FOR THE NEXT STEP,  SOL01380
C 2) PUT JOINT DISPLACEMENTS IN DDISP FOR RESTRAINED JOINTS.            SOL01390
C 3) PUT THE NEGATIVE ELEMENT LOAD IN DLOA FOR RESTRAINED JOINTS, THIS  SOL01400
C    IS USED TO CALCULATE REACTIONS.                                    SOL01410
C 4) ZERO THE UNBALANCED LOAD VECTOR                                    SOL01420
C    IS USED TO CALCULATE REACTIONS.                                    SOL01430
        DFL=DF                                                          SOL01440
        DF =FACTOR-FACTL                                                SOL01450
        DFN=FACT2 -FACTOR                                               SOL01460
        IF (FMAX.EQ.0  AND. DMAX.EQ.0  AND  NUMB.GT.0) DF=DF/NUMB       SOL01470
        IF (DF.EQ.0) GO TO 1000                                         SOL01480
C                                                                       SOL01490
          DO 106 I=L1,L4                                                SOL01500
  106       DLOAD(I)=(Q(I)+R(I))*DF                                     SOL01510
C                                                                       SOL01520
C         ..SWAP THE SIGN OF THE PEP AT REACTIONS                       SOL01530
          DO 110 I=L5,L6                                                SOL01540
            DLOAD(I)=-R(I)*DF                                           SOL01550
  110       DDISP(I)= Q(I)*DF                                           SOL01560
C                                                                       SOL01570
C-------------------------                                              SOL01580
C- FORMULATE STIFFNESS ----                                             SOL01590
C-------------------------                                              SOL01600
C  FORMULATE THE STIFFNESS IF THE FLAG NEWK IS TRUE, THIS FLAG IS       SOL01610
C       INITIALIZED ABOVE, AND SET WHEN THE ELEMENT FORCES ARE CALC.    SOL01620
C       THE STIFFNESS IS ALSO FORMULATED IF KG IS TO BE FORMED          SOL01630
      REPEAT=.TRUE.                                                     SOL01640
      KSAME=0                                                           SOL01650
  100 IF (NEWK.OR.NEWKG) THEN                                           SOL01660
        I1=1                                                            SOL01670
        CALL FORM(I1)                                                   SOL01680
        IF (KGLOAD.EQ.1) NEWKG=.FALSE.                                  SOL01690
      ENDIF                                                             SOL01700
C                                                                       SOL01710
C-----------------------------------------------                        SOL01720
C- MODIFY LOADS FOR RESTRAINT DISPLACEMENTS ---                         SOL01730
C-----------------------------------------------                        SOL01740
      IF (DSPPLG) THEN                                                  SOL01750
C       ----- ZERO INCRIMENTAL REACTION LOAD VECTOR ----                SOL01760
        DO 113 I=L1,L4                                                  SOL01770
  113     DRLOA(I)=0                                                    SOL01780
        CALL REACTN(1,BUG3,NNODE,L1,L4,L5,L6,DRLOA,DDISP,               SOL01790
     &              MD,STIFF,IMD,NDOF,1,1.0,                            SOL01800
     &              MAXNOD,IDOF,COORD,ID,TITLE,STEPID, FALSE.,          SOL01810
     &              NCOS,COSINE,JTCOS,JTFLG,SUMRC)                      SOL01820
C       ----- ADD INCRIMENTAL REACTION LOADS TO OTHER LOADS             SOL01830
        DO 114 I=L1,L4                                                  SOL01840
  114     DLOAD(I) = DLOAD(I) + DRLOA(I)                                SOL01850
      ENDIF                                                             SOL01860
C                                                                       SOL01870
C   SAVE IN LOADS IN DDSP FOR CALCULATING FREE DISPL FOR THIS STEP.     SOL01880
        DO 115 I=L1,L4                                                  SOL01890
  115     DDISP(I)= DLOAD(I)                                            SOL01900
C                                                                       SOL01910
C-------------------------                                              SOL01920
C- SOLVE FOR FREE DISPL -----                                           SOL01930
C-------------------------                                              SOL01940
      IF (BUG5) THEN                                                    SOL01950
        IF (NEWK .OR. NEWKG)                                            SOL01960
     &  CALL LMATRX(STIFF,MD,NDOF,IMD,'GLOBAL STIFFNESS DUMP=7')        SOL01970
        CALL WMATRX(DLOAD,NDOF,NLOAD,'LOAD MATRIX  DUMP=7')             SOL01980
      ENDIF                                                             SOL01990
C                                                                       SOL02000
        IF (NEWK .OR. NEWKG) THEN                                       SOL02010
          CALL MSOLL(1, FALSE.,L4,IMD,NDOF,NLOAD,L1,L4,MD,STIFF,DDISP,  SOL02020
     &               WORK,.FALSE.,MASS,MDMS,1,.FALSE.,KG,MDKG,1)        SOL02030
        ELSE                                                            SOL02040
          CALL MSOLL(2, FALSE.,L4,IMD,NDOF,NLOAD,L1,L4,MD,STIFF,        SOL02050
     &               DDISP,WORX,.FALSE.,MASS,MDMS,1,.FALSE.,KG,MDKG,1)  SOL02060
        ENDIF                                                           SOL02070
        CALL MSOLL(3, FALSE.,L4,IMD,NDOF,NLOAD,L1,L4,MD,STIFF,DDISP,    SOL02080
     &             WORK,.FALSE.,MASS,MDMS,1,.FALSE ,XG,MDKG,1)          SOL02090
C                                                                       SOL02100
      IF (BUG5) CALL WMATRX(DDISP,NDOF,NLOAD,'DISPLACEMENT  DUMP=7')    SOL02110
C                                                                       SOL02120
C-----------------------------------------------                        SOL02130
C- SCALE SOLN SUCH THAT PM LE PMAX, DM LE DMAX --                       SOL02140
C-----------------------------------------------                        SOL02150
      DM=0                                                              SOL02160
      PM=0                                                              SOL02170
      DO 130 I=L3,L4                                                    SOL02180
        PM=MAX(DLOAD(I),-DLOAD(I),PM)                                   SOL02190
  130   DM=MAX(DDISP(I),-DDISP(I),DM)                                   SOL02200
      IF ((DM.GT.DMAX .AND  DMAX.NE.0).OR.                              SOL02210
     &    (PM.GT.PMAX  AND  PMAX.NE.0) ) THEN                           SOL02220
        F1=0                                                            SOL02230
        F2=0                                                            SOL02240
        IF (DMAX.NE.0) F1=DM/DMAX                                       SOL02250
        IF (PMAX.NE.0) F2=PM/PMAX                                       SOL02260
        F=MAX(F1,F2)                                                    SOL02270
        DF=DF/F                                                         SOL02280
        NEWFAC= FALSE                                                   SOL02290
        DO 140 I=L1,L6                                                  SOL02300
          DRLOA(I)=DRLOA(I)/F                                           SOL02310
          DLOAD(I)=DLOAD(I)/F                                           SOL02320
  140     DDISP(I)=DDISP(I)/F                                           SOL02330
          DFDFN=1                                                       SOL02340
      ELSE IF (PMAX.EQ.0  AND. DMAX.EQ.0 .AND. NUMB.GT.1) THEN          SOL02350
        NEWFAC=.FALSE.                                                  SOL02360
        NUMB=NUMB-1                                                     SOL02370
      ELSE                                                              SOL02380
        NEWFAC= TRUE                                                    SOL02390
        DFDFN=DF*DFN                                                    SOL02400
      ENDIF                                                             SOL02410
C                                                                       SOL02420
      AFACT=FACTL+DF                                                    SOL02430
C                                                                       SOL02440
```

```
C-----------------------------------------                                SOL02450
C- SOLVE FOR DISPL DUE TO UNBALANCED FORCES ---                           SOL02460
C-----------------------------------------                                SOL02470
C                                                                         SOL02480
C ..... USE FIRST STIFFNESS TO CALCULATE UNBALANCED FORCES.               SOL02490
C       IF THE STEP IS REPEATED, DO NOT USE THE NEW STIFFNESS.            SOL02500
        IF (UNBAL AND REPEAT) THEN                                        SOL02510
          DO 104 I=L1,L4                                                  SOL02520
            UBLOAD(I) =  FUB(I)                                           SOL02530
  104       UBDISP(I) =  FUB(I)                                           SOL02540
          DO 105 I=L5,L6                                                  SOL02550
            UBLOAD(I) =  FUB(I)                                           SOL02560
  105       UBDISP(I) = 0.00                                              SOL02570
        IF (BUGS) CALL WMATRX(UBDISP,NDOF,NLOAD,'UNBALANCED LOAD MATRIX') SOL02580
          CALL MSOLL(2,.FALSE.,L4,IMD,NDOF,NLOAD,L1,L4,MD,STIFF,UBDISP,   SOL02590
     &             WORK, FALSE.,MASS,MDMS,1,.FALSE.,XG,MDKG,1)            SOL02600
          CALL MSOLL(3,.FALSE.,L4,IMD,NDOF,NLOAD,L1,L4,MD,STIFF,UBDISP,   SOL02610
     &             WORK, FALSE.,MASS,MDMS,1,.FALSE.,XG,MDKG,1)            SOL02620
          IF (BUGS)                                                       SOL02630
     &     CALL WMATRX(UBDISP,NDOF,NLOAD,'UNBALANCED DISPLACEMENT')       SOL02640
        ENDIF                                                             SOL02650
C                                                                         SOL02660
C   ... ADD UNBALANCED FORCES AND DISPL TO STRUCTURAL LOAD AND DISPL.     SOL02670
        IF (UNBAL) THEN                                                   SOL02680
          DO 107 I=L1,L4                                                  SOL02690
           DDISP(I)=DDISP(I)+ UBDISP(I)                                   SOL02700
  107     CONTINUE                                                        SOL02710
          DO 109 I=L5,L6                                                  SOL02720
            DLOAD(I)=DLOAD(I) - UBLOAD(I)                                 SOL02730
  109     CONTINUE                                                        SOL02740
        ENDIF                                                             SOL02750
C                                                                         SOL02760
C..... CALCULATE THE NORM OF THE UNBALANCED FORCES                        SOL02770
        SUM=0                                                             SOL02780
        DO 108 I=L1,L6                                                    SOL02790
          SUM=SUM+FUB(I)**2                                               SOL02800
  108   FUB(I)=0.00                                                       SOL02810
        FNORM = SQRT(SUM/L6)                                              SOL02820
C                                                                         SOL02830
C-------------------------                                                SOL02840
C- CALC PSUDO-VELOCITY ---                                                SOL02850
C-------------------------                                                SOL02860
        IF (ISTEP.EQ.1) THEN                                              SOL02870
          DO 120 I=L1,L6                                                  SOL02880
            VEL(I)=Q(I)*R(I)                                              SOL02890
  120       DVEL(I)=Q(I)*R(I)                                             SOL02900
          PSUVEL=2                                                        SOL02910
          DVEL1 =1                                                        SOL02920
        ELSE IF (FACTOR.EQ.AFACT .AND.                                    SOL02930
     &     NOT ((FACTL.LT.FACTOR .AND.  FACTOR.LT.FACT2) .OR.             SOL02940
     &          (FACTL.GT.FACTOR .AND.  FACTOR.GT.FACT2))) THEN           SOL02950
          PSUVEL=-PSUVEL                                                  SOL02960
          DO 121 I=L1,L6                                                  SOL02970
  121       DVEL(I)= PSUVEL * (Q(I)*R(I))                                 SOL02980
          DVEL1 =1                                                        SOL02990
        ELSE IF (DVEL1.NE.0) THEN                                         SOL03000
          DO 122 I=L1,L6                                                  SOL03010
  122     DVEL(I)=0                                                       SOL03020
          DVEL1 =0                                                        SOL03030
        ENDIF                                                             SOL03040
C                                                                         SOL03050
C-----------------------------------------                                SOL03060
C- CALCULATE INCRIMENTAL MEMBER FORCES ---                                SOL03070
C-----------------------------------------                                SOL03080
C ELE_LIB CALLS THE INDIVIDUAL ELEMENT LIBRARIES, EACH OF WHICH DETERMINSOL03090
C  1) DID THE STIFFNESS CHANGE?  IF SO; NEWK=.TRUE.                       SOL03100
C  2) SHOULD THE INCRIMENTAL DISPLACEMENT BE RECALCULATED?                SOL03110
C     IF SO; KSAME=KSAME-   (A VALUE DEPENDING ON ELEMENT TYPE)           SOL03120
C  3) THE INCRIMENTAL ELEMENT FORCES                                      SOL03130
C  4) THE TOTAL ELEMENT FORCES- STORED IN FUB                             SOL03140
C                                                                         SOL03150
        LSTTYP=0                                                          SOL03160
        PRINT=.FALSE.                                                     SOL03170
        HEAD =.FALSE.                                                     SOL03180
        NEWK=.FALSE.                                                      SOL03190
        DO 150 IELNO=1,NELMT                                              SOL03200
          IOPT=4                                                          SOL03210
          CALL ELELIB                                                     SOL03220
     &  (IOPT,LSTTYP,PRINT,IREL,HEAD,NAME,EESE,EPSE,DAMAGE,DUCFAG,        SOL03230
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDDSP,IZDLOA,MSTOR)      SOL03240
          IOPT=IOPT                                                       SOL03250
  150   CONTINUE                                                          SOL03260
C                                                                         SOL03270
C---------------------------                                              SOL03280
C- REPEAT STEP IF NESSARY ---                                             SOL03290
C---------------------------                                              SOL03300
C  IF AN ELEMENT STIFFNESS CHANGE DICTATED THAT THE STEP BE REANALIZED,   SOL03310
C     REFORMULATE THE STIFFNESS AND REANALIZE                             SOL03320
C     THE VARIABLE REPEAT IS SET UP TO SUPRESS THE REANALIZING OF A STEP  SOL03330
C     CURENTLY, EACH STEP MAY ONLY BE REANALIZED ONCE, THIS PREVENTS      SOL03340
C     AN ENDLESS LOOP.                                                    SOL03350
C                                                                         SOL03360
        IF (KSAME.NE.0 .AND. REPEAT) THEN                                 SOL03370
            REPEAT=.FALSE.                                                SOL03380
            GO TO 100                                                     SOL03390
        ENDIF                                                             SOL03400
C                                                                         SOL03410
C-------------------------                                                SOL03420
C- SOLVE FOR REACTIONS ---                                                SOL03430
C-------------------------                                                SOL03440
        IOPT=2                                                            SOL03450
        CALL REACTN(2,BUG3,NNODE,L1,L4,L5,L6,DLOAD,DDISP,                 SOL03460
     &              MD,STIFF,IMD,NDOF,1,1.0,                              SOL03470
     &              MAXNOD,IDOF,CGORD,ID,TITLE,STEPID,.FALSE.,            SOL03480
     &              NCOS,COSINE,JTCOS,JTFLG,SUMRC)                        SOL03490
C                                                                         SOL03500
C-----------------------------------------------------------              SOL03510
C- SOLVE FOR INCRIMENTAL GEOMETRIC STIFFNESS FORCE, DFG ---               SOL03520
C-----------------------------------------------------------              SOL03530
C                                                                         SOL03540
C------- INCRIMENTAL GEOMETRIC STIFFNESS FORCE                            SOL03550
        IF (KGFORM.EQ.2) THEN                                             SOL03560
          IF (BUGS) WRITE(108,*) '*------ INCRIMENTAL FORCE DUE TO KG '   SOL03570
          CALL TMPLY(BUGS,.TRUE.,IMD,L4,L3,L4,L3,L4,                      SOL03580
     &               MDKG,XG,DFG,DDISP)                                 SOL03590
C         SCALE FORCE TO INLCUDE GROUND ACCELERATION ...                  SOL03600
        ENDIF                                                             SOL03610
C                                                                         SOL03620
C--------------------                                                     SOL03630
C- PRINT STEP SIZE ---                                                    SOL03640
C--------------------                                                     SOL03650
        ISTEP=ISTEP                                                       SOL03660
        IPRNT=IPRNT                                                       SOL03670
        WRITE=(MOD(ISTEP,IPRNT).EQ.0)                                     SOL03680
        IF (WRITE) THEN                                                   SOL03690
          WRITE(108,245) TITLE(1),TITLE(2),STEPID,FACTOR,AFACT,DF,FACTL,  SOL03700
     &                   PMAX,DMAX,FNORM                                  SOL03710
  245   FORMAT ('1 STRUCTURE.. . . ',A,//'  SOLUTION... . ',A,           SOL03720
     &          //,2X,A//'    TARGET FACTOR. . . ',1P,G12.4,             SOL03730
     &          /'  FACTOR . . . ',1P,G12 4,'   INC  FACTOR .   ',G12.4, SOL03740
     &          '  PREV. FACTOR. ',G12 4/                                 SOL03750
     &          '    PMAX . . . . . ',1P,G12.4,'  DMAX.  . . .',G12.4,    SOL03760
     &          '  NORM OF UNBALANCED FORCE ',G12.4)                      SOL03770
          LINE=6                                                          SOL03780
        ENDIF                                                             SOL03790
C                                                                         SOL03800
C-----------------------------------------                                SOL03810
C- CALC AND ENERGIES AND DUCTILITIES  ---                                 SOL03820
C-----------------------------------------                                SOL03830
C                                                                         SOL03840
        IF (WRITE)  LINE=LINE+9                                           SOL03850
C---------- PULL JOINT DISPL FORCES OUT OF LOAD MATRIX ..                 SOL03860
        IF (DSPFLG) THEN                                                  SOL03870
          DO 137 I=L1,L4                                                  SOL03880
            LOAD(I)  = LOAD(I)  - RLOAD(I)                                SOL03890
  137       DLOAD(I) = DLOAD(I) - DRLOA(I)                                SOL03900
        ENDIF                                                             SOL03910
C                                                                         SOL03920
        CALL ENERGY(L1,L2,L3,L4,L5,L6,MAXNOD,NCOS,BUG11,                  SOL03930
     &       IMDMS,MDMS, FALSE.,WRITE,MASS,EIE,ESE,PSE,EKE,EDD,           SOL03940
     &       ALPHA,BETA,XGFORM,NELMT,NNODE,DT,SUMUB,SUMRC,                SOL03950
     &       DSUMRC,SUMFD,DSUMFD,FDAMP,DFDAMP,GA,GV,GD,GAT,               SOL03960
     &       GVT,GDT,WORK,WORK(L6+1),ZERO,ZERO,DACC,DVEL,DDISP,ACC,       SOL03970
     &       VEL,DISP,LOAD,DLOAD,IDOF,JTFLG,COSINE,JTCOS)                 SOL03980
C                                                                         SOL03990
C-----------------------------------------------                          SOL04000
C- GET TOTAL GLOBAL LOADS AND DISPL                 --                    SOL04010
C-----------------------------------------------                          SOL04020
        DO 238 I=L1,L4                                                    SOL04030
          LOAD(I)=LOAD(I)+DLOAD(I) + RLOAD(I)                             SOL04040
          RLOAD(I) = RLOAD(I) + DRLOA(I)                                  SOL04050
          DISP(I)=DISP(I)+DDISP(I)                                        SOL04060
          PG(I)  =PG(I)  +DPG(I)                                          SOL04070
  238   CONTINUE                                                          SOL04080
        DO 240 I=L5,L6                                                    SOL04090
          LOAD(I)=LOAD(I)+DLOAD(I)                                        SOL04100
          DISP(I)=DISP(I)+DDISP(I)                                        SOL04110
          PG(I)  =PG(I)  +DPG(I)                                          SOL04120
  240   CONTINUE                                                          SOL04130
C                                                                         SOL04140
C   ..  GET MAXIMUN LOADS AND DISPL                                       SOL04150
        DO 1007 I=L1,L6                                                   SOL04160
          IF (ABS(DISP(I)).GT.ABS(DSPMAX(I))) DSPMAX(I)=DISP(I)           SOL04170
          IF (ABS(LOAD(I)).GT.ABS(RCTMAX(I))) RCTMAX(I)=LOAD(I)           SOL04180
 1007   CONTINUE                                                          SOL04190
C                                                                         SOL04200
C-----------------------------------                                      SOL04210
C- PRINT TOTAL GLOBAL DISPLACEMENTS ---                                   SOL04220
C-----------------------------------                                      SOL04230
        IF (WRITE) THEN                                                   SOL04240
          IF (IPDISP+LINE.LE.60 ) THEN                                    SOL04250
              LINE=IPDISP+LINE                                            SOL04260
          ELSE                                                            SOL04270
              LINE=IPDISP+6                                               SOL04280
              WRITE(108,250) TITLE(1),TITLE(2),      ISTEP,STEPID         SOL04290
          ENDIF                                                           SOL04300
C                                                                         SOL04310
  250 FORMAT ('1 STRUCTURE..... ',A,/'  SOLUTION.... : ',A,//            SOL04320
     &        2X,  ',  LOADING #',I5,5X,A)                              SOL04330
        CALL JOIDSP(1,MAXNOD,NDOF,NNODE,1,L6,ID,IDOP,CNST,DISP,           SOL04340
     &              TITLE,'DISPLACEMENTS',STEPID,.FALSE ,                 SOL04350
     &              NCOS,COSINE,JTCOS)                                    SOL04360
        ENDIF                                                             SOL04370
C                                                                         SOL04380
C-----------------------------------                                      SOL04390
C- PRINT TOTAL GLOBAL REACTIONS ---                                       SOL04400
C-----------------------------------                                      SOL04410
        IF (WRITE) THEN                                                   SOL04420
          IF (IPRCTN+LINE.LE 60 ) THEN                                    SOL04430
              LINE=IPRCTN+LINE+4                                          SOL04440
          ELSE                                                            SOL04450
              LINE=IPRCTN+8                                               SOL04460
              WRITE(108,250) TITLE(1),TITLE(2),      ISTEP,STEPID         SOL04470
          ENDIF                                                           SOL04480
        ENDIF                                                             SOL04490
C                                                                         SOL04500
        CALL REACTN(3,WRITE,NNODE,L1,L4,L5,L6,LOAD,DISP,                  SOL04510
     &              MD,STIFF,IMD,NDOF,1,1.0,                              SOL04520
     &              MAXNOD,IDOF,CGORD,ID,TITLE,STEPID,.FALSE.,            SOL04530
     &              NCOS,COSINE,JTCOS,JTFLG,SUMRC)                        SOL04540
        SUMFOR=SQRT(SUMRC(1)**2+SUMRC(2)**2+SUMRC(3)**2)                  SOL04550
        SUMMOM=SQRT(SUMRC(4)**2+SUMRC(5)**2+SUMRC(6)**2)                  SOL04560
        SUMMFO=MAX(SUMFOR,SUMMFO)                                         SOL04570
        SUMMMO=MAX(SUMMOM,SUMMMO)                                         SOL04580
        IF (WRITE) WRITE(108,313) SUMFOR,SUMMOM                           SOL04590
  313 FORMAT (/4X,'RESULTANT OF REACTIONS,   FORCE=',1P,G12.4,             SOL04600
     &        '   MOMENT=',G12 4)                                         SOL04610
C                                                                         SOL04620
        DO 1008 I=1,6                                                     SOL04630
          IF (ABS(SUMRC(I)) GT.ABS(SUMRCM(I))) SUMRCM(I)=SUMRC(I)         SOL04640
 1008 CONTINUE                                                            SOL04650
C                                                                         SOL04660
C-----------------------------------------                                SOL04670
C- CALCULATE AND PRINT TOTAL MEMBER FORCES ---                            SOL04680
C-----------------------------------------                                SOL04690
        IOPT=5                                                            SOL04700
        LSTTYP=0                                                          SOL04710
        HEAD =.FALSE.                                                     SOL04720
        IF (WRITE) THEN                                                   SOL04730
            PRINT=.TRUE.                                                  SOL04740
            IF (IPELEM+LINE.LE.60 ) THEN                                  SOL04750
                LINE=IPELEM+LINE                                          SOL04760
             ELSE                                                         SOL04770
                LINE=IPELEM+4                                             SOL04780
                WRITE(108,250) TITLE(1),TITLE(2),      ISTEP,STEPID       SOL04790
             ENDIF                                                        SOL04800
        ELSE                                                              SOL04810
            PRINT=.FALSE.                                                 SOL04820
        ENDIF                                                             SOL04830
        DO 300 IELNO=1,NELMT                                              SOL04840
  300   CALL ELELIB                                                       SOL04850
     &  (IOPT,LSTTYP,PRINT,IREL,HEAD,NAME,EESE,EPSE,DAMAGE,DUCFAG,        SOL04860
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDDSP,IZDLOA,MSTOR)      SOL04870
C                                                                         SOL04880
C-----------------------------------------------                          SOL04890
C- PRINT MEMBER DUCTILITY AND EXCURSION RATIOS ---                        SOL04900
C-----------------------------------------------                          SOL04910
        IOPT=11                                                           SOL04920
        LSTTYP=0                                                          SOL04930
        HEAD =.FALSE.                                                     SOL04940
        IF (WRITE) THEN                                                   SOL04950
            PRINT=.TRUE.                                                  SOL04960
            IF (IPELEM+LINE.LE.60 ) THEN                                  SOL04970
                LINE=IPELEM+LINE                                          SOL04980
             ELSE                                                         SOL04990
                LINE=IPELEM+4                                             SOL05000
                WRITE(108,250) TITLE(1),TITLE(2),      ISTEP,STEPID       SOL05010
             ENDIF                                                        SOL05020
        ELSE                                                              SOL05030
            PRINT=.FALSE.                                                 SOL05040
        ENDIF                                                             SOL05050
        DO 310 IELNO=1,NELMT                                              SOL05060
  310   CALL ELELIB                                                       SOL05070
     &  (IOPT,LSTTYP,PRINT,IREL,HEAD,NAME,EESE,EPSE,DAMAGE,DUCFAG,        SOL05080
     &   IELNO ,IELDOF,KGDOF,PGEOM,RINPUT,AXIAL,IZDDSP,IZDLOA,MSTOR)      SOL05090
C                                                                         SOL05100
C---------------------------------                                        SOL05110
C- WRITE DATA TO OUTPUT FILES  ---                                        SOL05120
C---------------------------------                                        SOL05130
        IF (IWRITE.GT 0 AND MOD(ISTEP,IWRITE).EQ.0) THEN                  SOL05140
          IOPT=2                                                          SOL05150
          CALL DMPDAT(IOPT,IWRITE,TO,DT)                                  SOL05160
        ENDIF                                                             SOL05170
C                                                                         SOL05180
C-------------------------------------------------------------            SOL05190
C- EITHER GET AN NEW FACTOR, OR KEEP WORKING ON THE CURRENT ONE --         SOL05200
C-------------------------------------------------------------            SOL05210
        FACTL=FACTL+DF                                                    SOL05220
        IF (NEWFAC) GO TO 1000                                            SOL05230
        GO TO 1100                                                        SOL05240
C                                                                         SOL05250
C-----------------------------------                                      SOL05260
C- PRINT MAXIMUM  DISPLACEMENTS ---                                       SOL05270
C-----------------------------------                                      SOL05280
 5000   LINE=IPDISP+4                                                     SOL05290
        WRITE(108,250) TITLE(1),TITLE(2),0,'MAXIMUM DISPLACEMENTS'        SOL05300
        CALL JOIDSP(2,MAXNOD,NDOF,NNODE,1,L6,ID,IDOP,CNST,DSPMAX,         SOL05310
     &              TITLE,'DISPLACEMENTS','MAXIMUM VALUES',.FALSE.,       SOL05320
     &              NCOS,COSINE,JTCOS)                                    SOL05330
C                                                                         SOL05340
C-----------------------------------------                                SOL05350
C- PRINT MAXIMUM GLOBAL REACTIONS ---                                     SOL05360
```

```
C-------------------------------------                                  SOL05370
       WRITE(108,250) TITLE(1),TITLE(2),0,'MAXIMUM REACTIONS'          SOL05380
       CALL REACTN(4,.TRUE.,NMODE,L1,L4,L5,L6,RCTMAX,DSPMAX,           SOL05390
     &             MD,STIFF,IMD,NDOF,1,1.0,                          SOL05400
     &             MAXNOD,IDOF,COORD,ID,TITLE,'MAXIMUM VALUES',.FALSE.,SOL05410
     &             NCOS,COSINE,JTCOS,JTFLG,SUMRCM)                   SOL05420
       WRITE(108,5111) SUMMPO,SUMMMO                                   SOL05430
 5111 FORMAT (/4X,'MAXIMUM RESULTANT OF REACTIONS,   FORCE=',1P,G12.4,  SOL05440
     &        '   MOMENT=',G12.4)                                     SOL05450
C                                                                      SOL05460
C--------------------------------                                      SOL05470
C- PRINT MAXIMUM MEMBER FORCES ---                                     SOL05480
C--------------------------------                                      SOL05490
       LSTTYP=0                                                        SOL05500
       HEAD = .FALSE.                                                  SOL05510
       PRINT= .TRUE.                                                   SOL05520
       WRITE(108,250) TITLE(1),TITLE(2),0,'PEAK ELEMENT FORCES'        SOL05530
       DO 5300 IELNO=1,NELMT                                           SOL05540
 5300   CALL ELELIB                                                   SOL05550
     &   (12,LSTTYP,PRINT,IREL,HEAD,NAME ,EESE,EPSE,DAMAGE,DUCFAG,    SOL05560
     &    IELNO ,IELDOP,KGDOP,PGEOM,RINPUT,AXIAL,IZDDSP,IZDLOA,MSTOR)  SOL05570
C                                                                      SOL05580
C-----------------------------------------                             SOL05590
C- PRINT DUCTILITIES AND EXCURSION RATIOS ---                          SOL05600
C-----------------------------------------                             SOL05610
       LSTTYP=0                                                        SOL05620
       HEAD = .FALSE.                                                  SOL05630
       PRINT= .TRUE.                                                   SOL05640
       WRITE(108,250) TITLE(1),TITLE(2),0,                             SOL05650
     &              'PEAK DUCTILITIES AND EXCURSION RATIO''S'         SOL05660
       DO 5310 IELNO=1,NELMT                                           SOL05670
 5310   CALL ELELIB                                                   SOL05680
     &   (11,LSTTYP,PRINT,IREL,HEAD,NAME,EESE,EPSE,DAMAGE,DUCFAG,     SOL05690
     &    IELNO ,IELDOP,KGDOP,PGEOM,RINPUT,AXIAL,IZDDSP,IZDLOA,MSTOR)  SOL05700
C                                                                      SOL05710
C----------------------------                                          SOL05720
C- PRINT DAMAGE INDICIES  ---                                          SOL05730
C----------------------------                                          SOL05740
       LSTTYP=0                                                        SOL05750
       HEAD =.FALSE.                                                   SOL05760
       PRINT= .TRUE.                                                   SOL05770
       WRITE(108,250) TITLE(1),TITLE(2),0,'DAMAGE INDEX '              SOL05780
       DAMAGE=0                                                        SOL05790
       DO 5320 IELNO=1,NELMT                                           SOL05800
         CALL ELELIB                                                   SOL05810
     &   (14,LSTTYP,PRINT,IREL,HEAD,NAME,EESE,EPSE,EDAM,DUCFAG,       SOL05820
     &    IELNO ,IELDOP,KGDOP,PGEOM,RINPUT,AXIAL,IZDDSP,IZDLOA,MSTOR)  SOL05830
 5320    DAMAGE=DAMAGE+EDAM                                            SOL05840
       IF( (ESE-PSE) .EQ. 0 ) RETURN                                   SOL05850
       DAMAGE=DAMAGE/(ESE-PSE)                                         SOL05860
       WRITE(108,5330) DAMAGE                                          SOL05870
 5330 FORMAT (/10X,'STRUCTURE DAMAGE INDEX=',1P,G15.5)                 SOL05880
C                                                                      SOL05890
       RETURN                                                          SOL05900
       END                                                             SOL05910
C-------------------------------------------------------------        SOL00010
       SUBROUTINE SOL05                                                SOL00020
       LOGICAL BTEST,TEST                                              SOL00030
       COMMON /RSA/ ICOMN, RSAP(12)                                    SOL00040
       REAL ICOMN                                                      SOL00050
       CHARACTER*20 ICOM                                               SOL00060
$INCLUDE 'ZCOMN'                                                       SOL00070
C                                                                      SOL00080
C---------------                                                       SOL00090
C- PRINT HEADER -                                                      SOL00100
C---------------                                                       SOL00110
       OPEN(107,FILE='FEM51',FORM='UNFORMATTED',STATUS='UNKNOWN')      SOL00120
       WRITE(108,10) TITLE(1),TITLE(2)                                 SOL00130
   10 FORMAT('1 STRUCTURE.....: ',A,/'  SOLUTION..... ',A,/)           SOL00140
       DO 555 I=1,20                                                   SOL00150
  555    WRITE(108,*)                                                 SOL00160
       WRITE(108,557)                                                  SOL00170
  557  FORMAT(40X,'** RESPONSE SPECTRUM ANALYSIS **',///40X,           SOL00180
     &    '   NOTE: MODAL COMBINATION APPROACH CAN BE SRSS ','OR CQC') SOL00190
C                                                                      SOL00200
C---------------                                                       SOL00210
C- INPUT DATA ---                                                      SOL00220
C---------------                                                       SOL00230
       READ(105,*) ICOM,NMODE,DAMP,NA,NAP,IPRT                         SOL00240
       READ(105,*) NSTEP,MEIG                                          SOL00250
C NOTE: NSTEP =1 : CALCULATE EIGENPAIRS ,STORE EIGENPAIRS IN DISK      SOL00260
C       NSTEP =2 : READ EIGENPAIR FROM DISK AND DO SPECTRUM ANALYSIS   SOL00270
C       NSTEP =3 : CALCULATE EIGENPAIRS AND PERFORM SPECTRUM ANALYSIS, SOL00280
C                  EIGENPAIRS ARE NOT STORED IN DISK                   SOL00290
C                                                                      SOL00300
       WRITE(108,10) TITLE(1),TITLE(2)                                 SOL00310
       IF( TEST(ICOM,'SRSS' ,.FALSE.)) THEN                            SOL00320
         ICOMN=1                                                       SOL00330
         WRITE(108,11) 'SRSS MODAL COMBINATION METHOD',                SOL00340
     &                 '----------------------------'                 SOL00350
       ELSE IF( TEST(ICOM,'CQC' ,.FALSE.)) THEN                        SOL00360
         ICOMN=2                                                       SOL00370
         WRITE(108,11) 'CQC MODAL COMBINATION METHOD',                 SOL00380
     &                 '----------------------------'                 SOL00390
       ELSE                                                            SOL00400
         WRITE(108,*) 'INVALID MODAL COMBINATION METHOD'               SOL00410
         WRITE(108,11) ICOM,'--------------------'                     SOL00420
         RETURN                                                        SOL00430
       ENDIF                                                           SOL00440
C                                                                      SOL00450
       WRITE(108,12) NMODE,DAMP,NA,NAP                                 SOL00460
   11 FORMAT(1X,'SOLUTION #5, ',A,/1X,'--------------',A,/)            SOL00470
   12 FORMAT(1X,' NUMBER OF MODES CONSIDERED ..... ..  ..',0P,I4,/,    SOL00480
     &       1X,' DAMPING RATIO FOR ALL MODES ..... ..',0P,G12.3,/     SOL00490
     &       1X,' NO. OF SEISMIC COMPONENTS CONSIDERED  ',0P,I4,/,     SOL00500
     &       1X,' NO  OF INPUT POINTS FOR SPECTRUM ......',0P,I4)     SOL00510
C                                                                      SOL00520
       IF(NSTEP .EQ. 1 ) THEN                                          SOL00530
       WRITE(108,13) NSTEP,MEIG                                        SOL00540
   13 FORMAT(1X,' NSTEP ..... ....',0P,I4,/,                           SOL00550
     &       1X,' EIGEN-SOLUTION IS STORE IN DISK UNIT ..',0P,I4)      SOL00560
       ELSE IF(NSTEP .EQ. 2 ) THEN                                     SOL00570
       WRITE(108,14) NSTEP,MEIG                                        SOL00580
   14 FORMAT(1X,' NSTEP ..... ....',0P,I4,/,                           SOL00590
     &       1X,' EIGEN-SOLUTION IS READ FROM DISK UNIT .',0P,I4)      SOL00600
       ENDIF                                                           SOL00610
       ELSTIC=.TRUE.                                                   SOL00620
C                                                                      SOL00630
C      ---- INITIALIZE STORAGE                                         SOL00640
C      I(IZEVAL) = LOCATION OF THE EIGENVALUES                         SOL00650
C      I(IZEVEC) = LOCATION OF THE EIGENVECTORS                        SOL00660
C      I(IZA   ) = LOCATION OF THE SYMMETRIC STIFFNESS                 SOL00670
C      I(IZB   ) = LOCATION OF THE SYMMETRIC MASS                      SOL00680
C      I(IZLOAD) = LOCATION OF THE TOTAL LOAD MATRIX                   SOL00690
C      I(IZDISP) = LOCATION OF THE TOTAL DISPLACEMENT MATRIX           SOL00700
C                                                                      SOL00710
       IZEVAL = IZ                                                     SOL00720
       IZ     = IZ+NFREE                                               SOL00730
       IZEVEC = IZ                                                     SOL00740
       IZ     = IZ+NFREE*NFREE                                         SOL00750
       IZA    = IZ                                                     SOL00760
       IZ     = IZ+NFREE*NFREE                                         SOL00770
       IZB    = IZ                                                     SOL00780
       IZ     = IZ+NFREE*NFREE                                         SOL00790
       IF(IZLOAD .EQ. 0 ) THEN                                         SOL00800
          IZLOAD=IZ                                                    SOL00810
          IZ=IZ + NDOF                                                 SOL00820
          CALL CKSTOR(0,0)                                             SOL00830
          DO 1 I=1,NDOF                                                SOL00840
     1      Z(I+IZLOAD-1)=0                                           SOL00850
       ENDIF                                                           SOL00860
C                                                                      SOL00870
       IF(IZDISP .EQ. 0 ) THEN                                         SOL00880
          IZDISP=IZ                                                    SOL00890
          IZ=IZ + NDOF                                                 SOL00900
          CALL CKSTOR(0,0)                                             SOL00910
          DO 2 I=1,NDOF                                                SOL00920
     2      Z(I+IZDISP-1)=0                                           SOL00930
       ENDIF                                                           SOL00940
C                                                                      SOL00950
       IF(IZFUB .EQ. 0 ) THEN                                          SOL00960
          IZFUB=IZ                                                     SOL00970
          IZ=IZ + NDOF                                                 SOL00980
          CALL CKSTOR(0,0)                                             SOL00990
          DO 3 I=1,NDOF                                                SOL01000
     3      Z(I+IZFUB-1)=0                                            SOL01010
       ENDIF                                                           SOL01020
       IF (IZVEL.EQ 0) THEN                                            SOL01030
        IZVEL=IZ                                                       SOL01040
        IZ   =IZ+NDOF                                                  SOL01050
        DO 5 I=1,NDOF                                                  SOL01060
5         Z(I+IZVEL-1)=0                                               SOL01070
       ENDIF                                                           SOL01080
C                                                                      SOL01090
       IF (IZDVEL.EQ.0) THEN                                           SOL01100
        IZDVEL=IZ                                                      SOL01110
        IZ   =IZ+NDOF                                                  SOL01120
        DO 7 I=1,NDOF                                                  SOL01130
7         Z(I+IZDVEL-1)=0.                                             SOL01140
       ENDIF                                                           SOL01150
C---- ----DEFINE STORAGE FOR THE MODAL COMBINATION                     SOL01160
       IZRA=IZ                                                         SOL01170
       IZ    = IZ+NMODE*3                                              SOL01180
       IZRD=IZ                                                         SOL01190
       IZ    = IZ+NMODE*3                                              SOL01200
       IZFLU=IZ                                                        SOL01210
       IZ    = IZ+NFREE*3                                              SOL01220
       IZCMC=IZ                                                        SOL01230
       IZ    = IZ+NMODE*NMODE                                          SOL01240
       IZGM =IZ                                                        SOL01250
       IZ    = IZ+NMODE                                                SOL01260
       IZELF=IZ                                                        SOL01270
       IZ    = IZ+NMODE*3                                              SOL01280
       IZCQC=IZ                                                        SOL01290
       IZ    = IZ+(NCOND *NFREE)*NMODE                                 SOL01300
       IZDLD=IZ                                                        SOL01310
       IZ    = IZ+NDOF*NA                                              SOL01320
       IZELEF=IZ                                                       SOL01330
       IZ    = IZ+NELMT*12*(NMODE+NA)                                  SOL01340
       IZRCT=IZ                                                        SOL01350
       IZ    = IZ+6*(NMODE+NA)                                         SOL01360
       IZNAP=IZ                                                        SOL01370
       IZ    = IZ+2*NAP                                                SOL01380
       IZTMP2=IZ                                                       SOL01390
       IZ    = IZ+NFREE                                                SOL01400
       IZEEF=IZ                                                        SOL01410
       IZ    = IZ+NFREE*NA                                             SOL01420
       IZCQ2=IZ                                                        SOL01430
       IZ    = IZ+NFREE*NMODE                                          SOL01440
C                                                                      SOL01450
        IZTMP= IZ                                                      SOL01460
        IZ   = IZ+NDOF                                                 SOL01470
        IW   =IZ                                                       SOL01480
C                                                                      SOL01490
       CALL CKSTOR(NDOF,IW)                                            SOL01500
C                                                                      SOL01510
       IMD  =NZ(IZMD  +NDOF)                                           SOL01520
       IMDMS=NZ(IZMDMS+NDOF)                                           SOL01530
       IMDKG=NZ(IZMDKG+NDOF)                                           SOL01540
C                                                                      SOL01550
C---------- FORM MASS MATRIX -----                                     SOL01560
C                                                                      SOL01570
       DO 400 I=1,IMDMS                                                SOL01580
  400    Z(I+IZMASS-1)=0.                                             SOL01590
C                                                                      SOL01600
       CALL MFORM(2,INMASS,NMASS,Z(IZMASS),NZ(IZMDMS),MD,NDOF,IMDMS,   SOL01610
     &       NMODE,MAXNOD,BTEST(IBUG,5),IBUG,NZ(IZID),NZ(IZIDOF),      SOL01620
     &       Z(IZCORST),NZ(IZJCOS),Z(IZMDAT),Z(IW),NZ(IW+21),Z(IW+27),  SOL01630
     &       Z(IW+ 63),Z(IW+ 99),TITLE)                                SOL01640
C                                                                      SOL01650
       CALL SOL05A(IMD     ,IMDMS     ,IMDKG     , NMODE    , NCOS     , SOL01660
     &             IZDISP   ,IZLOAD   ,MCOND     , NELMT    , MAXNOD   , SOL01670
     &             NDOF     ,NCOND    ,NFREE     , TITLE    , STEPID   , SOL01680
     &             Z(IZSTIF),Z(IZMASS) ,Z(IZKG  ) , Z(IZA   ), Z(IZB   ), SOL01690
     &            NZ(IZMD  ),NZ(IZMDMS),NZ(IZMDKG), Z(IZDISP), Z(IZTMP ), SOL01700
     &             NA,NAP    ,NMODE     ,DAMP      , IPRT     , NSTEP    , SOL01710
     &            NZ(IZID  ),NZ(IZIDOF),NZ(IZJFLG), Z(IZCORD), IBUG     , SOL01720
     &             Z(IZCOS ),NZ(IZJCOS),Z(IZEVAL) , Z(IZEVEC), Z(IZRA  ), SOL01730
     &             Z(IZRD  ),Z(IZFLU  ),Z(IZCMC  ) , Z(IZGM  ), Z(IZELEF), SOL01740
     &             Z(IZELF ),Z(IZLOAD),Z(IZCQC  ) , Z(IZDLD  ), Z(IZRCT ), SOL01750
     &             Z(IZNAP ),Z(IZTMP2),Z(IZEEF  ) , Z(IZCQ2  ), Z(IW    )) SOL01760
C                                                                      SOL01770
       RETURN                                                          SOL01780
       END                                                             SOL01790
C                                                                      INTP0010
C ----- ------------------------------------------------------- ---    INTP0020
C                                                                      INTP0030
       SUBROUTINE INTPO(PERIOD,RSV,NAP,RSNP)                           INTP0040
       DIMENSION RSNP(2,NAP)                                           INTP0050
C---------------------------------------                               INTP0060
C -- INTERPORATION OF RESPONSE SPECTRUM VALUE                           INTP0070
C---------------------------------------                               INTP0080
C                                                                      INTP0090
       IF( PERIOD .LT. RSNP(1,1) ) THEN                                INTP0100
             RSV=RSNP(2,1)                                             INTP0110
             RETURN                                                    INTP0120
       ELSE IF( PERIOD .GT. RSNP(1,NAP) ) THEN                         INTP0130
             RSV=RSNP(2,NAP)                                           INTP0140
             RETURN                                                    INTP0150
       ENDIF                                                           INTP0160
C                                                                      INTP0170
       I=0                                                             INTP0180
  300  I=I+1                                                           INTP0190
       J=I+1                                                           INTP0200
       IF( PERIOD .GE. RSNP(1,I) .AND  PERIOD .LT  RSNP(1,J) ) THEN    INTP0210
           COEF=(RSNP(2,J)-RSNP(2,I))/(RSNP(1,J)-RSNP(1,I))            INTP0220
           RSV=(PERIOD-RSNP(1,I))*COEF + RSNP(2,I)                     INTP0230
       ELSE IF( PERIOD .GE. RSNP(1,J)) THEN                            INTP0240
           GO TO 300                                                   INTP0250
       ENDIF                                                           INTP0260
C                                                                      INTP0270
       RETURN                                                          INTP0280
       END                                                             INTP0290
C-----------------------------------------------------------------------SOL00010
       SUBROUTINE SOL05A(IMD , IMDMS , IMDKG , NMODE , NCOS  ,         SOL00020
     &                   IZDISP , IZLOAD , MCOND , NELMT , MAXNOD ,    SOL00030
     &                   NDOF , NCOND , NFREE , TITLE , STEPID ,       SOL00040
     &                   STIFF , MASS , KG , A , B ,                   SOL00050
     &                   MD , MDMS , MDKG , DISP , TMP ,               SOL00060
     &                   NA,NAP , NMODE , DAMP , IPRT , NSTEP ,        SOL00070
     &                   ID , IDOF , JTFLG , COORD , IBUG ,            SOL00080
     &                   COSINE , JTCOS , EVAL , EVEC , RA ,           SOL00090
     &                   RD , FLU , CMC , GM , ELEF ,                  SOL00100
     &                   ELF , LOAD , CQCM , DLOAD , RCT ,             SOL00110
     &                   RSNP , TMP2 , EEF , CQCM2 , WORK )            SOL00120
C                                                                      SOL00130
       COMMON/RSA/ICOMN,RSAP(12)                                       SOL00140
       REAL ICOMN,LOAD,MASS,KG                                         SOL00150
       LOGICAL TEST,BTEST,MCOND,PRIN0,PRIN5,AXIAL                      SOL00160
       CHARACTER NAME*1,OPTION*40                                      SOL00170
       CHARACTER*(*) TITLE(2),STEPID                                   SOL00180
       DIMENSION STIFF(IMD),MASS(IMDMS),KG(IMDKG),RINPUT(100)          SOL00190
       DIMENSION A(NFREE,NFREE),B(NFREE,NFREE),CQCM(NCOND*NFREE,NMODE) SOL00200
       DIMENSION MD(NDOF*1),MDMS(NDOF*1),MDKG(NDOF-1),DISP(NDOF)       SOL00210
       DIMENSION EVAL(NFREE),EVEC(NFREE,NFREE),SUBRCT(6),TMP(NDOF)     SOL00220
       DIMENSION COORD(MAXNOD,3),JTFLG(MAXNOD),JTCOS(MAXNOD)           SOL00230
       DIMENSION IDOF(MAXNOD,6),COSINE(3,3,NCOS)                       SOL00240
       DIMENSION TMP2(NFREE),EEF(NFREE,NA),CQCM2(NFREE,NMODE)          SOL00250
       DIMENSION V1(3),V2(3),V3(3),RCT(6,NMODE+NA),RSNP(2,NAP)         SOL00260
       DIMENSION RA(NMODE,3),LOAD(NDOF),RD(NMODE,3),DLOAD(NDOF,NA)     SOL00270
       DIMENSION FLU(NFREE,3),CMC(NMODE,NMODE),ID(MAXNOD)              SOL00280
       DIMENSION GM(NMODE),ELF(NMODE,3),WORK(NCOND*NFREE)              SOL00290
```

```
      DIMENSION ELEP(NELMT,12,NMODE*NA)                                 SOL00300
C                                                                       SOL00310
C     ---- SET UP CONSTANTS -- ------- ---                              SOL00320
      L1=1                                                              SOL00330
      L2=NCOND                                                          SOL00340
      L3=L2+1                                                           SOL00350
      L4=NCOND +NFREE                                                   SOL00360
      L5=L4+1                                                           SOL00370
      L6=NDOF                                                           SOL00380
      PRIN5=BTEST(IBUG,5)                                               SOL00390
C------------------------                                               SOL00400
C- FORMULATE STIFFNESS ----                                             SOL00410
C------------------------                                               SOL00420
      CALL FORM(1)                                                      SOL00430
      IF(PRIN5) CALL LMATRX(STIFF,MD,NDOF,IMD,'GLOBAL STIFFNESS MATRIX')SOL00440
C                                                                       SOL00450
C-------------------------------                                        SOL00460
C- CONDENSE OUT NON-FREE DOF ---                                        SOL00470
C-------------------------------                                        SOL00480
C     IF MCOND IS TRUE THE MASS MATRIX IS ALSO REDUCED BY GUYAN REDUCTIONSOL00490
      IF (NCOND.GT 0) THEN                                              SOL00500
C       SET UP ZERO LOAD VECTOR                                         SOL00510
        DO 28 I=1,NDOF                                                  SOL00520
   28   TMP(I)=0.                                                       SOL00530
        CALL MSOLL(1,BTEST(IBUG,6),L4,IMD,NDOF,1,L1,L2,MD,STIFF,TMP,    SOL00540
     &             WORK,MCOND,MASS,MDMS,IMDMS,.FALSE.,KG,MDKG,IMDKG)    SOL00550
      ENDIF                                                             SOL00560
C                                                                       SOL00570
C-------------------------------------                                  SOL00580
C- TRANSFER STIFFNESS INTO A MATRIX ---                                 SOL00590
C-------------------------------------                                  SOL00600
C  NOTE: IMSL ROUTINE GVCSP IS USED TO SOLVE FOR EIGENVALUES,           SOL00610
C        THIS ROUTINE USES A 'FULL SYMMETRIC MATRIX'.                   SOL00620
C                                                                       SOL00630
      DO 600 J=L3,L4,+1                                                 SOL00640
        DO 600 I=J,L3,-1                                                SOL00650
          IJ=MD(J)+J-I                                                  SOL00660
          IF (IJ .LT. MD(J+1) .AND. IJ .GE. MD(J) ) THEN                SOL00670
              A(I-L3+1,J-L3+1)=STIFF(IJ)                                SOL00680
              A(J-L3+1,I-L3+1)=A(I-L3+1,J-L3+1)                         SOL00690
          ELSE IF( IJ .GE. MD(J+1) ) THEN                               SOL00700
              A(I-L3+1,J-L3+1)=0                                        SOL00710
              A(J-L3+1,I-L3+1)=0                                        SOL00720
          ENDIF                                                         SOL00730
C                                                                       SOL00740
          IF (PRIN5) WRITE(108,29) I,J,A(I-L3+1,J-L3+1)                 SOL00750
   29     FORMAT (' I ',I5,' J:',I5,' A(I-L3+1,J-L3+1)',1P,G15.5)       SOL00760
C                                                                       SOL00770
C---------- CHECK FOR ZERO'S ON MAIN DIAGONAL                           SOL00780
          IF (I.EQ.J .AND. A(I-L3+1,J-L3+1) LE.0 ) THEN                 SOL00790
           WRITE(108,32) I,A(I-L3+1,J-L3+1)                             SOL00800
           STOP                                                         SOL00810
          ENDIF                                                         SOL00820
C                                                                       SOL00830
  600   CONTINUE                                                        SOL00840
   32 FORMAT (5X,'DOF #',I5,' IS .LE. 0 IN THE STIFFNESS MATRIX   ',    SOL00850
     &        5X,'K(I,I)=',1P,G12.4,'  SOLN IS ABORTED')                SOL00860
C                                                                       SOL00870
C--------------------------------                                       SOL00880
C- TRANSFER MASS INTO B MATRIX ---                                      SOL00890
C--------------------------------                                       SOL00900
      DO 700 J=L3,L4,+1                                                 SOL00910
        DO 700 I=J,L3,-1                                                SOL00920
          IJ=MDMS(J)+J-I                                                SOL00930
          IF (IJ .LT. MDMS(J+1)  AND. IJ .GE. MDMS(J) ) THEN            SOL00940
              B(I-L3+1,J-L3+1)=MASS(IJ)                                 SOL00950
              B(J-L3+1,I-L3+1)=B(I-L3+1,J-L3+1)                         SOL00960
          ELSE IF( IJ .GE. MDMS(J+1) ) THEN                             SOL00970
              B(I-L3+1,J-L3+1)=0.                                       SOL00980
              B(J-L3+1,I-L3+1)=0.                                       SOL00990
          ENDIF                                                         SOL01000
          IF (PRIN5) WRITE(108,39) I,J,B(I-L3+1,J-L3+1)                 SOL01010
   39     FORMAT (' I:',I5,' J:',I5,' B(I-L3+1,J-L3+1):', 1P,G15.5)     SOL01020
C                                                                       SOL01030
C--------- CHECK FOR ZERO'S ON MAIN DIAGONAL                           SOL01040
          IF (I.EQ.J .AND. B(I-L3+1,J-L3+1) LE.0 ) THEN                 SOL01050
           WRITE(108,42) I,B(I-L3+1,J-L3+1)                             SOL01060
           STOP                                                         SOL01070
          ENDIF                                                         SOL01080
C                                                                       SOL01090
  700 CONTINUE                                                          SOL01100
   42 FORMAT (5X,'DOF #',I5,' IS  LE. 0 IN THE MASS MATRIX. .',         SOL01110
     &        5X,'M(I,I)=',1P,G12.4,'  SOLN IS ABORTED')                SOL01120
C                                                                       SOL01130
C---------------------------------------                                SOL01140
C- SOLVE FOR EIGENVALUES AND EIGENVECTORS ---                           SOL01150
C---------------------------------------                                SOL01160
C------- SU_BROUTINE GVCSP ------------------------------------------- SOL01170
C  EIGENVALUES AND EIGENVECTORS OF A*X-LAMBDA*B*X=0                     SOL01180
C  A  - STIFFNESS - REAL, FULL SYMMETRIC MATRIX                         SOL01190
C  B  - MASS - REAL, FULL SYMMETRIC MATRIX                              SOL01200
C  N  - NUMBER OF DOF OF A AND B                                        SOL01210
C  EVAL- EIGENVALUES , N BY 1                                           SOL01220
C  EVEC- EIGENVECTORS, N BY N                                           SOL01230
C------------------------------------------------------------------     SOL01240
C                                                                       SOL01250
      IF (NSTEP.EQ.1.OR.NSTEP.EQ.3) THEN                                SOL01260
        OPEN(109,FILE='FEMSC',FORM='UNFORMATTED',STATUS='UNKNOWN')      SOL01270
        WRITE(109) A,B                                                  SOL01280
        CALL GVCSP(NFREE,A,B,EVAL,EVEC,WORK,TMP2,TMP)                   SOL01290
        REWIND 109                                                      SOL01300
        READ(109) A,B                                                   SOL01310
        DO 15 I=1,NFREE                                                 SOL01320
        DO 15 J=1,NFREE                                                 SOL01330
15      A(I,J)=A(I,J)/SQRT(B(I,I)*B(J,J))                               SOL01340
        CLOSE (109)                                                     SOL01350
C                                                                       SOL01360
C-------------------------------------------                            SOL01370
C- NORMALIZE MODE SHAPE TO MAX VALUE OF ONE ---                         SOL01380
C-------------------------------------------                            SOL01390
C                                                                       SOL01400
C NOTE  THE EIGENVECTOR IN IMSL DGVCSP HAS BEEN NORMALIZED              SOL01410
C                                                                       SOL01420
C-----WRITE EIGENVALUES AND EIGENVECTORS TO DISK                        SOL01430
        WRITE(107) (EVAL(I),I=1,IPRT)                                   SOL01440
        WRITE(107) ((EVEC(I,J),J=1,IPRT),I=1,NFREE)                     SOL01450
C                                                                       SOL01460
      ELSE IF(NSTEP  EQ. 2 ) THEN                                       SOL01470
        REWIND 107                                                      SOL01480
        READ(107) (EVAL(I),I=1,IPRT)                                    SOL01490
        READ(107) ((EVEC(I,J),J=1,IPRT),I=1,NFREE)                      SOL01500
      ENDIF                                                             SOL01510
C                                                                       SOL01520
C---------------------------------------                                SOL01530
C- PRINT EIGENVALUES AND EIGENVECTORS ---                               SOL01540
C---------------------------------------                                SOL01550
      IST =1                                                            SOL01560
  100 IEND=MIN(IST+6,IPRT)                                              SOL01570
C                                                                       SOL01580
      WRITE(108,110) (I,I=IST,IEND)                                     SOL01590
      DO 125 I=IST,IEND                                                 SOL01600
  125   EVAL(I)=SQRT( EVAL(I) )                                         SOL01610
      WRITE(108,126) (EVAL(I),I=IST,IEND)                               SOL01620
      DO 127 I=IST,IEND                                                 SOL01630
  127   EVAL(I)= EVAL(I)  / 6.2831853                                   SOL01640
      WRITE(108,128) (EVAL(I),I=IST,IEND)                               SOL01650
      DO 129 I=IST,IEND                                                 SOL01660
        IF (EVAL(I).LT.1.E-10) EVAL(I)=1.E-10                           SOL01670
  129   EVAL(I)= 1.00 / EVAL(I)                                         SOL01680
      WRITE(108,130) (EVAL(I),I=IST,IEND)                               SOL01690
      WRITE(108,135)                                                    SOL01700
  110 FORMAT(/'          MODE:',7(I10,5X))                             SOL01710
  126 FORMAT(' FREQ (RAD/SEC):',1P,7(G14.5,1X) )                        SOL01720
  128 FORMAT(' FREQUENCY (HZ):',1P,7(G14.5,1X) )                        SOL01730
  130 FORMAT('   PERIOD (SEC) ',1P,7(G14.5,1X) )                        SOL01740
  135 FORMAT('   EIGENVECTORS:')                                        SOL01750
C                                                                       SOL01760
C-------- EIGENVECTORS                                                 SOL01770
      DO 155 I=1,NFREE                                                  SOL01780
155   WRITE(108,150) I-NCOND,(EVEC(I,J),J=IST,IEND)                     SOL01790
150   FORMAT('      DOF(',I5,')',1P,7(G14.5,1X) )                       SOL01800
C                                                                       SOL01810
      IF (IEND.LT.IPRT ) THEN                                           SOL01820
        WRITE(108,10) TITLE(1),TITLE(2)                                 SOL01830
        IST=IEND+1                                                      SOL01840
        GO TO 100                                                       SOL01850
      ENDIF                                                             SOL01860
   10 FORMAT('1 STRUCTURE..    ',A,/'  SOLUTION..    . ',A,//)          SOL01870
C                                                                       SOL01880
      IF (NSTEP  EQ. 1 ) STOP                                           SOL01890
C                                                                       SOL01900
C-----ZERO GLOBAL BASE ACCELERATION AND DISPLACEMENT SPECTRUM VALUES    SOL01910
      DO 771 I=1,NMODE                                                  SOL01920
      DO 771 J=1,3                                                      SOL01930
        RA(I,J)=0.                                                      SOL01940
  771   RD(I,J)=0.                                                      SOL01950
C                                                                       SOL01960
C---------------------------------------------------------------------- SOL01970
C-- INPUT NUMBER OF SEISMIC COMPONENTS, SCALING FACTOR ,PRINT CONTROL - SOL01980
C---------------------------------------------------------------------- SOL01990
C                                                                       SOL02000
      READ(105,*) ASCALE,OPTION,PRIN0                                   SOL02010
C                                                                       SOL02020
C-----INPUT DIRECTION VECTORS V1 AND V2                                 SOL02030
      READ(105,*) V1,V2                                                 SOL02040
C                                                                       SOL02050
C------- CALCULATE V3 = V1 X V2                                        SOL02060
      CALL CROSS(V3,V1,V2,0)                                            SOL02070
C                                                                       SOL02080
C--------CALCULATE V2 = V3 X V1, AND NORMALIZE ALL VECTORS ..          SOL02090
      CALL CROSS(V2,V3,V1,1)                                            SOL02100
C---------------------------------                                      SOL02110
C-- INPUT RESPONSE SPECTRUM VALUES --                                   SOL02120
C---------------------------------                                      SOL02130
      WRITE(108,10) TITLE(1),TITLE(2)                                   SOL02140
      DO 30 I=1,NA                                                      SOL02150
        READ(105,*) IDIR                                                SOL02160
C                                                                       SOL02170
        IF (IDIR.GE.1 .AND. IDIR.LE.3) THEN                             SOL02180
          IF (TEST(OPTION,'ACC',.FALSE. )) THEN                         SOL02190
            WRITE(108,107) I                                            SOL02200
  107       FORMAT (//1X,'** ACCELERATION SPECTRUM VALUES :',           SOL02210
     &          'INPUTING TRANSLATIONAL #',I2,/)                        SOL02220
          ELSE IF(TEST(OPTION,'DISP',.FALSE. )) THEN                    SOL02230
            WRITE(108,1070) I                                           SOL02240
 1070       FORMAT (//1X,'** DISPLACEMENT SPECTRUM VALUES .',           SOL02250
     &          'INPUTING TRANSLATIONAL #',I2,/)                        SOL02260
          ENDIF                                                         SOL02270
        ELSE                                                            SOL02280
          WRITE(108,*) 'INVALID DIRECTION:',IDIR,' SET 0 < IDIR < 4'    SOL02290
          GO TO 30                                                      SOL02300
        ENDIF                                                           SOL02310
C                                                                       SOL02320
        IF (IDIR.EQ.1) THEN                                             SOL02330
          CX=V1(1)                                                      SOL02340
          CY=V1(2)                                                      SOL02350
          CZ=V1(3)                                                      SOL02360
          WRITE(108,92) V1                                              SOL02370
        ELSE IF (IDIR.EQ 2) THEN                                        SOL02380
          CX=V2(1)                                                      SOL02390
          CY=V2(2)                                                      SOL02400
          CZ=V2(3)                                                      SOL02410
          WRITE(108,92) V2                                              SOL02420
        ELSE IF (IDIR.EQ.3) THEN                                        SOL02430
          CX=V3(1)                                                      SOL02440
          CY=V3(2)                                                      SOL02450
          CZ=V3(3)                                                      SOL02460
          WRITE(108,92) V3                                              SOL02470
        ENDIF                                                           SOL02480
C                                                                       SOL02490
        READ(105,*) (RSNP(1,II),RSNP(2,II),II=1,NAP)                    SOL02500
        WRITE(108,640)                                                  SOL02510
  640   FORMAT(30X,'PERIOD         ',10X,'SPECTRUM VALUE',/             SOL02520
     &         30X,'____________',10X,'_____________')                  SOL02530
        WRITE(108,907) (RSNP(1,II),RSNP(2,II),II=1,NAP)                 SOL02540
  907   FORMAT(30X,G14.6,10X,G14.6)                                     SOL02550
C                                                                       SOL02560
          DO 20 J=1,NMODE                                               SOL02570
           CALL INTPO(EVAL(J),TMP(J),NAP,RSNP)                          SOL02580
           IF (TEST(OPTION,'ACC',.FALSE. ) ) THEN                       SOL02590
            ACC=TMP(J)*ASCALE                                           SOL02600
            DIS=TMP(J)*ASCALE/((6.2831853/EVAL(J))**2)                  SOL02610
           ELSE IF (TEST(OPTION,'DISP',.FALSE. ) ) THEN                 SOL02620
            DIS=TMP(J)*ASCALE                                           SOL02630
            ACC=TMP(J)*ASCALE*((6.2831853/EVAL(J))**2)                  SOL02640
           ENDIF                                                        SOL02650
           RA(J,1)=RA(J,1) + ACC*CX                                     SOL02660
           RA(J,2)=RA(J,2) + ACC*CY                                     SOL02670
           RA(J,3)=RA(J,3) + ACC*CZ                                     SOL02680
           RD(J,1)=RD(J,1) + DIS*CX                                     SOL02690
           RD(J,2)=RD(J,2) + DIS*CY                                     SOL02700
           RD(J,3)=RD(J,3) + DIS*CZ                                     SOL02710
   20      CONTINUE                                                     SOL02720
C                                                                       SOL02730
   30 CONTINUE                                                          SOL02740
      IF (PRIN0) THEN                                                   SOL02750
        WRITE(108,31) 'X'                                               SOL02760
        WRITE(108,93) (RA(J,1),J=1,NMODE)                               SOL02770
        WRITE(108,31) 'Y'                                               SOL02780
        WRITE(108,93) (RA(J,2),J=1,NMODE)                               SOL02790
        WRITE(108,31) 'Z'                                               SOL02800
        WRITE(108,93) (RA(J,3),J=1,NMODE)                               SOL02810
   31   FORMAT(/1X,'GLOBAL BASE ACCE.-SPECTRUM VALUES IN THE ',A,       SOL02820
     &         ' DIRECTION ',/)                                         SOL02830
        WRITE(108,319) 'X'                                              SOL02840
        WRITE(108,93) (RD(J,1),J=1,NMODE)                               SOL02850
        WRITE(108,319) 'Y'                                              SOL02860
        WRITE(108,93) (RD(J,2),J=1,NMODE)                               SOL02870
        WRITE(108,319) 'Z'                                              SOL02880
        WRITE(108,93) (RD(J,3),J=1,NMODE)                               SOL02890
  319   FORMAT(/1X,'GLOBAL BASE DISP.-SPECTRUM VALUES IN THE ',A,       SOL02900
     &         ' DIRECTION ',/)                                         SOL02910
      ENDIF                                                             SOL02920
C                                                                       SOL02930
C-----------------------------------------                              SOL02940
C-- CALCULATE INFLUENCE COEFFICIENT VECTORS --                          SOL02950
C-----------------------------------------                              SOL02960
      DO 440 I=1,NDOF                                                   SOL02970
      DO 440 J=1,3                                                      SOL02980
  440 PLU(I,J)=0.                                                       SOL02990
C                                                                       SOL03000
      DO 451 K=1,3                                                      SOL03010
C                                                                       SOL03020
C   ... OMIT ALL DOF THAT ARE CONSTRAINED TO A MASTER DOF...            SOL03030
       DO 554 NODE=1,NNODE                                              SOL03040
        ICOS=JTCOS(NODE)                                                SOL03050
        DO 650 JDIR=1,3                                                 SOL03060
         IF (.NOT. BTEST(JTFLG(NODE),JDIR+11)) THEN                     SOL03070
           II=IDOF(NODE,JDIR)-NCOND                                     SOL03080
           IF (II.GE.1.AND.II.LE.NFREE) PLU(II,K)=COSINE(JDIR,K,ICOS)   SOL03090
         ENDIF                                                          SOL03100
  650   CONTINUE                                                        SOL03110
  554  CONTINUE                                                         SOL03120
  451 CONTINUE                                                          SOL03130
C                                                                       SOL03140
      IF (PRIN5) CALL WMATRX(PLU,NFREE,3,'INFLUENCE COEF. MATRIX')      SOL03150
   92 FORMAT(1X,'DIRECTION OF COSINE:',4X,SS,F9.5,                      SOL03160
     &         ' I',SP,F9.5,' J',F9.5,' K  ',SS,/)                      SOL03170
   93 FORMAT (1X,1P,10G12.4)                                            SOL03180
   94 FORMAT (/1X,'INPUT RESPONSE SPECTRUM VALUES.',/)                  SOL03190
  929 FORMAT(1X,'DIRECTION OF DISPLACEMENT.',4X,SS,F9.5,                SOL03200
     &         ' I',SP,F9.5,' J',F9.5,' K  ',SS)                        SOL03210
```

```
C*--------------------------------------------------------------
C** CALCULATE CROSS-MODAL COEFFICIENTS FOR CQC APPROACH --
C*--------------------------------------------------------------
      IF ( ICOMN .EQ. 2. ) THEN
        DO 809 I=1,NMODE
          CMC(I,I)= 1.
        DO 809 J=I+1,NMODE
          WJI=EVAL(J)/EVAL(I)
          COEF1=8*(DAMP**2)*(1+WJI)*(WJI**1.5)
          COEF2=(1-WJI**2)**2 + 4*(DAMP**2)*WJI*(1+WJI)**2
          CMC(I,J)=COEF1/COEF2
          CMC(J,I)=CMC(I,J)
  809   CONTINUE
C
       WRITE(108,10) TITLE(1),TITLE(2)
       CALL WMATRX(CMC,NMODE,NMODE,'CROSS-MODAL COF  MATRIX')
       ENDIF
C*-----------------------------------------------
C** CALCULATE GENERALIZED MASS, GM=EVEC'*B*EVEC --
C*-----------------------------------------------
      DO 450 I=1,NMODE
        GM(I)=0.
       DO 460 II=1,NFREE
        DISP(II)=0
        DO 470 JJ=1,NFREE
  470    DISP(II)=DISP(II) + B(II,JJ)*EVEC(JJ,I)
        GM(I)=GM(I)+EVEC(II,I)*DISP(II)
  460  CONTINUE
  450 CONTINUE
      IF (PRIN5) CALL WMATRX(GM,NMODE,1,'GENERIZED MASS MATRIX')
C*--------------------------------------------------------------
C** CALCULATE EFFECTIVE-LOAD FACTOR FOR EACH MODE, ELF=EVEC'*B*PLU --
C*--------------------------------------------------------------
      DO 455 I=1,NMODE
       DO 457 K=1,3
        ELF(I,K)=0.
        DO 465 II=1,NFREE
         DISP(II)=0
         DO 475 JJ=1,NFREE
  475     DISP(II)=DISP(II) + B(II,JJ)*PLU(JJ,K)
         ELF(I,K)=ELF(I,K)+EVEC(II,I)*DISP(II)
  465   CONTINUE
  457  CONTINUE
  455 CONTINUE
      IF (PRIN5) CALL WMATRX(ELF,NMODE,3,'EFFECTIVE-LOAD MATRIX')
C*--------------------------------------------------------------
C** CALCULATE EFFECTIVE MASS FOR EACH MODE IN EACH INDIVIDUAL --
C   SEISMIC INPUT DIRECTION
C*--------------------------------------------------------------
C
C.....READ EFFECTIVE MASS RATIO LIMIT IN X, Y, Z DIRECION
C     NOTE: IF NO  OF MODE BASED ON THE LIMIT IS LESS THAN NMODE,
C           NO. OF MODES BASED ON THIS LIMIT IS USED.
C
      READ(105,*) XLIM,YLIM,ZLIM
C
      WRITE(108,10) TITLE(1),TITLE(2)
      WRITE(108,487)
  487 FORMAT(1X,'EFFECTIVE MASS IN GCS" TRANSLATIONAL DIRECTIONS',/
     &       1X,'-----------------------------------------------',/)
C
      WRITE(108,987) 'X',XLIM
      WRITE(108,987) 'Y',YLIM
      WRITE(108,987) 'Z',ZLIM
C
C ...CALCULATE TOTAL STRUCTURAL TRANSLATIONAL MASS
      TOTX=0.
      TOTY=0.
      TOTZ=0.
      DO 705 J=L3,L4
        IJ=MDMS(J)
        DO 706 I=1,NMODE
            IF ( .NOT. BTEST(JTFLG(I),12) ) THEN
              NDOFX=IDOF(I,1)
              IF (J .EQ. NDOFX) TOTX=TOTX + MASS(IJ)
            ENDIF
            IF ( .NOT. BTEST(JTFLG(I),13) ) THEN
              NDOFY=IDOF(I,2)
              IF (J .EQ. NDOFY) TOTY=TOTY + MASS(IJ)
            ENDIF
            IF ( .NOT. BTEST(JTFLG(I),14) ) THEN
              NDOFZ=IDOF(I,3)
              IF (J .EQ  NDOFZ) TOTZ=TOTZ + MASS(IJ)
            ENDIF
  706   CONTINUE
  705 CONTINUE
      WRITE(108,388) 'X',TOTX
      WRITE(108,388) 'Y',TOTY
      WRITE(108,388) 'Z',TOTZ
  388 FORMAT(1X,'NOTE. TOTAL STRUCTURAL MASS IN ',A,' DIRECTION IS ',
     &       0P,G15.6)
      WRITE(108,386)
  386 FORMAT(1X,'   MODE      ','GCS X-DIRECTION(%)    ',
     &          '           ','GCS Y-DIRECTION(%)    ',
     &          '           ','GCS Z-DIRECTION(%)    ')
  987 FORMAT(1X,'EFFECTIVE MASS LIMIT IN GCS" ',A,
     &       ' DIRECTION IS ',G15.6 )
C
      TXLIM=0.
      TYLIM=0
      TZLIM=0
      MODEX=0
      MODEY=0
      MODEZ=0
      FXME=0
      FYME=0.
      FZME=0
      DO 701 I=1,NMODE
        IF (TOTX .EQ. 0.) GO TO 707
          FXME=(ELF(I,1)**2/GM(I))/TOTX
          TXLIM=TXLIM + FXME
          IF (XLIM.NE.0 0.AND.MODEX.EQ.0.AND.TXLIM.GE.XLIM) MODEX=I
  707   IF (TOTY .EQ. 0.) GO TO 708
          FYME=(ELF(I,2)**2/GM(I))/TOTY
          TYLIM=TYLIM + FYME
          IF (YLIM NE.0.0.AND.MODEY.EQ.0.AND.TYLIM.GE.YLIM) MODEY=I
  708   IF (TOTZ .EQ. 0.) GO TO 709
          FZME=(ELF(I,3)**2/GM(I))/TOTZ
          TZLIM=TZLIM + FZME
          IF (ZLIM NE.0.0.AND.MODEZ.EQ.0.AND.TZLIM.GE.ZLIM) MODEZ=I
  709   WRITE(108,389) I,FXME,FYME,FZME
  389   FORMAT(5X,I3,5X,3(G15 4,6X))
  701 CONTINUE
C
      IMO=MAX(MODEX,MODEY,MODEZ)
      IF ( IMO .NE  0 .AND. IMO .LT. NMODE) THEN
       MMODE=IMO
       WRITE(108,*) ' ** BASED ON EFFECTIVE MASS LIMIT, FINAL NO  OF '
     &              'MODES USED IS ',MMODE,' **'
      ELSE
       MMODE=NMODE
      ENDIF
C*-----------------------------------------
C** CALCULATE MAXIMUM DISPLACEMENT RESPONSES --
C*-----------------------------------------
C
      DO 560 NN=1,NA
       DO 540 I=1,L6
  540  TMP(I)=0.
       DO 541 I=1,NFREE
  541  TMP2(I)=0
       DO 550 J=1,MMODE
        DO 500 II=1,NFREE
         DLOAD(II,1)=EVEC(II,J)*ELF(J,NN)*RD(J,NN)/GM(J)
         IF (ICOMN  EQ. 2.) THEN
          CQCM(II-NCOND,J)=DLOAD(II,1)
          ELSE IF (ICOMN .EQ  1 ) THEN
           TMP(II+NCOND)=DLOAD(II,1)**2 + TMP(II+NCOND)
          ENDIF
  500   CONTINUE
C
        IF (PRIN5) THEN
          WRITE(108,*) 'MODAL DISPLACEMENTS FOR MODE ',J
          WRITE(108,*) ('DISP(',I,')= ',DLOAD(II,1),I=1,NFREE)
        ENDIF
C
        DO 501 LL=L1,L2
  501    DISP(LL)=0.
        DO 502 LL=L3,L4
  502   DISP(LL)=DLOAD(LL-L2,1)
C
C*---------------------------------------
C** CLACULATE EQUIVALENT EARTHQUAKE FORCES --
C*---------------------------------------
        DO 753 I=L5,L6
         DISP(I)=0.
  753   LOAD(I)=0.
C
        DO 580 I=1,NFREE
         LOAD(I+NCOND)=0
         DO 735 JJ=1,NFREE
  735    LOAD(I+NCOND)=LOAD(I+NCOND)+A(I,JJ)*DISP(JJ+NCOND)
         IF (ICOMN .EQ  1 ) THEN
          TMP2(I)=TMP2(I)+LOAD(I+NCOND)**2
         ELSE IF (ICOMN .EQ. 2.) THEN
          CQCM2(I,J)=LOAD(I+NCOND)
         ENDIF
  580   CONTINUE
C
        IF (PRIN5) THEN
         WRITE(108,10) TITLE(1),TITLE(2)
         WRITE(108,605) J,NN
  605    FORMAT(1X,'TOTAL EQUIVALENT EARTHQUAKE FORCES FOR MODE ',I3,
     &          ' IN THE SEISMIC COMPONENT # ',I2/)
         WRITE(108,911)
         WRITE(108,212) (I+NCOND,LOAD(I+NCOND),I=1,NFREE)
        ENDIF
C
C*--------------------------------------
C** CALC RESPONSE FOR CONDENSED OUT DOF --
C*--------------------------------------
        CALL REACTN(1,BTEST(IBUG,5),NNODE,L1,L2,L3,L4,DISP,DISP,MD,
     &              STIFF,IMD,NDOF,1,1.0,MAXNOD,IDOF,COORD,ID,TITLE,
     &              STEPID, FALSE.,NCOS,COSINE,JTCOS,JTPLG,SUMRCT)
        CALL MSOLL(3,BTEST(IBUG,6),L4,IMD,NDOF,1,L1,L2,MD,STIFF,DISP,
     &             WORK,.FALSE.,MASS,MDMS,1, FALSE.,KG,MDKG,1)
C
        DO 675 I=L1,L2
         IF (ICOMN .EQ. 1 ) THEN
          TMP(I)=DISP(I)**2 + TMP(I)
         ELSE IF (ICOMN  EQ. 2.) THEN
          CQCM(I,J)=DISP(I)
         ENDIF
  675   CONTINUE
C
C*--------------------------------------
C- CALCULATE MEMBER FORCES ---
C*--------------------------------------
        LSTTYP=0
        DO 358 IELNO=1,NELMT
         DO 357 I=1,12
  357    RSAF(I)=0.
         CALL ELELIB
     &       (4,LSTTYP,PRIN5,IREL,.FALSE.,NAME,EPSE,EPSE,DAMAGE,DUCFAG,
     &        IELNO,IELDOF,KGDOF,PGECM,RINPUT,AXIAL,IZDISP,IZLOAD,MSTOR)
         DO 359 I=1,12
  359    ELEF(IELNO,I,J)=RSAF(I)
  358   CONTINUE
C
C*-------------------------
C- SOLVE FOR REACTIONS ---
C*-------------------------
C
        CALL REACTN(2,PRIN5,NNODE,L1,L4,L5,L6,LOAD,DISP,MD,
     &              STIFF,IMD,NDOF,1,1.0,MAXNOD,IDOF,COORD,ID,TITLE,
     &              STEPID,.FALSE.,NCOS,COSINE,JTCOS,JTPLG,SUMRCT)
C
C*-----------------------------
C- SOLVE FOR GLOBAL REACTIONS ---
C*-----------------------------
C
        DO 366 I=1,6
  366   SUMRCT(I)=0.
        CALL REACTN(3,PRIN5,NNODE,L1,L4,L5,L6,LOAD,DISP,MD,
     &              STIFF,IMD,NDOF,1,1.0,MAXNOD,IDOF,COORD,ID,TITLE,
     &              STEPID,.FALSE.,NCOS,COSINE,JTCOS,JTPLG,SUMRCT)
        DO 360 I=1,6
  360   RCT(I,J)=SUMRCT(I)
C
        IF (PRIN5) THEN
         WRITE(108,10) TITLE(1),TITLE(2)
         WRITE(108,607) J,NN
  607    FORMAT(1X,'REACTION FOR MODE ',I3,' IN SEISMIC COMP # ',I2/)
         WRITE(108,323) SUMRCT
        ENDIF
C
  550   CONTINUE
C
      IF (ICOMN .EQ. 1.) THEN
        DO 590 JJ=1,L4
  590    TMP(JJ)=SQRT(TMP(JJ))
        DO 547 JJ=1,NFREE
  547    TMP2(JJ)=SQRT(TMP2(JJ))
        DO 591 I=1,NELMT
        DO 591 II=1,12
        DO 592 J=1,MMODE
  592    ELEF(I,II,MMODE+NN)=ELEF(I,II,MMODE+NN)*ELEF(I,II,J)**2
         ELEF(I,II,MMODE+NN)=SQRT(ELEF(I,II,MMODE+NN) )
  591   CONTINUE
        DO 593 I=1,6
        DO 594 J=1,MMODE
  594    RCT(I,MMODE+NN)=RCT(I,MMODE+NN)*RCT(I,J)**2
         RCT(I,MMODE+NN)=SQRT(RCT(I,MMODE+NN) )
  593   CONTINUE
C
      ELSE IF(ICOMN .EQ. 2.) THEN
          DO 826 L=1,L4
            TMP(L)=0.
            DO 828 J=1,MMODE
            DO 828 JJ=1,MMODE
  828         TMP(L)=TMP(L)+CQCM(L,J)*CMC(J,JJ)*CQCM(L,JJ)
            TMP(L)=SQRT(ABS(TMP(L)))
  826     CONTINUE
C
          DO 837 L=1,NFREE
            DO 838 J=1,MMODE
            DO 838 JJ=1,MMODE
  838         TMP2(L)=TMP2(L)+CQCM2(L,J)*CMC(J,JJ)*CQCM2(L,JJ)
            TMP2(L)=SQRT(ABS(TMP2(L)))
  837     CONTINUE
C
          DO 595 I=1,NELMT
          DO 595 II=1,12
            SUM=0.
          DO 596 J=1,MMODE
          DO 596 JJ=1,MMODE
  596       SUM=SUM+ELEF(I,II,J)*CMC(J,JJ)*ELEF(I,II,JJ)
          SUM=ABS(SUM)
          ELEF(I,II,MMODE+NN)=SQRT(SUM)
  595     CONTINUE
          DO 597 I=1,6
            SUM=0
```

```
      DO 598 J=1,MMODE                                                  SOL06140
      DO 598 JJ=1,MMODE                                                 SOL06150
  598   SUM=SUM+RCT(I,J)*CMC(J,JJ)*RCT(I,JJ)                           SOL06160
        SUM=ABS(SUM)                                                    SOL06170
        RCT(I,MMODE+NN)=SQRT(SUM)                                       SOL06180
  597   CONTINUE                                                        SOL06190
C                                                                       SOL06200
      ENDIF                                                             SOL06210
C                                                                       SOL06220
      IF ( NA  GT  1 ) THEN                                             SOL06230
       WRITE(108,10) TITLE(1),TITLE(2)                                  SOL06240
       WRITE(108,489) NN                                                SOL06250
  489  FORMAT(1X,'MAXIMUM STRUCTURAL RESPONSES ',                       SOL06260
     &          'DUE TO SEISMIC COMPONENT # ',I2,/                      SOL06270
     &       1X,'------------------------------',                       SOL06280
     &          '------------------------------',/)                    SOL06290
       WRITE(108,913)                                                   SOL06300
  913  FORMAT(3(' : D O.F.    DISPL     :'),/)                          SOL06310
       WRITE(108,212) (I+NCOND,TMP(I+NCOND),I=1,NFREE)                  SOL06320
C                                                                       SOL06330
C-- CALCULATE TOTAL ELEMENT FORCES                                      SOL06340
       WRITE(108,10) TITLE(1),TITLE(2)                                  SOL06350
       WRITE(108,984) NN                                                SOL06360
  984  FORMAT(1X,'MAXIMUM ELEMENT FORCES DUE TO SEISMIC COMPONENT # ',  SOL06370
     &       I2,/1X,'--------------------------------------------',     SOL06380
     &       '----------',/)                                           SOL06390
       PRIN0= TRUE.                                                     SOL06400
       LSTTYP=0                                                         SOL06410
       DO 905 I=1,NELMT                                                 SOL06420
         IELNO=I                                                        SOL06430
         DO 959 II=1,12                                                 SOL06440
  959    RSAP(II)=ELEP(I,II,MMODE+NN)                                   SOL06450
         CALL ELELIB                                                    SOL06460
     &       (5,LSTTYP,PRIN0,IREL,.FALSE.,NAME,EESE,EPSE,DAMAGE,DUCFAG, SOL06470
     &       IELNO,IELDOF,KGDOF,PGECM,RINPUT,AXIAL,IZDISP,IZLOAD,MSTOR) SOL06480
  905 CONTINUE                                                          SOL06490
C                                                                       SOL06500
C----CALCULATE TOTAL EQUIVALENT EARTHQUAKE FORCES                       SOL06510
       WRITE(108,10) TITLE(1),TITLE(2)                                  SOL06520
       WRITE(108,184) NN                                                SOL06530
  184  FORMAT(1X,'MAXIMUM EQUIVALENT EARTHQUAKE FORCES ',               SOL06540
     &       'DUE TO SEISMIC COMPONENT # ',I2,/                         SOL06550
     &       1X,'----------------------------------------',             SOL06560
     &       '----------',/)                                           SOL06570
       WRITE(108,912)                                                   SOL06580
       WRITE(108,212) (I-NCOND,TMP2(I),I=1,NFREE)                       SOL06590
C                                                                       SOL06600
C----CALCULATE TOTAL REACTION AT GCS" ORIGIN                            SOL06610
       DO 696 I=1,6                                                     SOL06620
  696    SUMRCT(I)=RCT(I,MMODE+NN)                                      SOL06630
       WRITE(108,10) TITLE(1),TITLE(2)                                  SOL06640
       WRITE(108,232) NN                                                SOL06650
  232  FORMAT(//' TOTAL REACTIONS AT GCS" ORIGIN ',                     SOL06660
     &          'DUE TO SEISMIC COMPONENT # ',I2,/                      SOL06670
     &          ' ------------------------------ ',                    SOL06680
     &          '------------------------------',/                     SOL06690
     &  1X,'FX',13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ',/)         SOL06700
       WRITE(108,323) SUMRCT                                            SOL06710
      ENDIF                                                             SOL06720
C                                                                       SOL06730
C------ STORE RESPONSE TO DISP ARRAY                                    SOL06740
         DO 200 J=L1,L4                                                 SOL06750
  200     DLOAD(J,NN)=TMP(J)                                           SOL06760
         DO 220 J=L5,L6                                                 SOL06770
  220     DLOAD(J,NN)=0.                                               SOL06780
C                                                                       SOL06790
C------- STORE EQUIVALENT EARTHQUAKE FORCES IN EEP ARRAY                SOL06800
         DO 291 J=1,NFREE                                               SOL06810
  291     EEP(J,NN)=TMP2(J)                                            SOL06820
C                                                                       SOL06830
  560 CONTINUE                                                          SOL06840
C                                                                       SOL06850
C-----CALCULATE TOTAL RESPONSES                                         SOL06860
      DO 237 J=L1,L6                                                    SOL06870
        DISP(J)=0.                                                      SOL06880
      DO 237 I=1,NA                                                     SOL06890
  237   DISP(J)=DISP(J)+DLOAD(J,I)                                      SOL06900
      WRITE(108,10) TITLE(1),TITLE(2)                                   SOL06910
      WRITE(108,608)                                                    SOL06920
  608 FORMAT(1X,'TOTAL MAXIMUM STRUCTURAL RESPONSES',/                  SOL06930
     &       1X,'--------------------------------',/)                   SOL06940
       WRITE(108,913)                                                   SOL06950
       WRITE(108,212) (I+NCOND,DISP(I+NCOND),I=1,NFREE)                 SOL06960
C                                                                       SOL06970
C-----CALCULATE TOTAL EQUIVALENT EARTHQUAKE FORCE                       SOL06980
      DO 202 J=1,NFREE                                                  SOL06990
        TMP2(J)=0.                                                      SOL07000
      DO 202 I=1,NA                                                     SOL07010
  202   TMP2(J)=TMP2(J)+EEP(J,I)                                        SOL07020
      WRITE(108,10) TITLE(1),TITLE(2)                                   SOL07030
      WRITE(108,678)                                                    SOL07040
  678 FORMAT(1X,'TOTAL MAXIMUM EQUIVALENT EARTHQUAKE FORCES',/          SOL07050
     &       1X,'-----------------------------------------',/)         SOL07060
       WRITE(108,912)                                                   SOL07070
       WRITE(108,212) (I+NCOND,TMP2(I),I=1,NFREE)                       SOL07080
C                                                                       SOL07090
C-------------------------------                                        SOL07100
C- PRINT GLOBAL DISPLACEMENTS --                                        SOL07110
C-------------------------------                                        SOL07120
C                                                                       SOL07130
C----CALCULATE TOTAL ELEMENT FORCES                                     SOL07140
       WRITE(108,10)TITLE(1),TITLE(2)                                   SOL07150
       WRITE(108,980)                                                   SOL07160
  980  FORMAT(1X,'TOTAL MAXIMUM ELEMENT FORCES:',/,                     SOL07170
     &       1X,'----------------------------',/)                       SOL07180
      PRIN0= TRUE.                                                      SOL07190
      LSTTYP=0                                                          SOL07200
      DO 509 I=1,NELMT                                                  SOL07210
        IELNO=I                                                         SOL07220
        DO 599 II=1,12                                                  SOL07230
          RSAP(II)=0.                                                   SOL07240
        DO 599 J=1,NA                                                   SOL07250
  599     RSAP(II)=RSAP(II)+ELEP(I,II,MMODE+J)                          SOL07260
        CALL ELELIB                                                     SOL07270
     &       (5,LSTTYP,PRIN0,IREL,.FALSE.,NAME,EESE,EPSE,DAMAGE,DUCFAG, SOL07280
     &       IELNO,IELDOF,KGDOF,PGECM,RINPUT,AXIAL,IZDISP,IZLOAD,MSTOR) SOL07290
  509 CONTINUE                                                          SOL07300
C                                                                       SOL07310
C----CALCULATE TOTAL REACTION AT GCS" ORIGIN                            SOL07320
      DO 699 I=1,6                                                      SOL07330
        SUMRCT(I)=0.                                                    SOL07340
      DO 699 J=1,NA                                                     SOL07350
        SUMRCT(I)=SUMRCT(I)+RCT(I,MMODE+J)                              SOL07360
  699 CONTINUE                                                          SOL07370
      WRITE(108,10) TITLE(1),TITLE(2)                                   SOL07380
      WRITE(108,322)                                                    SOL07390
  322 FORMAT(//' TOTAL REACTIONS AT GCS" ORIGIN ',/                     SOL07400
     &       ' ------------------------------ ',/17X,'FX',             SOL07410
     &       13X,'FY',13X,'FZ',13X,'MX',13X,'MY',13X,'MZ',/)           SOL07420
      WRITE(108,323) SUMRCT                                             SOL07430
  323 FORMAT(' GCS SUMM.',6G15.7)                                       SOL07440
C                                                                       SOL07450
C-------------------------------------------------                      SOL07460
C-- CALCULATE MAXIMUM INERTIA EARTHQUAKE FORCES --                      SOL07470
C-------------------------------------------------                      SOL07480
C                                                                       SOL07490
      DO 207 NN=1,NA                                                    SOL07500
       DO 203 I=1,NFREE                                                 SOL07510
203     TMP(I)=0.                                                       SOL07520
        DO 204 J=1,MMODE                                                SOL07530
         DO 205 II=1,NFREE                                              SOL07540
          WORK(II)=0                                                    SOL07550
          DO 206 JJ=1,NFREE                                             SOL07560
  206     WORK(II)=WORK(II)+B(II,JJ)*EVEC(JJ,J)                         SOL07570
          DISP(II)=WORK(II)*ELP(J,NN)*RA(J,NN)/GM(J)                    SOL07580
  205   CONTINUE                                                        SOL07590
C                                                                       SOL07600
       DO 915 I=1,NFREE                                                 SOL07610
        IF (ICOMN  EQ  2.) THEN                                         SOL07620
         CQCM(I,J)=DISP(I)                                              SOL07630
        ELSE IF (ICOMN  EQ  1 ) THEN                                    SOL07640
         TMP(I)=TMP(I)+DISP(I)**2                                       SOL07650
        ENDIF                                                           SOL07660
  915  CONTINUE                                                         SOL07670
C                                                                       SOL07680
       IF (PRIN5) THEN                                                  SOL07690
        WRITE(108,*) 'MODAL FORCE FOR MODE ',J                          SOL07700
        WRITE(108,*) ('DISP(',I,')= ',DISP(I),I=1,NFREE)                SOL07710
       ENDIF                                                            SOL07720
  204  CONTINUE                                                         SOL07730
C                                                                       SOL07740
       DO 926 L=1,NFREE                                                 SOL07750
        IF (ICOMN  EQ  1 ) THEN                                         SOL07760
         TMP(L)=SQRT(TMP(L))                                            SOL07770
        ELSE IF(ICOMN  EQ. 2 ) THEN                                     SOL07780
         SUM=0.                                                         SOL07790
         DO 928 J=1,MMODE                                               SOL07800
         DO 928 JJ=1,MMODE                                              SOL07810
  928      SUM=SUM+CQCM(L,J)*CMC(J,JJ)*CQCM(L,JJ)                       SOL07820
         SUM=ABS(SUM)                                                   SOL07830
         TMP(L)=SQRT(SUM)                                               SOL07840
        ENDIF                                                           SOL07850
  926  CONTINUE                                                         SOL07860
C                                                                       SOL07870
       IF ( NA  GT  1 ) THEN                                            SOL07880
        WRITE(108,10) TITLE(1),TITLE(2)                                 SOL07890
        WRITE(108,488) NN                                               SOL07900
  488   FORMAT(1X,'MAXIMUM INERTIA FORCES ','DUE TO SEISMIC COMPONENT', SOL07910
     &       ' #',I3,/1X,'-----------------------------------------',   SOL07920
     &       '-------------',/)                                         SOL07930
        WRITE(108,912)                                                  SOL07940
  912   FORMAT(3(' : D O.F.    FORCE     :'),/)                         SOL07950
        WRITE(108,212) (I+NCOND,TMP(I),I=1,NFREE)                       SOL07960
  212   FORMAT(3(3X,I6,1X,G15.4))                                       SOL07970
       ENDIF                                                            SOL07980
C                                                                       SOL07990
C. ..STORE EARTHQUAKE FORCES TO LOAD ARRAY                              SOL08000
C                                                                       SOL08010
      DO 233 J=L1,L2                                                    SOL08020
  233   DLOAD(J,NN)=0                                                   SOL08030
      DO 234 J=L3,L4                                                    SOL08040
  234   DLOAD(J,NN)=TMP(J-L2)                                           SOL08050
      DO 235 J=L5,L6                                                    SOL08060
  235   DLOAD(J,NN)=0                                                   SOL08070
  207 CONTINUE                                                          SOL08080
C                                                                       SOL08090
C-----CALCULATE TOTAL FORCES                                            SOL08100
      DO 236 J=L1,L6                                                    SOL08110
       LOAD(J)=0.                                                       SOL08120
       DO 236 I=1,NA                                                    SOL08130
  236  LOAD(J)=LOAD(J)+DLOAD(J,I)                                       SOL08140
      WRITE(108,10) TITLE(1),TITLE(2)                                   SOL08150
      WRITE(108,686)                                                    SOL08160
  686 FORMAT(1X,'TOTAL MAXIMUM INERTIA FORCES',/,                       SOL08170
     &       1X,'-----------------------------',/)                      SOL08180
      WRITE(108,912)                                                    SOL08190
      WRITE(108,212) (I+NCOND,LOAD(I+NCOND),I=1,NFREE)                  SOL08200
C                                                                       SOL08210
      RETURN                                                            SOL08220
      END                                                               SOL08230
C                                                                       GVCS0010
C .... .. .......... ..... .. .......... .............                  GVCS0020
C                                                                       GVCS0030
      SUBROUTINE GVCSP(N,A,B,D,X,DL,C1,C2)                              GVCS0040
      DIMENSION A(N,N),B(N,N),D(N),X(N,N),DL(N),C1(N),C2(N)             GVCS0050
      CALL REDUC1(N,A,B,DL,FAIL)                                        GVCS0060
      DO 20 I=2,N                                                       GVCS0070
      DO 20 J=1,I-1                                                     GVCS0080
20    A(J,I)=A(I,J)                                                     GVCS0090
      CALL JACOBI(N,A,D,X,C1,C2)                                        GVCS0100
      CALL REBAC1(N,B,DL,X)                                             GVCS0110
      DO 200 I=1,N                                                      GVCS0120
       IP1=I+1                                                          GVCS0130
       K=I                                                              GVCS0140
       P=D(I)                                                           GVCS0150
       IF (N.GT.I) THEN                                                 GVCS0160
        DO 120 J=IP1,N                                                  GVCS0170
         IF (D(J).GE.P) GOTO 120                                        GVCS0180
         K=J                                                            GVCS0190
         P=D(J)                                                         GVCS0200
120     CONTINUE                                                        GVCS0210
       ENDIF                                                            GVCS0220
       IF (K.EQ.I) GOTO 200                                             GVCS0230
       D(K)=D(I)                                                        GVCS0240
       D(I)=P                                                           GVCS0250
       DO 140 J=1,N                                                     GVCS0260
        P=X(J,I)                                                        GVCS0270
        X(J,I)=X(J,K)                                                   GVCS0280
        X(J,K)=P                                                        GVCS0290
140    CONTINUE                                                         GVCS0300
200   CONTINUE                                                          GVCS0310
      RETURN                                                            GVCS0320
      END                                                               GVCS0330
C                                                                       REDU0010
C ---------------------------------------------------                  REDU0020
C                                                                       REDU0030
      SUBROUTINE REDUC1(N,A,B,DL,FAIL)                                  REDU0040
      DIMENSION A(N,N),B(N,N),DL(N)                                     REDU0050
      DO 110 I=1,N                                                      REDU0060
      DO 110 J=I,N                                                      REDU0070
       X=B(I,J)                                                         REDU0080
       II=I-1                                                           REDU0090
       IF (II) 120,120,130                                              REDU0100
130    DO 140 K=1,II                                                    REDU0110
140    X=X-B(I,K)*B(J,K)                                                REDU0120
120    IF (I-J) 150,160,150                                             REDU0130
160    FAIL=X                                                           REDU0140
       IF (X) 400,400,170                                               REDU0150
170    DL(I)=SQRT(X)                                                    REDU0160
       GOTO 110                                                         REDU0170
150    B(J,I)=X/DL(I)                                                   REDU0180
110   CONTINUE                                                          REDU0190
200   DO 210 I=1,N                                                      REDU0200
      DO 210 J=I,N                                                      REDU0210
       X=A(I,J)                                                         REDU0220
       II=I-1                                                           REDU0230
       IF (II) 220,220,230                                              REDU0240
230    DO 240 K=1,II                                                    REDU0250
240    X=X-B(I,K)*A(J,K)                                                REDU0260
220    A(J,I)=X/DL(I)                                                   REDU0270
210   CONTINUE                                                          REDU0280
300   DO 310 J=1,N                                                      REDU0290
      DO 310 I=J,N                                                      REDU0300
       X=A(I,J)                                                         REDU0310
       II=I-1                                                           REDU0320
       IF (II-J) 320,330,330                                            REDU0330
330    DO 340 K=J,II                                                    REDU0340
340    X=X-A(K,J)*B(I,K)                                                REDU0350
320    JJ=J-1                                                           REDU0360
       IF (JJ) 350,350,360                                              REDU0370
360    DO 370 K=1,JJ                                                    REDU0380
370    X=X-A(J,K)*B(I,K)                                                REDU0390
350    A(I,J)=X/DL(I)                                                   REDU0400
310   CONTINUE                                                          REDU0410
400   RETURN                                                            REDU0420
      END                                                               REDU0430
C                                                                       REBA0010
C ---------- ---------- ----------                                      REBA0020
C                                                                       REBA0030
      SUBROUTINE REBAC1(N,B,DL,Z)                                       REBA0040
      DIMENSION B(N,N),DL(N),Z(N,N)                                     REBA0050
      DO 100 J=1,N                                                      REBA0060
```

```
          P=1.E-15                                                      REBA0070
          DO 60 I=N,1,-1                                                REBA0080
            X=Z(I,J)                                                    REBA0090
            I1=I+1                                                      REBA0100
            IF ((I1-N).GT.0) GOTO 50                                    REBA0110
            DO 40 K=I1,N                                                REBA0120
40          X=X-B(K,I)*Z(K,J)                                           REBA0130
50          Z(I,J)=X/DL(I)                                              REBA0140
            IF (ABS(Z(I,J)).GT.P) P=ABS(Z(I,J))                         REBA0150
60        CONTINUE                                                      REBA0160
          DO 80 I=1,N                                                   REBA0170
80        Z(I,J)=Z(I,J)/P                                               REBA0180
100     CONTINUE                                                        REBA0190
        RETURN                                                          REBA0200
        END                                                             REBA0210
C                                                                       JACO0010
C  ------------------------------------------ -                        JACO0020
C                                                                       JACO0030
        SUBROUTINE JACOBI(N,A,D,V,B,Z)                                  JACO0040
        DIMENSION A(N,N),D(N),V(N,N),B(N),Z(N)                          JACO0050
C  ------------------------------------------                           JACO0060
20      DO 21 I=1,N                                                     JACO0070
        DO 22 J=1,N                                                     JACO0080
22      V(I,J)=0.0                                                      JACO0090
21      V(I,I)=1.                                                       JACO0100
10      DO 30 K=1,N                                                     JACO0110
        D(K)=A(K,K)                                                     JACO0120
        B(K)=D(K)                                                       JACO0130
        Z(K)=0.0                                                        JACO0140
30      CONTINUE                                                        JACO0150
        DO 40 I=1,500                                                   JACO0160
        SM=0.0                                                          JACO0170
        N1=N-1                                                          JACO0180
        DO 50 IP=1,N1                                                   JACO0190
        IP1=IP+1                                                        JACO0200
        DO 50 IQ=IP1,N                                                  JACO0210
50      SM=SM+ABS(A(IP,IQ))                                             JACO0220
        IF (SM) 60,70,60                                                JACO0230
60      IF (I-4) 80,90,90                                               JACO0240
80      TRESH=0.2*SM/FLOAT(N)**2                                        JACO0250
        GOTO 100                                                        JACO0260
90      TRESH=0.0                                                       JACO0270
100     DO 110 IP=1,N1                                                  JACO0280
        IP1=IP+1                                                        JACO0290
        DO 110 IQ=IP1,N                                                 JACO0300
        G=100.0*ABS(A(IP,IQ))                                           JACO0310
        IF (G.LE 1.E-20) G=0.0                                          JACO0320
        IF (I-4) 120,120,130                                            JACO0330
130     IF (ABS(D(IP))+G-ABS(D(IP))) 120,140,120                        JACO0340
140     IF (ABS(D(IQ))+G-ABS(D(IQ))) 120,150,120                        JACO0350
150     A(IP,IQ)=0.0                                                    JACO0360
        GOTO 110                                                        JACO0370
120     IF (ABS(A(IP,IQ))-TRESH) 110,110,160                            JACO0380
160     H=D(IQ)-D(IP)                                                   JACO0390
        IF (ABS(H)+G-ABS(H)) 170,180,170                                JACO0400
180     T=A(IP,IQ)/H                                                    JACO0410
        GOTO 190                                                        JACO0420
170     THETA=0.5*H/A(IP,IQ)                                            JACO0430
        T=SIGN(1.0,THETA)/(ABS(THETA)+SQRT(1.+THETA*THETA))             JACO0440
190     C=1./SQRT(1.0+T*T)                                              JACO0450
        S=T*C                                                           JACO0460
        TAU=S/(1.+C)                                                    JACO0470
        H=T*A(IP,IQ)                                                    JACO0480
        Z(IP)=Z(IP)-H                                                   JACO0490
        Z(IQ)=Z(IQ)+H                                                   JACO0500
        D(IP)=D(IP)-H                                                   JACO0510
        D(IQ)=D(IQ)+H                                                   JACO0520
        A(IP,IQ)=0.0                                                    JACO0530
200     IP2=IP+1                                                        JACO0540
        IF (IP2) 210,210,201                                            JACO0550
201     DO 205 K=1,IP2                                                  JACO0560
        G=A(K,IP)                                                       JACO0570
        H=A(K,IQ)                                                       JACO0580
        A(K,IP)=G-S*(H+G*TAU)                                           JACO0590
        A(K,IQ)=H+S*(G-H*TAU)                                           JACO0600
205     CONTINUE                                                        JACO0610
210     IQ1=IQ+1                                                        JACO0620
        IF (IP1-IQ1) 211,211,220                                        JACO0630
211     DO 215 K=IP1,IQ1                                                JACO0640
        G=A(IP,K)                                                       JACO0650
        H=A(K,IQ)                                                       JACO0660
        A(IP,K)=G-S*(H+G*TAU)                                           JACO0670
        A(K,IQ)=H+S*(G-H*TAU)                                           JACO0680
215     CONTINUE                                                        JACO0690
220     IQ1=IQ+1                                                        JACO0700
        IF (IQ1-N) 221,221,230                                          JACO0710
221     DO 225 K=IQ1,N                                                  JACO0720
        G=A(IP,K)                                                       JACO0730
        H=A(IQ,K)                                                       JACO0740
        A(IP,K)=G-S*(H+G*TAU)                                           JACO0750
        A(IQ,K)=H+S*(G-H*TAU)                                           JACO0760
225     CONTINUE                                                        JACO0770
230     DO 235 K=1,N                                                    JACO0780
        G=V(K,IP)                                                       JACO0790
        H=V(K,IQ)                                                       JACO0800
        V(K,IP)=G-S*(H+G*TAU)                                           JACO0810
        V(K,IQ)=H+S*(G-H*TAU)                                           JACO0820
235     CONTINUE                                                        JACO0830
110     CONTINUE                                                        JACO0840
        DO 240 K=1,N                                                    JACO0850
        B(K)=B(K)+Z(K)                                                  JACO0860
        D(K)=B(K)                                                       JACO0870
        Z(K)=0.0                                                        JACO0880
240     CONTINUE                                                        JACO0890
40      CONTINUE                                                        JACO0900
70      RETURN                                                          JACO0910
        END                                                             JACO0920
C------------ BEGIN COMMON ------------ --- ------------------------- ZCOM0010
$INCLUDE:'ZCOMN1'                                                       ZCOM0020
$INCLUDE:'ZCOMN2'                                                       ZCOM0030
$INCLUDE:'ZCOMN3'                                                       ZCOM0040
C------------ END COMMON ----- ------------ ------------------------- ZCOM0050
C------------ BEGIN BLOCK - ZCOMN1 - ------------------------------C ZCOM0060
      CHARACTER*112 TITLE(2)                                            ZCOM0070
      LOGICAL BUG,MCOND,KGCOND,DOUBLE,NEWKG                            ZCOM0080
      INTEGER FMASS,FDAMP,CPUREM,CPUELA                                 ZCOM0090
      COMMON /ZDATAP/ TITLE ,BUG    ,MAXEDF,MD(25),GRAV                 ZCOM0100
      COMMON /ZDATA/  IZ    ,IZREL  ,CPUREM,CPUELA,ITLCPU,INCCPU,       ZCOM0110
     &  KSAME ,NEWKG ,MCOND ,KGCOND,DOUBLE,NCOND ,NFREE ,               ZCOM0120
     &  NREST ,NMAT  ,NELMT ,NMASS ,NSOLN ,IZID  ,IZIDOF,IZCORD,        ZCOM0130
     &  IZCOS ,IZCNST,IZJFLG,IZJCOS,IZMAT ,IZELE ,IZIELD,               ZCOM0140
     &  IZELD ,IZKE  ,IZLM  ,IZSTIP,IZMD  ,IZKEKG,IZLMKG,IZMASS,        ZCOM0150
     &  IZMDMS,IZMDAT,IZDAMP,IZXGDT,IZXG  ,IZMDKG,IZLOAD,IZDISP,        ZCOM0160
     &  IZVEL ,IZACC ,IZDLOA,IZDDSP,IZDVEL,IZDACC,IZAGTX,IZAGTY,        ZCOM0170
     &  IZAGT2,IZDUMP,IZFUB ,IZAN  ,IZAR  ,FMASS ,FDAMP ,IUNIT ,       ZCOM0180
     &  BSE   ,PSE   ,EKE   ,EDD   ,EDUMMY,SUMRCT(6),GROUND(9),        ZCOM0190
     &  IDUMMY(500)                                                     ZCOM0200
C                                                                     C ZCOM0210
C------------ BEGIN BLOCK - ZCOMN2 ---------- ------------------------C ZCOM0220
      PARAMETER (MAXZ=10000,MAXD2=MAXZ/2)                               ZCOM0230
      COMMON /DATA/ Z(0:MAXZ)                                           ZCOM0240
      DIMENSION NZ(0:MAXZ),DZ(0:MAXZ)                                   ZCOM0250
      EQUIVALENCE (Z(0),NZ(0),DZ(0))                                    ZCOM0260
C                                                                     C ZCOM0270
C------------ END   BLOCK - ZCOMN2 ---------- ------------------------C ZCOM0280
C------------ BEGIN BLOCK - ZCOMN3 ---------- ------------------------ ZCOM0290
      CHARACTER*40  STEPID                                              ZCOM0300
      LOGICAL NEWK,ELSTIC                                               ZCOM0310
        COMMON /ZDATAG/INC      ,IBUG       ,MAXELD     ,NELD       , ZCOM0320
     &  NDOF      ,NLOAD      ,MAXNOD     ,NNODE      ,NCOS       ,    ZCOM0330
     &  IUNIT     ,STEPID     ,NEWK       ,ELSTIC                      ZCOM0340
C------ ----- END   COMMON BLOCK - ZCOMN ---------- ------------------- ZCOM0350
C------------ BEGIN BLOCK - ZCOMN4 ---- ------------------------------ ZCOM0360
      CHARACTER*112 TITLE(2)                                            ZCOM0370
      CHARACTER*40  STEPID                                              ZCOM0380
      LOGICAL BUG,MCOND,KGCOND,DOUBLE,NEWK,NEWKG,ELSTIC,ABORT          ZCOM0390
      INTEGER FMASS,FDAMP,CPUREM,CPUELA                                 ZCOM0400
      DOUBLE PRECISION BIE,BSE,PSE,EKE,EDD,EDUMMY                       ZCOM0410
      COMMON /ZDATAP/ TITLE ,BUG    ,IBUG   ,MAXEDF,MD(25),STEPID,GRAV  ZCOM0420
      COMMON /ZDATA/  IZ    ,IZREL  ,CPUREM,CPUELA,ITLCPU,INCCPU,       ZCOM0430
     &  NEWK  ,KSAME ,NEWKG ,MCOND ,KGCOND,DOUBLE,ELSTIC,ABORT ,        ZCOM0440
     &  NNODE ,NDOF  ,NCOND ,NFREE ,NREST ,MAXNOD,NCOS  ,NMAT  ,        ZCOM0450
     &  NELMT ,NMASS ,NLOAD ,NSOLN ,MAXELD,NELD  ,IZID  ,IZIDOF,        ZCOM0460
     &  IZCORD,IZCOS ,IZCNST,IZJFLG,IZJCOS,IZMAT ,IZELE ,IZIELD,        ZCOM0470
     &  IZELD ,IZKE  ,IZLM  ,IZSTIP,IZMD  ,IZKEKG,IZLMKG,IZMASS,        ZCOM0480
     &  IZMDMS,IZMDAT,IZDAMP,IZXGDT,IZXG  ,IZMDKG,IZLOAD,IZDISP,        ZCOM0490
     &  IZVEL ,IZACC ,IZDLOA,IZDDSP,IZDVEL,IZDACC,IZAGTX,IZAGTY,        ZCOM0500
     &  IZAGT2,IZDUMP,IZFUB ,IZAN  ,IZAR  ,FMASS ,FDAMP ,IUNIT ,       ZCOM0510
     &  BIE   ,BSE   ,PSE   ,EKE   ,EDD   ,EDUMMY,SUMRCT(6),GROUND(9), ZCOM0520
     &  INC   ,IDUMMY(500)                                              ZCOM0530
C                                                                       ZCOM0540
C------------ END   BLOCK - ZCOMN4 ---------- --- ---------------------- ZCOM0550
```

REPORT DOCUMENTATION PAGE	1. REPORT NO. Structural Series 96-5	2.	3. Recipient's Accession No.
4. Title and Subtitle INRESB-3D-SUPII Program Listing for PC: General Purpose Program for Inelastic Analysis of RC and Steel Building Systems for 3D Static and Dynamic Loads and Seismic Excitations			5. Report Date August 1996 6.
7. Author(s) Franklin Y. Cheng, J.F. Ger, Dan Li and J.S. Yang			8. Performing Organization Rept. No. Structural Series
9. Performing Organization Name and Address Department of Civil Engineering University of Missouri-Rolla Rolla, MO 65409-0030			10. Project/Task/Work Unit No. 11. Contract(C) or Grant(G) No. (C) (G) NSF BCS 9001494 NSF MSS 9214664
12. Sponsoring Organization Name and Address National Science Foundation 4201 Wilson Blvd. Arlington, VA 22230			13. Type of Report & Period Covered Final 14.

15. Supplementary Notes

16. Abstract (Limit: 200 words) **INRESB-3D-SUPII (PC version) is written in FORTRAN and contains about 30,000 statements. While the PC and supercomputer versions have identical functions, the PC solution capacity is limited by its memory. INRESB-3D-SUPII is a general purpose computer program for analyzing elastic and inelastic building systems subject to static loads, dynamic forces, multicomponent earthquake motion, and pseudo-static cyclic loads. Also the program is capable of calculating inelastic post-buckling behavior of steel members and systems, natural frequency, and free vibration. A joint-based system is used to define the geometry of a structure. Structural members may be elastic 3D prismatic beams, nonlinear RC shear walls, nonlinear springs, inelastic 3D-beam-column elements, finite-segment elements, and non-linear bracing elements. INRESB-3D-SUPII had five main functions. They are 1) elastic static analysis with multiple load cases based on control program SOL01; 2) elastic or inelastic analysis of a structure subject to 3D ground acceleration based on SOL02; 3) calculation of either natural frequencies and mode shape or buckling load and mode shape of an elastic structure based on SOL03; 4) calculation of nonlinear static cyclic response of a given loading pattern which consists of joint loads, imposed displacements, or element loads based on SOL04; and 5) calculation of maximum response of an elastic structure subject to pseudo-dynamic force obtained from the response spectrum based on SOL05. INRESB-3D-SUPII pertains to analysis of building systems above ground level. Any connection with basements is treated as a boundary condition fixed, hinged, or with a spring.**

17. Document Analysis a. Descriptors

3D structure
Buckling
Building system
Dynamic analysis
Finite-segment element
FORTRAN
Ground acceleration
Inelastic analysis
Mode shape
Multicomponent earthquake motion
Natural frequency
Nonlinear material
Personal computer
Post-buckling
Reinforced concrete
Pseudo-dynamic force
Response spectrum
Shear wall

b. Identifiers/Open-Ended Terms

c. COSATI Field/Group

18. Availability Statement

19. Security Class (This Report) Unclassified
21. No. of Pages 111
20. Security Class (This Page) Unclassified
22. Price