REPORT NO.
UCB/EERC-82/21
NOVEMBER 1982

EARTHQUAKE ENGINEERING RESEARCH CENTER

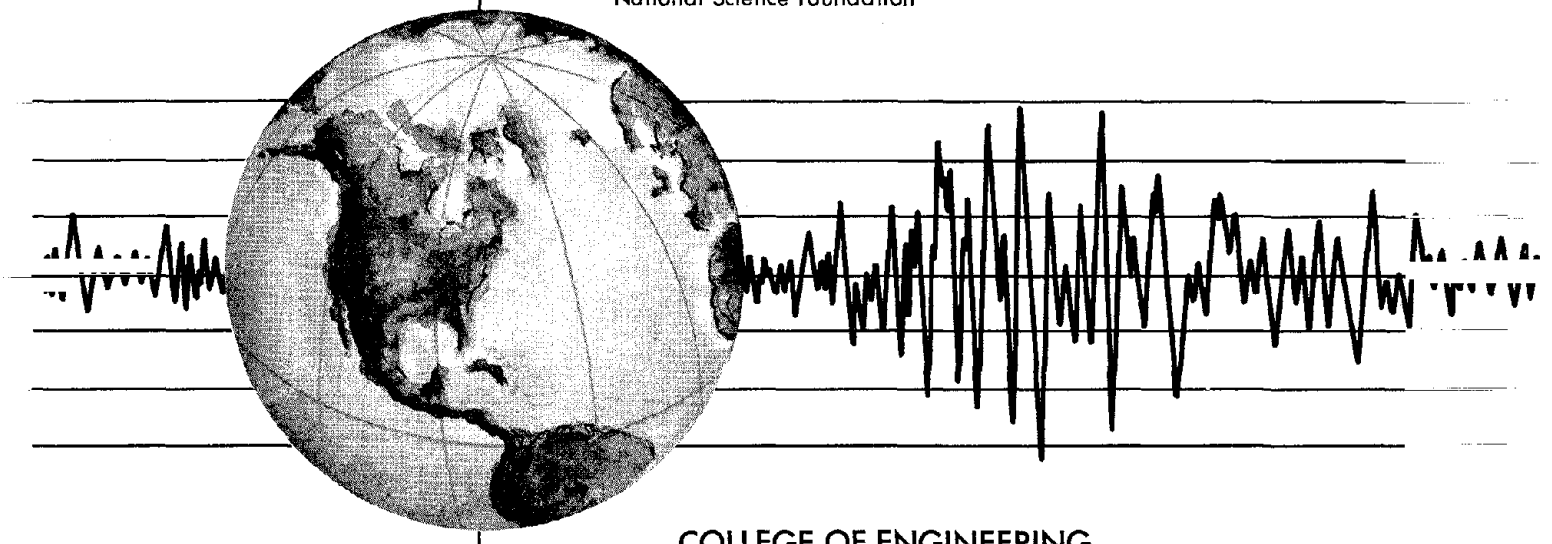
ANSR - III

GENERAL PURPOSE COMPUTER PROGRAM FOR NONLINEAR STRUCTURAL ANALYSIS

by

CARLO VAHE OUGHOURLIAN
GRAHAM H. POWELL

Report to Sponsor:
National Science Foundation



COLLEGE OF ENGINEERING

UNIVERSITY OF CALIFORNIA · Berkeley, California

For sale by the National Technical Information Service, U.S. Department of Commerce, Springfield, Virginia 22161.

See back of report for up to date listing of EERC reports.

DISCLAIMER

ANSR-III

GENERAL PURPOSE COMPUTER PROGRAM FOR NONLINEAR STRUCTURAL ANALYSIS

by

Carlo Vahe Oughourlian
Graduate Student

and

Graham H. Powell
Professor of Civil Engineering

Report to
National Science Foundation
under Grant No. CEE 8105790

Report No. UCB/EERC-82/21
Earthquake Engineering Research Center
College of Engineering
University of California
Berkeley, California

November 1982

ABSTRACT

ANSR is a general purpose computer code for the static and dynamic analysis of nonlinear structures. The first two versions of the code (ANSR-I and ANSR-II) were released in 1975 and 1979, respectively. A third version (ANSR-III) is described in this report. This version is a substantially extended and improved version, which has the following new features.

(1) A flexible and substantially automatic static analysis strategy.

(2) Two separate options for dynamic analysis, namely a new event-to-event scheme and an improvement of the earlier iteration scheme.

(3) Visco-static and creep analysis options.

(4) An option for rotated coordinate systems.

(5) Configuration-dependent static loading.

(6) Time-history record interpolation to allow variable time step analysis and arbitrary time delays for loading by propagating waves.

(7) An option for the Hilber-Hughes-Taylor integration method for dynamic analysis.

(8) Simpler and more modular iteration logic for static, dynamic and visco-static analysis.

(9) Energy balance calculations for dynamic analysis.

(10) Reorganization of element data and computational modules for greater generality and flexibility.

(11) An improved element library.

This report describes the planning and design, and most of the features, of ANSR-III. For some features, reference is made to the work

of other students. A detailed user guide for the program is provided, with explanatory notes, and the procedures to be followed in adding new elements to the program are described in detail. The theory for a new element (a U-bar restraint element) is included and used to illustrate the programming procedures.

ACKNOWLEDGEMENTS

The research described in this report was completed under the sponsorship of the National Science Foundation, Grant No. CEE 8105790. The authors wish to express their sincere appreciation for this support.

Sincere appreciation is also expressed to Linda Calvin for preparation of the typescript and to Gail Feazell and Mary Edmunds-Boyle for preparation of the figures.

TABLE OF CONTENTS

	Page
ABSTRACT	i
ACKNOWLEDGEMENTS	iii
TABLE OF CONTENTS	v
1. INTRODUCTION	1
1.1 OBJECTIVE	1
1.2 IMPROVEMENTS INCORPORATED INTO ANSR-III	1
1.3 REPORT LAYOUT	4
2. COMPUTATIONAL ASPECTS OF NONLINEAR ANALYSIS	5
2.1 INTRODUCTION	5
2.2 OPERATIONS FOR LINEAR ANALYSIS	5
2.2.1 General	5
2.2.2 Data Input	5
2.2.3 Steps for Linear Static Analysis	6
2.2.4 Data Modules for Linear Static Analysis	7
2.2.5 Linear Step-By-Step Dynamic Analysis	8
2.2.5.1 Newmark, β = 1/4 Method	8
2.2.5.2 Hilber-Hughes-Taylor (HHT) Method	9
2.2.5.3 Hibbitt's Modification of the HHT Method	10
2.3 OPERATIONS FOR NONLINEAR ANALYSIS	11
2.3.1 General	11
2.3.2 Data Input	11
2.3.3 Steps for Nonlinear Static and Dynamic Analysis	12
2.3.4 Data Modules for Nonlinear Analysis	13
2.4 SOME COMPUTATIONAL ASPECTS OF NONLINEAR ANALYSIS	14
2.4.1 General	14

Table of Contents (cont'd) Page

2.4.2 Stiffness Matrix Modifications 15

2.4.3 Displacement Vector Calculations 16

2.4.4 Structure State Determination 17

2.4.5 Time Step Control in Dynamics 17

2.4.5.1 Midstep Error for Linear Problem 17

2.4.5.2 Midstep Error for Nonlinear Problem . . 18

2.4.6 Event-to-Event Strategy 19

2.4.7 Dynamic Load Scaling Using the Event-to-Event Strategy 20

2.4.8 Energy Calculations 21

2.5 SUMMARY OF COMPUTATIONAL MODULES 22

2.5.1 General 22

2.5.2 Modules at Structure Level 22

2.5.2.1 Input Phase 22

2.5.2.2 Analysis Phase 23

2.5.2.3 Output Phase 24

2.5.2.4 Data Management 24

2.5.3 Modules at Element Level 25

2.5.3.1 Input Phase 25

2.5.3.2 Analysis Phase 25

2.5.3.3 Output Phase 26

3. ANALYSIS PROCEDURES AND OPTIONS IN ANSR-III 27

3.1 OVERALL ARCHITECTURE OF THE PROGRAM 27

3.2 ANSR-III ANALYSIS SUBROUTINES 27

3.2.1 Subroutines at Structure Level 27

Table of Contents (cont'd) Page

3.2.2 Subroutines at Element Level Called from Structure Subroutines 29

3.3 INPUT AND ANALYSIS PHASES 30

3.3.1 General 30

3.3.2 Input Phase 30

3.3.3 Analysis Phase 30

3.3.4 Static Analysis 31

3.3.5 Dynamic Analysis 31

3.3.6 Visco-Static Analysis 31

3.3.7 Creep Analysis 32

3.4 STEP-BY-STEP DYNAMICS 32

3.4.1 Basic Equations 32

3.4.2 Time Step Control 34

3.4.3 Event-to-Event Strategy 35

3.4.4 Midstep Error Calculation 35

3.4.5 Viscous Damping 36

3.4.6 Energy Calculation 36

3.5 EVENT-TO-EVENT SCHEME WITH VARIABLE TIME STEP 37

3.6 ITERATION SCHEME WITH CONSTANT TIME STEP 38

4. ADDITION OF ELEMENTS 41

4.1 ELEMENT SUBROUTINES 41

4.2 CONTROL VARIABLES 42

4.2.1 Control Variable Categories 42

4.2.2 Structure Control Variables 42

4.2.3 Group Control Variables 43

4.2.4 Element Control Variables 44

4.3 LABELLED COMMON BLOCKS 46
4.3.1 List of Blocks 46
4.3.2 Element Group Information 46
4.3.3 Element Information 47
4.3.4 Event Data 47
4.3.5 Input/Output Units 48
4.3.6 Double Precision Constants 48
4.3.7 Temporary Storage 48
4.4 ELEMENT SUBROUTINES 49
4.4.1 INEL** - Group and Element Data Input 49
4.4.1.1 Beginning Statements 49
4.4.1.2 Variable Types 49
4.4.1.3 Purpose 49
4.4.1.4 Tasks to be Performed 50
4.4.2 STIF** - Element Tangent Stiffness Calculation . 51
4.4.2.1 Beginning Statements 51
4.4.2.2 Variable Types 51
4.4.2.3 Purpose 51
4.4.2.4 Tasks to be Performed 52
4.4.3 STAT** - Element State Determination Calculations 52
4.4.3.1 Beginning Statements 52
4.4.3.2 Variable Types 53
4.4.3.3 Purpose 53
4.4.3.4 Tasks to be Performed 53
4.4.4 RINT** - Internal Resisting Loads 54

Table of Contents (cont'd)

		Page
4.4.4.1	Beginning Statements	54
4.4.4.2	Variable Types	54
4.4.4.3	Purpose	54
4.4.4.4	Tasks to be Performed	55
4.4.5	RDYN** - Initial Dynamic Load	55
4.4.5.1	Beginning Statements	55
4.4.5.2	Variable Types	55
4.4.5.3	Purpose	55
4.4.5.4	Tasks to be Performed	56
4.4.6	INIT** - Initial Static Load	56
4.4.6.1	Beginning Statements	56
4.4.6.2	Variable Types	56
4.4.6.3	Purpose	57
4.4.6.4	Tasks to be Performed	57
4.4.7	EVNT** - Event Factor Calculation	57
4.4.7.1	Beginning Statements	57
4.4.7.2	Variable Types	58
4.4.7.3	Purpose	58
4.4.7.4	Tasks to be Performed	58
4.4.8	MDSE** - Midstep Equilibrium Error	58
4.4.8.1	Beginning Statements	58
4.4.8.2	Variable Types	59
4.4.8.3	Purpose	59
4.4.8.4	Tasks to be Performed	59
4.4.9	OUTS** - Output Element Current State	59
4.4.9.1	Beginning Statements	59

Table of Contents (cont'd) Page

4.4.9.2 Variable Types 60

4.4.9.3 Purpose 60

4.4.9.4 Tasks to be Performed 60

4.4.10 EOUT** - Output Element Envelope Results 60

4.4.10.1 Beginning Statements 60

4.4.10.2 Purpose 60

4.4.10.3 Tasks to be Performed 61

5. U-BAR RESTRAINT ELEMENT 63

5.1 INTRODUCTION 63

5.2 ELEMENT PROPERTIES 63

5.2.1 General Characteristics 63

5.2.2 Normal Displacement Option 64

5.2.3 Viscous Damping Option 65

5.2.4 Stiffness Reformulation Tolerance 66

5.3 THEORY AND COMPUTATIONAL PROCEDURE 67

5.3.1 Element Stiffness 67

5.3.2 Effective Dynamic Stiffness 67

5.3.3 Geometric Stiffness 67

5.3.4 Stiffness Reformulation 68

5.3.5 Displacements Normal to U-Bar Plane 69

5.3.6 Event Factor Correction for Large Displacements . 69

6. EXAMPLE ANALYSES 71

6.1 GENERAL 71

6.2 FRAME WITH BASE UPLIFT 71

6.2.1 General 71

6.2.2 Three-Story Steel Frame 71

Table of Contents (cont'd) Page

6.2.3 Analytical Model 72

6.2.4 Solution Strategies 73

6.2.5 Results 74

6.2.5.1 General 74

6.2.5.2 Third Floor Displacements 74

6.2.5.3 Column Uplift 74

6.2.5.4 Reactions in Restraint Element 75

6.2.5.5 Time Step Size 75

6.2.5.6 Energy Quantities 75

6.2.6 Conclusions 76

7. CONCLUSION 77

7.1 PURPOSE OF STUDY 77

7.2 ACHIEVEMENTS 77

7.3 FUTURE DEVELOPMENTS 78

REFERENCES 79

FIGURES 81

TABLES 125

APPENDIX A - ANSR-III PROGRAM - USER'S GUIDE 135

APPENDIX B - U-BAR RESTRAINT ELEMENT 207

1. INTRODUCTION

1.1 OBJECTIVE

The first two versions of the ANSR program (ANSR-I and ANSR-II) have been extensively used for research in the area of nonlinear structural behavior [1]. The first version (ANSR-I) formed the basis for a practical general purpose computer code, but possessed limitations in terms of modularity, computational efficiency, and solution strategy. ANSR-I also had a limited element library and did not allow the restart of any analysis. The second version (ANSR-II) added several important features, including:

(1) Restart option.

(2) Provision to allow static and dynamic analyses in any sequence.

(3) Static analysis with prescribed nodal displacements as well as prescribed nodal forces.

(4) Out-of-phase support motions for earthquake analysis.

(5) Out-of-core solvers for both symmetrical and unsymmetrical equilibrium equations.

Nevertheless, ANSR-II was limited in such areas as the following:

(1) Only a constant load iteration strategy was included for static analysis. This strategy is not always reliable, especially for buckling problems.

(2) Only constant time steps were permitted for dynamic analysis. This is undesirable for problems of some types, particularly those with large, sudden stiffness changes.

(3) The element library was limited.

In the period since ANSR-II was released, improved static and dynamic strategies and new elements have been developed. The main purpose of this research has been to incorporate all of these improvements into the code in order to produce a greatly improved version, ANSR-III.

1.2 IMPROVEMENTS INCORPORATED INTO ANSR-III

The major improvements made for the new version are as follows:

(1) Improved dynamic capabilities have been added to improve accuracy and reliability for a wider variety of problems. The existing dynamic analysis scheme with iteration has been modified to make it more flexible (not restricted to Newton-Raphson or constant stiffness iteration only). For greater reliability, a new scheme has been added in which variable time steps are applied and an event-to-event scheme is used. This new scheme uses a step-by-step integration strategy which automatically varies the time step to ensure accuracy. This allows large steps to be used when possible, and small steps when necessary. The event-to-event strategy is used to force the solution path to follow the exact path closely, and hence avoid large equilibrium errors. The procedures for the dynamic analysis are discussed extensively in this report.

(2) An extended static analysis strategy has been developed by Simons [2]. This strategy allows "displacement-controlled" iteration, which greatly improves the reliability and convergence rate in many problems. The strategy includes event-to-event and line search options. A detailed description of the strategy is given in [2] and, therefore, is not included in this report.

(3) An improved version of the visco-static analysis, originally developed by Riggs [3] has been incorporated into the new version of the code. A detailed description of the procedure is given in [3] and is not included in this report.

(4) A creep analysis option has been added. The theory and procedure have been explained in detail by Mosaddad [4] and are not included in this report.

(5) Rotated coordinate systems can be specified and related to the global system by means of direction cosines. This feature was originally incorporated into a version of ANSR-II by Morris. The procedure has not yet been documented.

(6) New elements developed by a number of students in the course of their research have been added to the new version. The most important of these are as follows: a beam-column element by Chen [5]; a pipe element by Mahasuverachai [6]; and a shell element by Nour-Omid [7]. Two other elements have also been added, namely an improved

version of the truss bar element and a U-bar restraint element. The U-bar element has been developed as part of the research described herein and is described later in this report.

(7) An option for configuration-dependent static loading has been added. This type of loading permits the analysis of structures which are subjected to loads which change direction as the structure deforms.

(8) The procedure for storage and processing of time-history records has been improved. The record value is now obtained by interpolation during the analysis rather than before the analysis begins. This makes it possible to use varying time steps and simplifies the specification of time delays due to travelling wave effects.

(9) An option for the Hilber-Hughes-Taylor step-by-step integration method [8] has been added. This method introduces numerical damping and may improve the solution stability for certain types of nonlinear problems.

(10) An option to perform an energy balance calculation has been added. The kinetic energy, elastic-plastic work, viscous-damping work, and external work are computed at the end of each step. This provides an estimate of the work done by the unbalanced loads, with the aim of providing information on the reliability of the solution.

(11) The scheme for storage of data has been modified. The procedure for out-of-core storage of element data has been modified to improve the modularity of the program. A storage length for the structure stiffness matrix can be specified to control the size of each block for out-of-core equation blocking. If a single block is sufficient to hold the stiffness matrix in core, an in-core equation solver is used.

(12) The data in the element information block has been reorganized. The element information block is subdivided into three parts, namely state information, stiffness control information, and element stiffness. This format provides greater flexibility in the state determination and stiffness update calculations, allowing new solution algorithms to be developed more easily.

(13) Element subroutines are organized in more modular form. Each element module performs a limited and well-defined task required by the base program. This adds more flexibility and ease in developing new elements for the program.

1.3 REPORT LAYOUT

Chapter 2 reviews the computational aspects of nonlinear analysis, explaining the procedures and options which are needed, and reviews step-by-step iterative algorithms. Chapter 3 describes in detail the architecture of the ANSR-III base program and the element subroutines. This chapter also lays out the details of the dynamic analysis logic and illustrates this logic by means of flow diagrams. Brief descriptions of the static, visco-static and creep analysis options are also given. Chapter 4 describes the procedures which must be followed in adding new elements to the program. The tasks to be performed in each element module are explained in detail. Chapter 5 describes the U-bar restraint element and describes the theoretical and computational aspects of the element. Chapter 6 contains an example analysis (a two-dimensional frame with base uplift) which has been chosen to test and illustrate the dynamic analysis features of ANSR-III. Chapter 7 contains concluding remarks and suggests possible future developments.

Appendix A contains the user guide for the ANSR-III base program, with explanatory notes. Appendix B contains the user guide for the U-bar restraint element, along with a listing of the code illustrating the different element modules which must be developed.

This report does not include descriptions of the other elements available for ANSR-III. Separate reports, some of which were referred to earlier, will be published to describe these elements.

2. COMPUTATIONAL ASPECTS OF NONLINEAR ANALYSIS

2.1 INTRODUCTION

In recent years a variety of problems involving nonlinear structural behavior have been presented to structural engineers, and several different types of nonlinear behavior have been identified. A number of numerical solution schemes have been developed or proposed, but no single scheme has been found to be suitable for all types of nonlinear behavior. Hence, a general purpose computer program for nonlinear analysis must be adaptable to a variety of solution schemes. This requires the identification of solution steps which are common to all schemes and the construction of a modular program which can execute these steps in appropriate sequences.

In this chapter, the computational schemes used in nonlinear analysis are reviewed, and the computational tasks which must be performed in a general purpose computer program are identified.

2.2 OPERATIONS FOR LINEAR ANALYSIS

2.2.1 General

Several of the computational steps and data modules used in linear analysis are used also in nonlinear analysis. The steps for static and dynamic linear analysis are summarized in this section. Extensions for nonlinear analysis are then considered in later sections.

2.2.2 Data Input

The data required to define the structure and loading is essentially as follows.

(1) Node locations.

(2) Structure degrees of freedom (dof). These are typically translational and rotational displacements at the nodes.

(3) Element locations.

(4) Element elastic properties.

(5) Nodal loads. These are external forces applied directly on the nodes.

(6) Element loads. These consist of: (a) external forces applied within the element (e.g., gravity and pressure) and (b) initial strain effects (e.g., temperature change).

The input data may be read directly from an input data file or recalled from a previously established "restart" file, on which processed data has been saved.

2.2.3 Steps for Linear Static Analysis

For linear static analysis, the essential computational steps are as follows.

(1) Form element elastic stiffnesses, $\underline{k}$.

(2) Form element consistent nodal loads, $\underline{Q}_E$, for forces applied within the elements.

(3) Form element initial stresses, $\underline{S}_o$, and corresponding nodal forces, $\underline{Q}_o$.

(4) Assemble element stiffnesses into structure stiffness, $\underline{K}$.

(5) Assemble consistent nodal loads from elements into a structure load vector, $\underline{R}_E$. This can be represented as:

$$\underline{R}_E = \sum \underline{Q}_E \tag{2.1}$$

(6) Assemble initial nodal loads from elements into a structure load vector, $\underline{R}_o$:

$$\underline{R}_o = \sum \underline{Q}_o \tag{2.2}$$

(7) Form load vector, $\underline{R}$:

$$\underline{R} = \underline{R}_N + \underline{R}_E + \underline{R}_o \tag{2.3}$$

where $\underline{R}_N$ = nodal loads.

(8) Solve the equilibrium equations:

$$\underline{K}\,\underline{r} = \underline{R} \tag{2.4}$$

for the displacements, $\underline{r}$. For solution of the equations by direct elimination (Gauss, Crout, etc.), the solution process is in two parts, namely (a) reduction (decomposition) of $\underline{K}$ and (b) reduction and back-substitution of $\underline{R}$.

(9) Determine the element deformations (strains), $\underline{v}$, and hence the element actions (stresses), $\underline{S}$.

(10) Perform an optional equilibrium check. For each element, determine the nodal forces, $\underline{Q}_I$, in equilibrium with the element stresses, $\underline{S}$. Assemble $\underline{Q}_I$ into a structure resisting force vector, $\underline{R}_I$. Calculate the equilibrium error vector, $\underline{R}_U$, as:

$$\underline{R}_U = \underline{R}_N + \underline{R}_E - \underline{R}_I \tag{2.5}$$

The terms of $\underline{R}_U$ should be of the order of the computer round-off error.

(11) Output the results by printing and/or by saving on a postprocessing file.

2.2.4 Data Modules for Linear Static Analysis

The major data modules for static linear analysis are as follows.

(1) Node and degree of freedom data.

(2) Element data.

(3) Load data.

(4) Element stiffnesses, $\underline{k}$; load vectors, $\underline{Q}_E$ and $\underline{Q}_o$; and transformations relating nodal displacements to element actions and deformations.

(5) Structure stiffness, $\underline{K}$. During the equation solving phase, this is typically replaced by its decomposed form.

(6) Structure load vector, $\underline{R}$. During the equation solving phase, this is typically replaced by $\underline{r}$.

(7) Structure external load vector, $\underline{R}_N + \underline{R}_E$, if an equilibrium check is to be performed. During the equilibrium check, this can be replaced by $\underline{R}_U$.

(8) Element actions and deformations.

2.2.5 Linear Step-by-Step Dynamic Analysis

2.2.5.1 Newmark, $\beta = 1/4$ Method

The most commonly used scheme for step-by-step dynamic analysis is the Newmark, $\beta = 1/4$ (constant average acceleration) scheme. This scheme is based on the following equations for the increments of acceleration and velocity in a time step:

$$\Delta \underline{\ddot{r}} = -2\underline{\ddot{r}}_o - \frac{4}{\Delta t}\underline{\dot{r}}_o + \frac{4}{\Delta t^2}\Delta \underline{r} \tag{2.6}$$

$$\Delta \underline{\dot{r}} = -2\underline{\dot{r}}_o + \frac{2}{\Delta t}\Delta \underline{r} \tag{2.7}$$

where

$\Delta \underline{r}$ = increment in displacement;

$\underline{\dot{r}}_o$, $\underline{\ddot{r}}_o$ = velocity and acceleration vectors at beginning of step; and

Δt = time step.

With these relationships, an incremental equilibrium equation can be written as:

$$\left[\frac{4}{\Delta t^2}\underline{M} + \frac{2}{\Delta t}\underline{C} + \underline{K}\right]\Delta \underline{r} = \Delta \underline{R}_E + \underline{M}\left[2\underline{\ddot{r}}_o + \frac{4}{\Delta t}\underline{\dot{r}}_o\right] + 2\underline{C}\,\underline{\dot{r}}_o \tag{2.8}$$

or

$$\underline{K}^*\Delta \underline{r} = \Delta \underline{R}^* \tag{2.9}$$

where

$\underline{K}^*$ = effective stiffness matrix;

$\Delta \underline{R}^*$ = effective load vector;

$\underline{M}$ = mass matrix;

$\underline{C}$ = damping matrix;

$\underline{K}$ = static stiffness matrix; and

$\Delta \underline{R}_E$ = increment of external load for current step.

Hence, for step-by-step dynamic analysis, there are three major differences from linear static analysis, as follows:

(1) Form effective stiffness matrix:

$$\underline{K}^* = \frac{4}{\Delta t^2}\underline{M} + \frac{2}{\Delta t}\underline{C} + \underline{K} \tag{2.10}$$

(2) Form the effective load vector:

$$\Delta \underline{R}^* = \Delta \underline{R}_E + \underline{M}\left[2\ddot{\underline{r}}_o + \frac{4}{\Delta t}\dot{\underline{r}}_o\right] + 2\underline{C}\,\dot{\underline{r}}_o \tag{2.11}$$

(3) Perform equilibrium check by calculating the equilibrium error vector, $\underline{R}_U$:

$$\underline{R}_U = \underline{R}_E - \left[\underline{R}_I + \underline{M}\,\ddot{\underline{r}} + \underline{C}\,\dot{\underline{r}}\right] \tag{2.12}$$

where

$\dot{\underline{r}}, \ddot{\underline{r}}$ = velocity and acceleration vectors at end of step;

$\underline{R}_E$ = external load vector at end of step; and

$\underline{R}_I$ = static internal resisting load vector at end of step.

Additional data modules are needed, as follows:

(1) Mass matrix.

(2) Damping matrix.

(3) Nodal velocities and accelerations.

(4) Additional load vectors.

2.2.5.2 Hilber-Hughes-Taylor (HHT) Method

The HHT method [8] adds numerical damping by solving the following modified equilibrium equation:

$$\left[\frac{4}{\Delta t^2}\underline{M} + \frac{2}{\Delta t}\underline{C} + (1+\alpha)\underline{K}\right]\Delta \underline{r} = \Delta \underline{R}_E + \underline{M}\left[2\ddot{\underline{r}}_o + \frac{4}{\Delta t}\dot{\underline{r}}_o\right] + 2\underline{C}\,\dot{\underline{r}}_o \tag{2.13}$$

where α is a damping parameter ($-1/3 \leqslant \alpha \leqslant 0$).

For dynamic analysis using the HHT method, there are two modifications from the Newmark $\beta = 1/4$ method, as follows:

(1) Form effective stiffness matrix:

$$\underline{K}^* = \frac{4}{\Delta t^2}\underline{M} + \frac{2}{\Delta t}\underline{C} + (1+\alpha)\,\underline{K} \tag{2.14}$$

(2) Perform equilibrium check:

$$\underline{R}_U = \underline{R}_E - \left[\underline{R}_I + \underline{M}\,\ddot{\underline{r}} + \underline{C}\,\dot{\underline{r}}\right] - \alpha\,(\underline{R}_I - \underline{R}'_I) \tag{2.15}$$

where

$\underline{R}'_I$ = internal resisting load vector at beginning of step; and

$\alpha\,(\underline{R}_I - \underline{R}'_I)$ is a correction which must be applied because true equilibrium is not satisfied.

2.2.5.3 Hibbitt's Modification of the HHT Method

Hibbitt and Karlsson [9] further modified the HHT equilibrium equation to the form:

$$\left[\frac{4}{\Delta t^2}\underline{M} + \frac{2}{\Delta t}\underline{C} + (1+\alpha)\,\underline{K}\right]\Delta\underline{r} = (1+\alpha)\,\Delta\underline{R}_E + \underline{M}\left[2\ddot{\underline{r}}_o + \frac{4}{\Delta t}\dot{\underline{r}}_o\right] + 2\underline{C}\,\dot{\underline{r}}_o \tag{2.16}$$

This scheme requires two modifications from the HHT method, as follows:

(1) Form effective load vector:

$$\Delta\underline{R}^* = (1+\alpha)\,\Delta\underline{R}_E + \underline{M}\left[2\ddot{\underline{r}}_o + \frac{4}{\Delta t}\dot{\underline{r}}_o\right] + 2\underline{C}\,\dot{\underline{r}}_o \tag{2.17}$$

(2) Perform equilibrium check:

$$\underline{R}_U = \underline{R}_E - \left[\underline{R}_I + \underline{M}\,\ddot{\underline{r}} + \underline{C}\,\dot{\underline{r}}\right] + \alpha\,\Delta\underline{R}_E - \alpha(\underline{R}_I - \underline{R}'_I) \tag{2.18}$$

where $\alpha\,\Delta\underline{R}_E - \alpha\,(\underline{R}_I - \underline{R}'_I)$ is a correction term because true equilibrium is not satisfied.

2.3 OPERATIONS FOR NONLINEAR ANALYSIS

2.3.1 General

The computational steps involved in nonlinear analysis are similar to those for linear analysis, but the solution sequence is quite different and the data structure is also substantially different. The steps for static and dynamic nonlinear analysis are summarized in this section, and the procedures and strategies used in nonlinear dynamics are explained. The techniques used in nonlinear statics have not been considered in the research described herein and are explained in detail in [2].

2.3.2 Data Input

The data required to define the structure and loading is essentially as follows:

(1) Node locations.

(2) Structure degrees of freedom (dof).

(3) Nodal masses.

(4) Element locations.

(5) Element elastic and inelastic properties.

(6) Nodal loads. These are external forces applied directly on the nodes, which may be configuration-dependent (i.e., the load directions may change as the structure deforms).

(7) Element loads. These consist of: (a) external forces applied within the elements; and (b) initial strain effects.

The input data may be read directly from an input file or recalled from a previously established restart file.

2.3.3 Steps for Nonlinear Static and Dynamic Analysis

For nonlinear analysis, the following computational modules must be programmed, then arranged in appropriate sequences to construct a complete solution strategy.

(1) Form element static tangent stiffnesses, $\underline{k}_T$. The complexity of the calculation depends greatly on the type of element.

(2) Form element effective tangent stiffnesses for dynamics, $\underline{k}_T^*$. These include damping terms and also inertia terms if mass is distributed throughout the elements. The terms vary with the step-by-step integration method being used.

(3) Form element consistent nodal loads, $\underline{Q}_E$, for forces applied within the elements.

(4) Form element initial stresses, $\underline{S}_o$, and corresponding nodal forces, $\underline{Q}_o$.

(5) Form element initial dynamic forces, $\underline{Q}_o^*$, for a time step. These include damping forces, plus inertia forces if masses are distributed throughout the elements.

(6) Assemble the element stiffnesses into the structure stiffness, $\underline{K}^*$, which may be symmetric or unsymmetric and may be held in-core or out-of-core.

(7) Form change in element static tangent stiffnesses, $\Delta \underline{k}_T$.

(8) Form change in element effective tangent stiffnesses, $\Delta \underline{k}_T^*$.

(9) Assemble the element stiffness changes into a change in structure stiffness, $\Delta \underline{K}^*$. If there is only localized nonlinearity, $\underline{K}^*$ may change only locally. Computational effort can be saved by modifying $\underline{K}^*$ rather than forming a complete new matrix.

(10) Assemble the consistent nodal loads from the elements into a structure load vector, $\underline{R}_E$.

(11) Assemble the vector of nodal loads, $\underline{R}_N$, (which may be configuration-dependent) using the current geometry of the structure.

(12) Assemble the initial strain forces, $\underline{Q}_o$, into a structure load vector, $\underline{R}_o$.

(13) Assemble the dynamic initial forces, $\underline{Q}_o^*$, into a vector $\underline{R}_o^*$.

(14) Form the effective load vector, $\Delta \underline{R}^*$, for dynamics as:

$$\Delta \underline{R}^* = \underline{R}_N + \underline{R}_E + \underline{R}_o^* + \underline{R}_o \tag{2.19}$$

(15) Solve the incremental equilibrium equations:

$$\underline{K}^* \Delta \underline{r} = \Delta \underline{R}^* \tag{2.20}$$

for the increment in displacement. If only a change of stiffness, $\Delta \underline{K}^*$, has been assembled, only a partial reduction of $\underline{K}^*$ is needed.

(16) Determine the element deformations (strains), $\underline{v}$, and hence the element actions (stresses), $\underline{S}$. This calculation depends on the type of element and amount of nonlinearity. For complex elements or materials, the calculations may be lengthy.

(17) Accumulate envelope values of stresses and strains.

(18) Accumulate displacements, velocities, and accelerations.

(19) Accumulate displacement, velocity, and acceleration envelopes.

(20) Calculate the unbalanced load vector, $\underline{R}_U$, and check the equilibrium error. For each element, determine the nodal forces, $\underline{Q}_I$, in equilibrium with the element stresses, $\underline{S}$. Assemble $\underline{Q}_I$ into a structure resisting force vector, $\underline{R}_I$. Calculate $\underline{R}_U$ as:

$$\underline{R}_U = \underline{R}_N + \underline{R}_E - \left[\underline{R}_I + \underline{M}\,\ddot{\underline{r}} + \underline{C}_T \dot{\underline{r}}\right] \tag{2.21}$$

(21) Iterate for convergence if desired.

(22) Output the results (current and envelope values) by printing and/or by saving on a post-processing file.

2.3.4 Data Modules for Nonlinear Analysis

The major data modules for nonlinear static and dynamic analysis are as follows.

(1) Node and degree of freedom data.

(2) Element data.

(3) Load data and nodal masses.

(4) Element stiffnesses, $\underline{k}_T$; element load vectors, $\underline{Q}_E$, $\underline{Q}_o$, and $\underline{Q}_o^*$; and transformations relating nodal displacements to element actions and deformations.

(5) Structure stiffness $\underline{K}_T^*$ and change in stiffness $\Delta\underline{K}_T^*$. During the equation solving phase, $\underline{K}_T^*$ is either reformed completely and replaced by its decomposed form, or modified by $\Delta\underline{K}_T^*$ and only selectively decomposed. In the latter case, a duplicate $\underline{K}_T^*$ must be stored to allow for updating.

(6) Structure load vector $\Delta\underline{R}^*$. During the equation solving phase, this is typically replaced by $\Delta\underline{r}$.

(7) Displacements, velocities, and accelerations.

(8) Structure external load vector, $\underline{R}_N + \underline{R}_E$.

(9) Initial load vector, $\underline{R}_o + \underline{R}_o^*$.

(10) Unbalanced load vector, $\underline{R}_U$. During the equilibrium error calculation, this can be replaced by $\Delta\underline{R}^*$ for the next iteration.

(11) Element state data, including actions, deformations, yield status codes, envelope results, etc. It may be necessary to store a duplicate back-up state in case an iterative solution fails to converge or in case a time step has to be repeated.

2.4 SOME COMPUTATIONAL ASPECTS OF NONLINEAR ANALYSIS

2.4.1 General

The computational techniques applied in linear analysis are well established and will not be reviewed herein. The techniques may be substantially different, however, for nonlinear analysis. Some of these differences are considered in this section, with particular emphasis on step-by-step dynamics.

2.4.2 Stiffness Matrix Modifications

In nonlinear analysis, the structure stiffness matrix may be modified at every iteration (e.g. using Newton-Raphson iteration (Fig. 2.1) for a large displacements problem) or may remain constant for several iterations (e.g. using Modified Newton-Raphson iteration (Fig. 2.2) or if the behavior is piecewise linear). In addition, a Quasi-Newton stiffness modification may be used [10]. Further, to allow for situations in which an iterative solution fails to converge, some strategies require that a "back-up" stiffness be saved to allow return to a previously converged state.

Because of these complications, a flexible procedure for forming, modifying and storing the structure stiffness is needed. The procedure adopted for ANSR-III is as follows.

(1) Original and decomposed structure stiffness:

For linear analysis, the decomposed stiffness can overwrite the original because the original will not be needed after decomposition. For nonlinear analysis, however, the decomposed and original stiffnesses must be stored separately because the original must be kept to allow modification and updating.

(2) "Back-up" stiffness:

In static analysis, a "back-up" stiffness may be saved in case the solution fails to converge, to allow the stiffness at the beginning of the step to be restored. As an alternative to saving a "back-up" stiffness, a complete new stiffness may be assembled using "back-up" element states.

In dynamic analysis, a "back-up" stiffness may be saved in case the time step must be reduced and the step repeated. The effective dynamic stiffness includes inertia and damping effects and must be modified if the time step changes. If there is no tangent stiffness ($\beta_T K_T$) damping, and if all mass is lumped at the nodes, the effective dynamic stiffness can be adjusted for the time step change, using the "back-up" stiffness. However, if $\beta_T K_T$ damping is included, assembly of a complete new stiffness is required. This latter

procedure is adopted in ANSR-III.

2.4.3 Displacement Vector Calculations

The equilibrium equations to be solved at frequent intervals during the analysis are of the form:

$$\underline{K}^* \cdot \Delta \underline{r} = \Delta \underline{R}^* \tag{2.22}$$

For dynamic analysis, $\underline{K}^*$ and $\Delta \underline{R}^*$ are effective matrices containing inertia and damping terms.

If all terms in $\Delta \underline{r}$ are initially unknown, the equation solving operation is straight-forward. In some cases, however, certain terms in $\Delta \underline{r}$ are specified. Such imposed displacements (e.g. support displacements due to ground motion) can be accounted for during the equation solving. This is done by skipping over the columns and rows of the corresponding equation during the reduction of the stiffness matrix, $\underline{K}^*$, and subtracting from $\Delta \underline{R}^*$ the inner product of the matrix containing the symmetric part of the skipped columns or rows and the vector containing the imposed displacements. The back-substitution to compute the displacements is then done also by skipping over the equations corresponding to imposed displacements.

The theory of the procedure is as follows. Let equation numbers i and 1 correspond to imposed displacements. The equations could be rearranged and the matrix partitioned as:

$$\underline{K} = \begin{array}{c} \\ \mathrm{i} \\ \mathrm{1} \end{array} \begin{array}{c} \begin{array}{cc} \mathrm{i} & \mathrm{1} \end{array} \\ \begin{bmatrix} \cdots k_{ii} \cdots & k_{i1} \cdots \\ \cdot & \cdot \\ \cdots k_{1i} \cdots & k_{11} \cdots \end{bmatrix} \end{array} \rightarrow \begin{array}{c} \\ \mathrm{i} \\ \mathrm{1} \end{array} \begin{array}{c} \begin{array}{cc} & \mathrm{i1} \end{array} \\ \left[\begin{array}{c|c} \underline{A} & \underline{B} \\ \hline \underline{D} & \underline{C} \end{array}\right] \end{array} \tag{2.23}$$

The equations to be solved can thus be expressed as:

$$\begin{bmatrix} \underline{A} & \underline{B} \\ \underline{D} & \underline{C} \end{bmatrix} \begin{bmatrix} \Delta \underline{r}_r \\ \Delta \underline{r}_d \end{bmatrix} = \begin{bmatrix} \underline{R}_r \\ \underline{R}_d \end{bmatrix} \tag{2.24}$$

or

$$\underline{A} \cdot \Delta \underline{r}_r = \underline{R}_r - \underline{B} \cdot \Delta \underline{r}_d \tag{2.25}$$

and

$$\underline{D} \cdot \Delta \underline{r}_r + \underline{C} \cdot \Delta \underline{r}_d = \underline{R}_d \tag{2.26}$$

where

$\underline{A}$, $\underline{B}$, $\underline{C}$, and $\underline{D}$ = submatrices of $\underline{K}$;

$\Delta \underline{r}_r$ = unknown displacements;

$\Delta \underline{r}_d$ = imposed displacements;

$\underline{R}_r$ = loads corresponding to unknown displacements; and

$\underline{R}_d$ = loads corresponding to imposed displacements.

If $\underline{K}$ is symmetric, then $\underline{A}$ and $\underline{C}$ are also symmetric, and:

$$\underline{D} = \underline{B}^T \tag{2.27}$$

ANSR-II incorporated this option. It has been retained in ANSR-III for both static and dynamic loads. The option provides a convenient means of applying out-of-phase support motions in seismic analysis.

2.4.4 Structure State Determination

With path dependent state determination, element forces $\underline{S}_e$ are calculated at the end of iteration i using the increment in displacement, $\Delta \underline{r}_i$, for that iteration and the current element state (Fig. 2.3). In contrast with path independent state determination, element forces are calculated using the sum of the displacements, $\sum_i \Delta \underline{r}_i$, accumulated since the last reference state was defined (Fig. 2.4). ANSR-III incorporates both options.

The envelope values of element stresses, element strains, and nodal displacements are updated after each state determination. If a back-up state is saved and restored, as when a time step or load step must be repeated, the envelope values of nodal displacement are re-initialized to the values of the beginning of the step.

2.4.5 Time Step Control in Dynamics

2.4.5.1 Midstep Error for Linear Problem

Hibbitt and Karlsson [9] have proposed a criterion for the selection of the time step, Δt. This criterion assumes that if the equilibrium error, $\Delta \underline{R}_m$, at the middle of any time step is

small, then overall equilibrium errors will also be small and the time step is acceptable. The midstep error can easily be calculated for a linear problem, as shown in Fig. 3.1, as:

$$\Delta \underline{R}_m = \frac{\Delta t}{8} \cdot \underline{K} \cdot \Delta \underline{\dot{r}} \tag{2.28}$$

where

$\underline{K}$ = elastic stiffness matrix;

$\Delta \underline{\dot{r}}$ = increment in nodal velocities; and

Δt = time step.

Within each step, a test is made to check whether the time step is too large for acceptable accuracy. This is done by comparing a norm of the midstep error with an upper tolerance. If the tolerance is exceeded, the time step Δt is reduced by a user-specified factor (usually 0.5). If the error is smaller than a lower tolerance for two consecutive steps, then Δt is increased by a second factor (usually 2). This algorithm thus controls the equilibrium error by automatically adjusting the time step.

2.4.5.2 Midstep Error for Nonlinear Problem

For a nonlinear problem, exact equilibrium is never, in general, satisfied either at the end of a step or within the step. Nevertheless, Eqn. 2.28 can still be used to estimate the midstep error, provided that it is recognized that the stiffness may change within the step. If the stiffness does not change, the tangent stiffness matrix must be used when calculating the midstep error, as follows:

$$\Delta \underline{R}_m = \frac{\Delta t}{8} \cdot \underline{K}_T \cdot \Delta \underline{\dot{r}} \tag{2.29}$$

where $\underline{K}_T$ = static tangent stiffness matrix.

With the event-to-event solution strategy (described in the following section), if any event occurs within a step, the midstep error, $\Delta \underline{R}_m$, is weighted by the governing event factor and accumulated over the entire step in the following manner:

$$\Delta \underline{R}_m = \sum_i \alpha_i \, \underline{K}_{Ti} \, \Delta \underline{\dot{r}}_i \tag{2.30}$$

where i = substep number. An alternative method for the computation of the midstep error would be to perform a state determination calculation, and hence, determine the equilibrium error exactly at midstep. With this procedure, however, it is essential to iterate to convergence at the end of the step in order to eliminate any effect due to the unbalanced load $\underline{R}^U$. In general, the use of Eqn. 2.30 is to be preferred because it eliminates the need to iterate and is also more efficient.

The midstep error calculation could be carried out either at the element level (in which case $\Delta\underline{R}_m$ is assembled from the element contributions) or directly at the structure level. In the latter case, the static tangent stiffness matrix $\underline{K}_T$ must be available. In ANSR-III the computation is performed at the element level.

2.4.6 Event-to-Event Strategy

The event-to-event solution strategy for nonlinear analysis provides means of controlling the equilibrium error. Any significant "event" occurring within any element (such as, yielding, nonlinear unloading, gap closure, etc.) determines a substep. The tangent stiffness is modified in each substep, and hence, the solution closely follows the exact response.

If the exact response of the structure is piecewise linear (Fig. 2.5), and if no "overshoot" of the exact event is allowed, the solution will follow the exact load-deflection curve and no equilibrium unbalance will result. In general, however, the response is not linear between well-defined events, and the type of behavior shown in Fig. 2.6 is more likely to be present. In this case, the solution does not follow the exact response, with the result that there is an unbalanced load, $\underline{R}_U$, at the end of the step. This unbalanced load will be small if the events are well chosen (i.e., if they represent major nonlinearities in the behavior of the structure).

To apply the event-to-event strategy, an event factor must be calculated for each element, and the minimum factor chosen to determine a scaling factor for the displacement increment, $\Delta\underline{r}$, for the substep. An event could also be defined in terms of nodal displacement limits. Scaling the displacements $\Delta\underline{r}$ by the minimum event factor requires that the following compli-

cations be considered.

(1) The state determination calculations for elements which have initial strain loadings requires that the governing event factor be used to obtain the correct proportion of the initial strains which are to be used. The state determination calculation is essentially:

$$\Delta \underline{S} = \underline{k}_T(\Delta \underline{v} - \alpha \Delta \underline{v}_o) \tag{2.31}$$

where

$\Delta \underline{S}$ = element force increment;

$\underline{k}_T$ = element tangent stiffness matrix;

$\Delta \underline{v}$ = element strains due to scaled displacement increment;

$\Delta \underline{v}_o$ = initial element strains calculated assuming an unscaled load increment; and

α = governing event factor.

(2) Path dependent state determination is consistent with the event-to-event strategy, whereas the path independent scheme can lead to inconsistencies. This is because the event factor is calculated from the current state. Hence, the state determination calculations should also begin using the current state.

(3) After any event, the load vector may be multiplied by $(1-\alpha)$ to obtain the remaining portion of the load to be applied in the next substep. If initial strain effects are present, however, simply scaling the load vector, $\Delta \underline{R}^*$, is not consistent because the tangent stiffness, $\underline{k}_T$, may have changed, so that the initial strain loads may have to be recomputed. Hence, $\Delta \underline{R}^*$ must generally be recalculated at each substep and cannot be simply scaled.

2.4.7 Dynamic Load Scaling Using the Event-to-Event Strategy

The effective load vector for dynamic analysis contains contributions of initial stress type, and hence, recalculation of the load vector may strictly be necessary at each substep to account for changes in $\underline{k}_T$. For computational convenience, however, $\Delta \underline{R}^*$ can be assumed to be unaffected by changes in $\underline{k}_T$, so that it can be scaled by $(1-\alpha)$ for each substep. In effect, the

solution within the step is found as if the loading were static with no initial strain effects. This can introduce some inconsistencies during the state determination and some unbalanced loads $\underline{R}_U$ (Fig. 2.6) which are not corrected at the end of each substep. Hence, any accumulated unbalances must be corrected in the following step.

An alternative approach is to recalculate $\Delta \underline{R}^*$ at the end of each substep, including unbalanced load effects. This has the disadvantage of requiring an internal resisting force calculation at each substep, which requires the current velocities and accelerations. These quantities are not readily available in consistent form within a substep.

2.4.8 Energy Calculations

The different types of energy which can be calculated are as follows.

(1) Work done by external loads.

(2) Kinetic energy.

(3) Viscous-damping work.

(4) Internal work done by the elasto-plastic forces in the elements.

(5) Work done by seismic base shear forces moving through support displacements.

Ideally, the internal and viscous-damping energies should be calculated exactly in the elements during the state determination phase. However, an alternative and simpler approach is to calculate these energies at the structure level using nodal forces and displacements. With this procedure, the internal energy, $\Delta \underline{E}_I$, a step is calculated as:

$$\Delta \underline{E}_I = \frac{1}{2} (\underline{R}'_I + \underline{R}_I)^T \cdot \Delta \underline{r} \tag{2.32}$$

where

$\underline{R}'_I$ = internal resisting forces at beginning of step; and

$\underline{R}_I$ = internal resisting forces at end of step.

The elastic-plastic work is calculated if $\underline{R}_I$ = resisting forces due to element elastic-plastic actions only, and the viscous damping work is calculated if $\underline{R}_I$ = resisting forces due to viscous

damping effects within the elements. Other energies can be calculated similarly. The base shear energy can be obtained using the imposed nodal displacements and the corresponding nodal forces in the $\underline{R}_I$ vector. The kinetic energy is conveniently calculated as the work done by the inertia forces moving through the corresponding nodal displacements.

After all energies have been computed, a check on the energy balance can provide both a measure of the error produced during a nonlinear dynamic analysis and an idea of the contributions of the different types of energy.

2.5 SUMMARY OF COMPUTATIONAL MODULES

2.5.1 General

In order to perform the types of operations described in the preceding sections, and hence construct a general purpose computer program for nonlinear analysis, a variety of computational modules are needed. These modules are identified in this section.

The modules are classified as being in the input, analysis, or output phases, and as being at the structure or element levels. The modules in the analysis phase are well defined, although they will vary in detail for elements of different types. The modules in the input and output phases are less well defined and can vary in both detail and concept, depending on the level of sophistication required in data input and results processing.

In the lists of Section 2.5.2, each module marked by an asterisk requires an associated module at the element level for each new element type.

2.5.2 Modules at Structure Level

2.5.2.1 Input Phase

1. Node coordinate specification and processing.
2. Rotated coordinate system specification (optional).

3. Nodal mass specification.

4. Identification of structure degrees of freedom.

5. Element data input and initialization.*

6. Load specification and processing.

7. Restart file creation and access.

2.5.2.2 Analysis Phase

1. Complete static or dynamic stiffness assembly, in-core.*

2. Complete static or dynamic stiffness assembly, out-of-core.*

3. Partial static or dynamic stiffness assembly, in-core.*

4. Partial static or dynamic stiffness assembly, out-of-core.*

5. Complete reduction of structure stiffness.

6. Partial reduction of structure stiffness.

7. External load vector formation.

8. Initial static forces assembly.*

9. Initial dynamic forces assembly.*

10. Initial creep forces assembly.*

11. Load vector reduction and back-substitution.

12. Minimum event factor calculation.*

13. Midstep error vector assembly.*

14. Midstep error check and time step selection.

15. Displacement modification for BFGS stiffness modification.

16. Displacement scaling, accumulation, and envelope update.

17. Velocity and acceleration update.

18. State determination and state update.*

19. Static or dynamic internal resisting force assembly.*

20. Load vector scaling for substeps.

21. Unbalanced force vector calculation.

22. Convergence check.

23. Energy balance calculations.

24. Line search for unbalanced force minimization.

25. Nodal envelope values update.

26. Envelope values re-initialization.

2.5.2.3 Output Phase

1. Element state printout.*

2. Envelope values printout.*

3. Element state saving for restart.

4. Element state saving for post-processing.*

5. Nodal values printout.

6. Nodal values saving for post-processing.

2.5.2.4 Data Management

1. Current element state update.

2. Reference element state maintenance and access.

3. Back-up element state maintenance and access.

4. Current element stiffness update.

5. Reference element stiffness maintenance and access.

6. Back-up element stiffness maintenance and access.

7. Load vector maintenance and access.

8. Mass and node-dof connectivity maintenance and access.

9. Restart file maintenance and access.

10. Out-of-core (blocked) structure stiffness maintenance and access.

2.5.3 Modules at Element Level.

2.5.3.1 Input Phase

1. Data input and initialization.

2.5.3.2 Analysis Phase

1. Static tangent stiffness formation.

2. Dynamic tangent stiffness formation.

3. Static stiffness change.

4. Dynamic stiffness change.

5. Static initial force calculation.

6. Dynamic initial force calculation.

7. Creep initial force calculation.

8. Event factor computation.

9. Midstep error calculation.

10. State determination and envelope update.

11. Static internal resisting force calculation.

12. Dynamic internal resisting force calculation.

2.5.3.3 Output Phase

1. Current state printout.
2. Envelope values printout.
3. Envelope re-initialization.
4. Current state saving for post-processing.

3. ANALYSIS PROCEDURES AND OPTIONS IN ANSR-III

3.1 OVERALL ARCHITECTURE OF THE PROGRAM

The overall LOGIC of the ANSR-III program is shown in Table 3.1. In each analysis phase, a number of computational steps of the type considered in Chapter 2 are performed, and certain data is generated. Each type of computational step is executed by an ANSR subroutine (or subroutine sequence). The computational steps are similar for each analysis phase, and hence, different phases can share subroutines at both the structure and element levels. These subroutines are identified in the following section.

3.2 ANSR-III ANALYSIS SUBROUTINES

3.2.1 Subroutines at Structure Level

TEMPLD	═══	Initial Temperature and Pressure Load Assembly
RDYNAM	═══	Initial Dynamic Load Assembly
CRLOAD	═══	Initial Creep Load Assembly
DYNMBC	═══	Boundary Conditions for Dynamics
ICOND	═══	Initial Conditions for Dynamics
STIFST	═══	Assembly of Static Tangent Stiffness or Change in Stiffness
STIFD1,STIFD2	═══	Assembly of Dynamic Tangent Stiffness or Change in Stiffness (for DYNMIT and DYNMIC).
PLD	═══	Nodal Pressure Load Formation
EXTLD	═══	External Static Load Formation
INCLD	═══	Static Load Increment Formation
CONLD	═══	Configuration Dependent Static Load Formation
PDSET	═══	Prescribed Displacements Formation
DYNAP,DYNAPP	═══	Application of Dynamic Loads
EFFL,MEFFL	═══	Effective Dynamic Load Formation and Modification for Imposed Displacements
DISSOL	═══	Solution of Equilibrium Equations

SEVENT,DEVENT	═══	Minimum Event Factor Calculation for Statics and Dynamics
MIDSTEP	═══	Midstep Error Assembly for Time Step Control
BFGS	═══	BFGS Stiffness Modification
SCALE	═══	Displacement Scaling
RESPON	═══	State Determination for Statics
RESPD1,RESPD2, RESPD3,RESPD4	═══	State Determination for Dynamics (DYNMIT,DYNMIC; In-core, Out-of-Core)
CRESP1,CRESP2	═══	State Determination for Creep (In-core, Out-of-Core)
ENVUP	═══	Envelope Values Update
UNBAL	═══	Unbalanced Load Calculation for Static Analysis
UNBALN	═══	Unbalanced Load Calculation for Dynamic Analysis
UNBALL,UNBALV	═══	Unbalanced Load Calculation for Visco-Static Analysis
ENERGY	═══	Energy and Energy Balance Calculations
PRINT,PRINDY	═══	Current State Printout (Static, Dynamic)
ENVOUT	═══	Envelope Values Printout (Nodal and Element)
RSTRT3	═══	Save Results on Restart File

Some of the above subroutines call other subroutines which control element data transfer. These are as follows:

RDYNAM	→	RDYNX	═══	Obtain Initial Dynamic Load from Elements
CRLOAD	→	CRLDX	═══	Obtain Initial Creep Load from Elements
STIFST, STIFD1,	→	STIFX	═══	Obtain Static or Dynamic Tangent Stiffness or Change in Stiffness from Elements
STIFD2				
	→	OPTSPD	═══	In-Core Symmetric Equation Solver
	→	OPSUPD	═══	In-Core Unsymmetric Equation Solver
DISSOL				

	→	OPBSPD	═══	Out-of-Core Symmetric Equation Solver
	→	OPBUPD	═══	Out-of-Core Unsymmetric Equation Solver
RESPON, RESPD1,	→	STATX	═══	State Determination of Elements
RESPD2, RESPD3, RESPD4	→	RINTX	═══	Obtain Internal Resisting Forces from Elements
PRINT, PRINDY	→	OUTSTA→OUTSX	═══	Output Current State of Elements

3.2.2 Subroutines at Element Level Called from Structure Subroutines

For each element, ten computational steps can be identified. For each new element type which is added to the element library, ten subroutines must be developed, one for each computational step. The subroutines are identified as RDYN**, CRLD**, etc., where ** is a two-digit element type number. The element subroutines and the base program subroutines from which they are called are as follows.

Base Program	Element Subroutines
RDYNX →	RDYN01,RDYN02,... = Initial Dynamic Load
CRLDX →	CRLD01,CRLD02,... = Initial Creep Load
TEMPLD →	INIT01,INIT02,... = Initial Strain Loads
STIFX →	STIF01,STIF02,... = Tangent Stiffness
STATX →	STAT01,STAT02,... = State Determination
RINTX →	RINT01,RINT02,... = Internal Resisting Force
SEVENT, DEVENT →	EVNT01,EVNT02,... = Event Factor
MIDSTEP →	MDSE01,MDSE02,... = Midstep Error
OUTSX →	OUTS01,OUTS02,... = Current State Printout
ENVOUT →	EOUT01,EOUT02,... = Results Envelope Printout

3.3 INPUT AND ANALYSIS PHASES

3.3.1 General

The input and analysis phases involve sequences of subroutine calls. These sequences are summarized in this section.

3.3.2 Input Phase

Subroutine INPUT controls storage allocation and data input by calling the following subroutines.

XYZORD	= Input Node Coordinates
DISDOF	= Set up Node DOF Connectivity Array
LUMASS	= Input Nodal Masses
CRINPT	= Input Creep Property Tables (Stored In-Core)
ELINPT	= Input Element Data (Location, Properties, ...): Calls Element Subroutines INEL01, INEL02, etc.
STLOAD	= Input Static Load Patterns
CONFIG	= Input Configuration-Dependent Loads
TEMPIN	= Input Temperature Loads
PRESIN	= Input Pressure Loads
PDPAT	= Input Prescribed Displacement Patterns
AXLREC	= Input Ground Acceleration Records
PTRECD	= Input Dynamic Force Records
SDRECD	= Input Support Displacement Records
BLOCK	= Stiffness Matrix Blocking
RSTRT1, RSTRT2	= Data Saving and Recall Using Restart File

3.3.3 Analysis Phases

The analysis controllers (STATIC, DYNMIC, etc.) string together sequences of operations as considered in Chapter 2 and identified in Section 3.2. The essential features of these

controllers are listed below. The static, visco-static, and creep analyses are only briefly described herein and are explained in detail in [2], [3], and [4]. The dynamic analysis phase has, however, been a part of the current research and is described in detail in Section 3.4.

3.3.4 Static Analysis

The static analysis phase makes use of the following subroutines:

TEMPLD; STIFST; PLD; EXTLD; INCLD; CONLD; PDSET; DISSOL; SEVENT; BFGS; SCALE; RESPON; ENVUP; UNBAL; PRINT; ENVOUT; RSTRT3.

The actual sequence of operations, as controlled by STATIC, depends on the solution strategy being used.

3.3.5 Dynamic Analysis

The dynamic analysis phases make use of the following subroutines:

DYNMBC; ICOND; EXTLD; DYNAP and DYNAPP; EFFL and MEFFL; STIFD1 or STIFD2 (for DYNMIT or DYNMIC); DISSOL; DEVENT (for DYNMIC only); MDSTEP (for DYNMIC only); RESPD1, RESPD2, RESPD3, or RESPD4 (for DYNMIT or DYNMIC, in-core or out-of-core); ENVUP; UNBALN; ENERGY; PRINDY; ENVOUT; RSTRT3.

The sequence of operations is controlled by the DYNMIC or DYNMIT analysis controller.

3.3.6 Visco-Static Analysis

The visco-static analysis phase makes use of the following subroutines:

ICOND; EXTLD; RDYNAM; STIFST; DISSOL; RESPD1 or RESPD2 (in-core or out-of-core); ENVUP; UNBALL; UNBALV; PRINDY; ENVOUT; RSTRT3.

Subroutine VISCOS controls these operations.

3.3.7 Creep Analysis

The creep analysis phase makes use of the following subroutines:

CRLOAD; STIFST; DISSOL; DEVENT; CRESP1 or CRESP2 (in-core or out-of-core); UNBAL; PRINDY.

Subroutine CREEP controls these operations.

3.4 STEP-BY-STEP DYNAMICS

3.4.1 Basic Equations

The step-by-step solution strategy is based on the constant average acceleration assumption. The solution is obtained by solving the following equations:

$$\left[\frac{4}{\Delta t^2}\underline{M} + \frac{2}{\Delta t}\underline{C}_T + \underline{K}_T\right]\Delta\underline{r} = \underline{R}^U + \underline{M}\left[2\ddot{\underline{r}}_o + \frac{4}{\Delta t}\dot{\underline{r}}_o\right] + 2\underline{C}_T\dot{\underline{r}}_o \tag{3.1}$$

$$\Delta\dot{\underline{r}} = -2\dot{\underline{r}}_o + \frac{2}{\Delta t}\Delta\underline{r} \tag{3.2}$$

$$\Delta\ddot{\underline{r}} = -2\ddot{\underline{r}}_o - \frac{4}{\Delta t}\dot{\underline{r}}_o + \frac{4}{\Delta t^2}\Delta\underline{r} \tag{3.3}$$

$$\underline{R}^U = \underline{R}^E - \left[\underline{R}_o^I + \underline{M}\ddot{\underline{r}}_o + \underline{C}_T\dot{\underline{r}}_o\right] \tag{3.4}$$

where

$\underline{K}_T$ = tangent stiffness matrix;

$\underline{C}_T$ = tangent damping matrix;

$\underline{M}$ = mass matrix;

$\underline{R}^U$ = unbalanced load vector at beginning of step;

$\underline{R}^E$ = external load vector at end of step;

$\underline{R}_o^I$ = internal resisting load vector at beginning of step (only elasto-plastic forces included);

$\dot{\underline{r}}_o, \ddot{\underline{r}}_o$ = velocity and acceleration vectors at beginning of step;

$\Delta \underline{r}$, $\Delta \underline{\dot{r}}$, $\Delta \underline{\ddot{r}}$ = displacement, velocity, and acceleration increments; and

Δt = time step.

Eqn. 3.1 can be written as:

$$\underline{K}_T^* \Delta \underline{r} = \Delta \underline{R}^* \tag{3.5}$$

where

$\underline{K}_T^*$ = effective tangent stiffness matrix;

$\Delta \underline{R}^*$ = effective load vector;

$$\underline{K}_T^* = \frac{4}{\Delta t} \underline{M} + \frac{2}{\Delta t} \underline{C}_T + \underline{K}_T \tag{3.6}$$

$$\Delta \underline{R}^* = \underline{R}^E - \underline{R}_o^I + \underline{M} \left[\ddot{\underline{r}}_o + \frac{4}{\Delta t} \dot{\underline{r}}_o \right] + \underline{C}_T \dot{\underline{r}}_o \tag{3.7}$$

If mass and stiffness proportional damping are used:

$$\underline{C}_T = \alpha_m \underline{M} + \beta_o \underline{K}_o + \beta_T \underline{K}_T \tag{3.8}$$

where α_m, β_o, and β_T are proportionality constants. The effective stiffness matrix and effective load vector are then:

$$\underline{K}_T^* = \left[\frac{4}{\Delta t^2} + \frac{2\alpha_m}{\Delta t} \right] \underline{M} + \left[\frac{2}{\Delta t} \beta_o \right] \underline{K}_o + \left[1 + \frac{2}{\Delta t} \beta_T \right] \underline{K}_T \tag{3.9}$$

$$\Delta \underline{R}^* = \underline{R}^E - R_o^I + \left[1 + \alpha_m \right] \underline{M} \ddot{\underline{r}}_o + \left[\frac{4}{\Delta t} + \alpha_m \right] \underline{M} \dot{\underline{r}}_o + \left[\beta_o \underline{K}_o + \beta_T \underline{K}_T \right] \dot{\underline{r}}_o \tag{3.10}$$

For the Hilber-Hughes-Taylor (HHT) scheme, $\underline{K}_T^*$ is modified as follows.

$$\underline{K}_T^* = \left[\frac{4}{\Delta t^2} + \frac{2\alpha_m}{\Delta t} \right] \underline{M} + \left[\frac{2}{\Delta t} \beta_o \right] \underline{K}_o + \left[1 + \frac{2}{\Delta t} \beta_T + \alpha \right] \underline{K}_T \tag{3.11}$$

where α = HHT integration constant.

For Hibbitt's version of the HHT scheme, $\underline{K}_T^*$ is given by Eqn. 3.11 and $\Delta \underline{R}^*$ is modified as follows:

$$\begin{aligned} \Delta \underline{R}^* = {} & (1 + \alpha) \underline{R}^E - \alpha \underline{R}_o^E - \underline{R}_o^I + [1 + \alpha_m] \underline{M} \ddot{\underline{r}}_o \\ & + \left[\frac{4}{\Delta t} + \alpha_m \right] \underline{M} \dot{\underline{r}}_o + [\beta_o \underline{K}_o + \beta_T \underline{K}_T] \dot{\underline{r}}_o \end{aligned} \tag{3.12}$$

3.4.2 Time Step Control

A criterion for time step control based on the equilibrium error at midstep has been proposed by Hibbitt and Karlsson [9]. This criterion assumes that if the midstep error is small, then since the error at the end of the step is zero (assuming iteration to convergence), the overall effect of equilibrium errors should be small, and the calculated response should be accurate.

Hibbitt and Karlsson appear to iterate to convergence in each step, then obtain the midstep error (half step residual) by an explicit equilibrium error calculation at midstep. For the ANSR scheme, use is made of the observation that for a linear problem the midstep equilibrium error, $\Delta \underline{R}_m$, can be easily calculated as shown in Fig. 3.1. Because the midstep value of $\ddot{\underline{r}}$, $\dot{\underline{r}}$, and $\underline{R}$ (assuming a linear variation of $\underline{R}$) are averages of the step end values, the midstep error is related to the displacement deviation at the midstep, $\Delta \underline{r}_m$. The midstep equilibrium error vector, $\Delta \underline{R}_m$, can thus be obtained as:

$$\Delta \underline{R}_m = \underline{K}_T \cdot \Delta \underline{r}_m = \frac{\Delta t}{8} \underline{K}_T \Delta \dot{\underline{r}} \tag{3.13}$$

or, from Eqn. 3.2,

$$\Delta \underline{R}_m = \frac{\Delta t}{8} \underline{K}_T \left(-2\dot{\underline{r}}_o + \frac{2}{\Delta t} \Delta \underline{r} \right) \tag{3.14}$$

For a linear problem this midstep error calculation is exact. For a nonlinear problem it is only approximate but is easy to calculate and provides an effective means of controlling the time step.

In ANSR-III, midstep error calculations are performed at the element level, and $\Delta \underline{R}_m$ is assembled from the element contributions. If the maximum norm of $\Delta \underline{R}_m$ exceeds a user-specified upper tolerance, the time step is reduced by a user-specified factor (usually 0.5). If the norm is less than a lower tolerance for two consecutive steps, the time step is increased by a second user-specified factor (usually 2.0).

3.4.3 Event-to-Event Strategy

The event-to-event strategy used in the dynamic analysis is an extension of the static event-to-event procedure. Fig. 3.2 illustrates behavior which is linear between well defined events. In this case, the solution follows the exact response path ABCD. In contrast, with Newton-Raphson iteration, the solution follows path AB'B''C'C'', the load unbalances R_1^U and R_2^U being corrected during the iteration.

The main disadvantage of using the event-to-event strategy is the need for calculation of event factors. The strategy has the advantage, however, that it follows the true equilibrium path more closely, and hence, tends to be more stable.

A more likely type of behavior, which is not linear between events, is illustrated in Fig. 3.3. In this case, the solution does not follow the exact response path, and an equilibrium error, $\underline{R}^U$, is present at the end of the step. In ANSR-III, this equilibrium error is usually small, because of the time step control and because the choice of events avoids substantial non-linearity in the elements. Any error at the end of the time step can in most cases be eliminated by adjusting the nodal accelerations to satisfy equilibrium. Alternatively, the option is available to apply an equilibrium correction in the following time step. If a structure degree of freedom has zero mass, this second option is automatically used at that degree of freedom.

3.4.4 Midstep Error Calculation

If no event occurs within a time step, the midstep error, $\Delta\underline{R}_m$, is calculated by Eqn. 3.14. If, however, one or more events occur during the step, the complete step is divided into sub-steps. If the proportion of the complete time step is α_i for substep i, then $\underline{R}_m$ is calculated as:

$$\Delta\underline{R}_m = \frac{\Delta t}{8}\sum_i \underline{K}_{Ti}\left[-2\alpha_i \dot{\underline{r}}_o + \frac{2}{\Delta t}\Delta\underline{r}_i\right] \tag{3.15}$$

where $\underline{K}_{Ti}$ = tangent stiffness matrix in substep i and $\Delta\underline{r}_i$ = displacement increment for sub-step i.

3.4.5 Viscous Damping

In ANSR-III, the effective load vector, $\Delta \underline{R}^*$, in Eqn. 3.7 is assumed to remain constant throughout the step (i.e., it is assumed not to be affected by events within the step). This is a correct assumption if $\underline{C}_T$ remains constant during the step but is not correct if $\underline{C}_T$ changes.

In ANSR-III, the option is available to assume mass and stiffness proportional viscous damping, so that $\underline{C}_T$ is:

$$\underline{C}_T = \alpha_m \underline{M} + \beta_o \underline{K}_o + \beta_T \underline{K}_T \tag{3.16}$$

where α_m, β_o, and β_T are proportionality constants. Because the term $\beta_T K_T$ changes as a structure becomes nonlinear, it is recommended that β_T be specified as zero. This is partly to avoid errors due to changes in $\Delta \underline{R}^*$ when events occur in a time step. A more important reason, however, is to avoid the large equilibrium unbalances, and the associated numerical shock, which can occur when the damping forces suddenly change because of a change in $\underline{K}_T$.

3.4.6 Energy Calculation

Inaccuracy or instability in a dynamic step-by-step scheme is most often associated with energy errors. In particular, if the numerical scheme allows progressive accumulation of energy, instability will result.

There are three types of energy to consider, as follows.

(1) Work done by external loads and reactions on the structure and by $\alpha_m \underline{M}$ damping forces.

(2) Kinetic energy.

(3) Internal element energy, consisting of elastic strain energy, plastic work, and work done against viscous damping forces of $\beta \underline{K}$ type.

Ideally, the external work during an analysis should exactly equal the kinetic energy plus the internal energy. This is not difficult to achieve for a linear problem. In the nonlinear case, however, energy errors will generally be present because equilibrium is not exactly satisfied and work is done by the unbalanced loads. Step-by-step strategies have been proposed which can ensure an energy balance [11,12]. However, those strategies are too complex for

implementation in a general purpose program such as ANSR. Hence, energy errors are not eliminated in ANSR-III. Instead, these errors are monitored to provide an indication of whether substantial errors may have developed.

The internal element energy is ideally obtained by calculating the areas under the load-deflection curves for all elements. This procedure has been used, for example, in the DRAIN-2D2 program [13]. This procedure is difficult to implement in ANSR-III, however, because the ANSR elements are much more complex than those in DRAIN-2D2. Hence, a simpler, but approximate, procedure has been used. This procedure involves computation of the work done by the internal resisting load, $\underline{R}^I$, moving through the corresponding nodal displacements. This procedure is only approximate because it ignores any nonlinearity within the step (see Fig. 3.4).

In ANSR-III, the internal element work in time step i is calculated as:

$$\Delta W_I = \tfrac{1}{2}\left(\underline{R}^I_{i-1} + \underline{R}^I_i\right)^T \cdot \Delta \underline{r}_i \tag{3.17}$$

where $\underline{R}^I_{i-1}$ = resisting force at end of step i-1; $\underline{R}^I_i$ = resisting force at end of step i; and $\Delta \underline{r}_i$ = displacement increment in step i.

The kinetic energy and the work done by the external forces are obtained in a similar way.

3.5 EVENT-TO-EVENT SCHEME WITH VARIABLE TIME STEP

The flow diagram in Table 3.2 illustrates, the logic followed by the event-to-event strategy with time step control. The major operations are as follows for each time step.

(1) Begin with the state at the beginning of the step.

(2) Set up the load vector, $\Delta \underline{R}^*$.

(3) Increment the substep counter (NSUB) and check against the maximum allowable number of substeps (MAXSUB). If MAXSUB is exceeded, reduce the time step (DT) by the factor CRED, reform the stiffness matrix, and repeat from Step (1).

(4) Solve the equation: $\underline{K}^*_T \Delta \underline{r} = \Delta \underline{R}^*$.

(5) Compute the velocity increment, $\Delta \underline{\dot{r}}$.

(6) Calculate the event factors and retain the minimum factor (FACTOR).

(7) Calculate the midstep error and assemble $\underline{R}_m$.

(8) Compare the maximum norm of $\underline{R}_m$ against the upper tolerance (TOLUP). If the tolerance is exceeded, repeat from Step (3).

(9) Check the norm of $\underline{R}_m$ against the lower tolerance (TOLOW). If the tolerance is exceeded, set the counter LOTOL to one.

(10) Check if FACTOR = 1. If so, the step is complete and the displacements, velocities, and accelerations may be updated. If not, scale the displacement, $\Delta \underline{r}$, by FACTOR.

(11) Perform a state determination.

(12) If the step is complete, check LOTOL. If LOTOL is zero, increase the time step indicator (INCDT) by one, and check if equal to two. If so, increase DT by the factor CINC, reset INCDT to zero, reform $\underline{K}_T^*$, and continue from Step (1).

(13) If the step is not complete, modify $\underline{K}_T^*$, compute the remaining proportion of the load vector by scaling $\Delta \underline{R}^*$ by (1 - FACTOR), then continue from Step (3).

3.6 ITERATION SCHEME WITH CONSTANT TIME STEP

The flow diagram in Table 3.3 illustrates the logic followed by the iteration solution strategy with constant time step. The major operations are as follows for each time step.

(1) Initialize the state at the beginning of the step.

(2) Set up the load vector, $\Delta \underline{R}^*$.

(3) Update the velocities and accelerations by the terms not involving the displacement $\Delta \underline{r}$.

(4) Begin the iteration. Increment the iteration counter (NIT) by one.

(5) Set the stiffness update code (ISTIF) and state update code (ISTAT) to zero or one depending on the solution scheme used (e.g. Newton-Raphson, path dependent, etc.).

(6) If ISTIF=1, modify $\underline{K}_T^*$.

(7) If ISTAT=1, reinitialize the displacements and update the state.

(8) Solve the equation: $\underline{K}_T^* \Delta \underline{r} = \Delta \underline{R}^*$.

(9) Increment the velocities, accelerations, and displacements.

(10) Perform the state determination.

(11) Perform an equilibrium check, by calculating $\underline{R}_U$, checking the Euclidean norm or maximum norm against the convergence tolerance. If the tolerance is exceeded, set $\Delta \underline{R}^* = \underline{R}_U$ and perform the next iteration from Step (4).

(12) Begin the next step, from Step (1).

4. ADDITION OF ELEMENTS

4.1 ELEMENT SUBROUTINES

Ten interface subroutines are called from the base program to perform element computations. These subroutines must be provided for each new element type that is added to the element library. The names of these subroutines and their functions are as follows. The "**" in the subroutine name must be the element type number in the library.

SUBROUTINE NAME	FUNCTION
(1) INEL**	Input and initialize element group data and general element data.
(2) STIF**	Calculate element static or dynamic total stiffness or stiffness change.
(3) STAT**	Perform element state determination calculations.
(4) RINT**	Calculate internal resisting forces for current state.
(5) RDYN**	Calculate dynamic initial forces.
(6) INIT**	Calculate static initial forces.
(7) EVNT**	Calculate event factor for event-to-event strategy.
(8) MDSE**	Calculate midstep equilibrium error for control of time step.
(9) OUTS**	Output element current state.
(10) EOUT**	Output element envelope results.

The subroutine names for Element 1 (truss bar element), for example, are INEL01, STIF01, STAT01, etc. Additional subroutines may need to be added if new analysis options are added. For example, for creep analysis, a subroutine CRLD01 is required for creep initial force calculation.

The interface subroutines may call other subroutines as appropriate. It is recommended that all subroutines called from the interface subroutines be identified with the element type number (e.g. MULT01). This will avoid duplicate subroutine names.

4.2 CONTROL VARIABLES

4.2.1 Control Variable Categories

A number of variables are used to control the computation. These variables can be grouped into three categories, as follows.

(1) Structure control variables.

(2) Group control variables.

(3) Element control variables.

The control variables are defined in the following sections. The procedures for using them are described later.

4.2.2 Structure Control Variables

Structure control variables are defined at the structure level and are the same for all element groups. All control variables in this category are set by the base program.

ISTEP	= Current step number in the step-by-step integration sequence. For static analysis, ISTEP $\leqslant$ 0. For dynamic analysis, ISTEP $\geqslant$ 1.
TIME	= Current time in a dynamic analysis.
KPR	= Print indicator. If results are not to be printed, KPR = 0; otherwise, KPR = element group number.
NJT	= Total number of nodes in the structure.
NDKOD(NJT,6)	= Array containing the structure degree-of-freedom (equation) numbers corresponding to the X,Y,Z displacements and X,Y,Z rotations for each node of the structure.
X(NJT),Y(NJT),Z(NJT)	= Nodal coordinates.
TEMPN(NJT),PRESN(NJT)	= Nodal temperatures and pressures.
ZERO,ONE	= Variables set to 0.0 and 1.0, respectively, to simplify double precision conversion.
ISTFC	= Indicator specifying whether the total element stiffness or only the change in stiffness is required (1 = total; 0 = change).

C7,C8 = Integration constants (e.g. for Newmark $\beta = 1/4$ method, C7 = 1 and C8 = 0).

EVFAC = Event factor for governing event in event-to-event analysis.

IEVEL = Element number corresponding to governing event.

IEVGR = Element group number corresponding to governing event.

IEVNT = Event type number (depends on element type).

4.2.3 Group Control Variables

Group control variables are defined at the element group level and are the same for all elements in the current element group. Some of these control variables are set by the base program, and some must be set within the element subroutines.

DKO = Initial stiffness damping factor, β_0.

CDKO = Value of constant $C\beta_0$ (e.g. for Newmark $\beta = 1/4$ method, C = 2/DT, where DT = time step) in effective stiffness calculations.

DKT = Current tangent stiffness damping factor, β_T.

CDKT = Value of constant $C\beta_T$ in effective stiffness calculations.

NGR = Element group number.

NELS = Number of elements in group.

MFST = Element number of first element in group.

IGRHED(10) = Array containing group heading, in 10A4 format.

NINFC = Number of words of information stored for each element in group. This is the length of the labelled common block /INFEL/ (see Section 4.3) and is also equal to the sum of LSTAT, LSTC, and LSTF (defined below). To compute the length for double precision, each integer variable counts as one word (4 bytes) and each real variable as two words (8 bytes). If integer variables are used at the beginning or middle of /INFEL/, it is advisable to arrange them in even numbered groups, so that all real variables begin on 8-byte boundaries. This is necessary to avoid segmentation problems on some computers.

LSTAT = Number of words of *state information* stored for each element in /INFEL/ (see Section 4.3).

LSTC = Number of words of *stiffness control information* stored for each element in /INFEL/ (see Section 4.3).

LSTF = Number of words of *stiffness information* stored for each element in /INFEL/ (see Section 4.3).

NDOF = Number of degrees of freedom for elements in group (number of rows in element stiffness matrix).

EPROP(1000) = Array which may contain element group information (i.e. information which is common to all elements in a group). This avoids having to store the information separately for each element.

The control variables DKO, DKT, NGR, NELS, MFST, and IGRHED(10) are set by the base program when the input data is read for each element group. The control variables CDKO and CDKT are computed by the base program and provided for each element group for use in the element stiffness subroutine STIF**. The control variables NINFC, LSTAT, LSTF, LSTC, and NDOF must be set in the element input subroutine INEL**.

4.2.4 Element Control Variables

Element control variables are defined at the element level and may be different for each element. Some of these variables are set by the base program, and some must be set within the element subroutines.

IMEM = Element number within its group.

NDF = Number of element degrees of freedom (number of rows in element stiffness matrix).

FK(NDF,NDF) = Array for returning the total element stiffness (if ISTFC = 1), or the change in stiffness (if ISTFC = 0).

INDFK = Indicator determining the element stiffness storage scheme. For a square matrix, INDFK = 0; for the lower half of the matrix only, compacted column-wise, INDFK = 1.

ISTFC = Indicator used to specify if array FK contains the total element stiffness or the change in stiffness. If ISTFC = 1, FK contains the total stiffness. If ISTFC = 0, FK contains the change in stiffness.

KST Stiffness update code, as follows:

(a) KST=1: The tangent stiffness of the element has changed in the current step.

(b) KST=0: No change in element stiffness has occurred.

KEXEC	= Data check parameter. If any error occurs in the input data, set KEXEC to 1. Execution will then stop at the end of the input phase.
EFAC	= Element event factor (proportion of the current displacement increment at which the next nonlinear event occurs).
IEV	= Event type indicator (a code to indicate whether the event is yield, unloading, gap closure, etc.). The exact meaning of IEV varies with the element type.
LM(NDF)	= Element "location matrix" containing numbers that refer to the location in NDKOD(NJT,6) of the structure degree of freedom number corresponding to each element degree of freedom.
Q(NDF)	= Increment in nodal displacements.
VEL(NDF)	= Current nodal velocities.
DVELE(NDF)	= Increment in nodal velocities. This is replaced by the midstep error vector for the current element in MDSE**.
ACC(NDF)	= Current nodal accelerations.
FE(NDF)	= Elasto-plastic resisting forces for the current element.
FD(NDF)	= Viscous resisting forces for the current element.
FZ(NDF)	= Initial dynamic nodal forces for the current element for the next time step.
RF(NDF)	= Initial static forces due to initial strain effects.

The variable ISTFC is set by the base program. The arrays Q, VEL, and ACC are sent in by the base program. The variables IMEM, NDF, INDFK, KST, KEXEC, EFAC, IEV, and the arrays FK, LM, FE, FD, FZ, and RF must be set in the element subroutines. The array DVELE is sent in by the base program and modified in the element subroutine.

4.3 LABELLED COMMON BLOCKS

4.3.1 List of Blocks

The following labelled common blocks are used for element processing:

(1) Element Group Information:

COMMON /INFGR/ NGR,NELS,MFST,IGRHED(10),NINFC,

LSTAT,LSTF,LSTC,NDOF,DKO,DKT,EPROP(1000)

(2) Element Information:

COMMON /INFEL/ IMEM,KST,LM(....

(3) Event Data:

COMMON /EVNT/ EVFAC,IEVNT,IEVEL,IEVGR

(4) Input/Output Units:

COMMON /TAPES/ NIU,NOU,NT1,NT2,NT3,NT4,NT5,NT6,NT7,NT8,

NT9,NT10,NT11,NT12,NT13,NT14,NT15,NT16

(5) Double Precision Constants:

COMMON /PRECIS/ ZERO,ONE

(6) Temporary Storage:

COMMON /WORK/ IWORK(4000)

4.3.2 Element Group Information

Common block /INFGR/ contains information common to all elements in the current element group. The variables in /INFGR/ are initialized in subroutine INEL**. The base program stores the data and makes /INFGR/ for the current element group available to subroutines STAT**, STIF**, RINT**, RDYN**, INIT**, EVNT**, MDSE**, OUTS**, and EOUT**. The data in /INFGR/ must not be modified in these subroutines.

4.3.3 Element Information

For each element, a common block /INFEL/, containing element information, must be created and continually updated. All the information to be retained for any element must be contained either within this block or within the /INFGR/ block. The total length of /INFEL/ is NINFC, which is set in INEL** and is the same for all elements in any group. The information must be arranged in the following order:

(1) State Information: This data must begin with variables IMEM, KST, and LM( ), respectively. The remaining data will typically consist of: node numbers; control codes for results output, large displacements, yield behavior, etc.; element stiffness properties; cross-section properties; strain-displacement transformations; stresses and strains; tolerances for event calculations; and peak (envelope) values of stresses and strains.

(2) Stiffness Control Information: This data will typically consist of yield status information, which will be used to control the stiffness update code KST.

(3) Stiffness Information: This data will typically consist of the tangent stiffness matrix. If the stiffness matrix is symmetric, only the lower triangle may be stored.

The /INFEL/ block must appear in all interface subroutines. During the execution of subroutine INEL**, the block must be initialized for each element. It is then transferred, by the base program, to two storage areas (the first for state information, the second for stiffness control and stiffness information). Each time computations for an element are to be carried out, the base program first transfers the appropriate data from the storage areas to the /INFEL/ block, then calls the appropriate interface subroutines. When control is returned from the subroutine to the base program, the base program transfers the updated data to the appropriate storage areas (this is done for STIF** and STAT** only).

4.3.4 Event Data

For each element, event calculations are performed in EVNT**, and an event factor for the element is returned to the base program. The minimum factor is retained by the base pro-

gram, together with the corresponding element group number, element number, and event type indicator. This information is stored in common block /EVNT/ by the base program and may be needed during certain state determination calculations in the STAT** subroutines. For example, if temperature strains are to be accounted for, the event factor is needed in STAT** to get the amount of initial strain to be considered in the current state determination.

4.3.5 Input/Output Units

Common block /TAPES/ contains file numbers assigned by the base program. Input unit NIU and output unit NOU are required by the element subroutines whenever data is to be read or printed. Units NT1 through NT16 are used by the base program as scratch files and must not be used in any element subroutine.

4.3.6 Double Precision Constants

Common block /PRECIS/ contains real constants 0.0 and 1.0. It is recommended that these constants be used whenever real variables are compared with zero or one in IF statements. This avoids the need to change the IF statements when conversion to or from double precision is made.

4.3.7 Temporary Storage

Common block /WORK/ provides a work area for use by the element programmer. This area must be used only for scratch storage within a subroutine and should generally not be used to transfer data between subroutines, because the data will not generally be retained after control is returned from any interface subroutine to the base program.

4.4 ELEMENT SUBROUTINES

4.4.1 INEL** - Group and Element Data Input

4.4.1.1 Beginning Statements

The SUBROUTINE, COMMON, and DIMENSION statements for subroutine INEL** are typically as follows:

```
SUBROUTINE INEL** (NJT,NDKOD,X,Y,Z,KEXEC)
COMMON /INFGR/ NGR,NELS,MFST,IGRHED(10),NINFC,LSTAT,LSTF,
LSTC,NDOF,DKO,DKT,...
COMMON /INFEL/ IMEM,KST,LM(...
COMMON /TAPES/ NIU,NOU,NT1,...,NT16
COMMON /WORK/ ...
COMMON /PRECIS/ ZERO,ONE
DIMENSION NDKOD(NJT,6),X(NJT),Y(NJT),Z(NJT)
```

4.4.1.2 Variable Types

The variables transferred through the argument list are of two types, as follows.

(a) Input: These are set by the base program and must not be modified.

(b) Output: These are set by the interface subroutine and returned for use by the base program. They are typically initialized to zero before entry to the interface subroutine. In certain cases they may be left unchanged in the interface subroutine.

For subroutine INEL**, the variables are of the following types.

(a) Input: NJT, NDKOD, X, Y, Z.

(b) Output: KEXEC.

4.4.1.3 Purpose

Subroutine INEL** is called from the base program once for each element group in the input phase. Its purpose is to read the input data for all elements in the group, echo print the input data in appropriate forms, and initialize the variables in the /INFGR/ and /INFEL/

blocks.

4.4.1.4 Tasks to be Performed

(1) Set the values of KST, NINFC, LSTAT, LSTF, LSTC, and NDOF in /INFGR/. KST must be set to one.

(2) If desired, set up element property tables in EPROP(1000) in /INFGR/.

(3) Read all element properties (node numbers, large displacement codes, material properties, initial states of stress, etc.).

(4) For each element, carry out the following initialization operations.

(a) Set IMEM in /INFEL/ to the element number within the group.

(b) Set up the LM array in the /INFEL/ block. This is done by calling subroutine NCODLM for each element degree of freedom with the statement:

CALL NCODLM (LM(*),NJT,NOD,N)

where * is the element degree of freedom number, NJT is the total number of nodes in the structure, NOD is the global node number, and N is the local degree of freedom number at node NOD.

(c) Initialize all variables in /INFEL/ to appropriate values corresponding to the initial state of the structure.

(d) Call subroutine BAND with the statement:

CALL BAND

to permit the base program to establish information on the form of the structure stiffness matrix. This call may be made any time after the LM array has been defined.

(e) Call subroutine COMPAC with the statement:

CALL COMPAC

to store the element data temporarily on scratch files. This must be made after the data in /INFEL/ is fully initialized.

4.4.2 STIF** - Element Tangent Stiffness Calculation

4.4.2.1 Beginning Statements

The SUBROUTINE, COMMON, and DIMENSION statements for subroutine STIF** are typically as follows.

```
SUBROUTINE STIF** (ISTEP,NDF,CDKO,CDKT,FK,INDFK,ISTFC)
COMMON /INFGR/ NGR,NELS,MFST,...
COMMON /INFEL/ IMEM,KST,LM(...
COMMON /WORK/ ...
COMMON /PRECIS/ ZERO,ONE
DIMENSION FK(NDF,NDF)
```

4.4.2.2 Variable Types

The variables transferred through the argument list are of the following types.

(a) Input: ISTEP,NDF,CDKO,CDKT,ISTFC

(b) Output: INDFK,FK

4.4.2.3 Purpose

Subroutine STIF** is called from the base program whenever the structure stiffness is to be formed or modified. If the total element stiffness matrix is to be formed, the subroutine is called once for each element. If a change in stiffness is being formed, the subroutine is called only for those elements which have undergone stiffness changes. The total stiffness or change in stiffness must be returned in array FK, for assembly into the structure stiffness.

4.4.2.4 Tasks to be Performed

(1) Set the value of INDFK, specifying the storage scheme in array FK. If FK stores a square matrix, set INDFK to 0. If FK stores only the lower half of the matrix, set INDFK to 1.

(2) If ISTEP=0, form the static stiffness of the element (add geometric stiffness if appropriate), and return in array FK in the appropriate form. This stiffness is used to set up the static structure stiffness at the beginning of a new static analysis phase.

(3) If ISTEP<0, form either the total tangent stiffness or the change in stiffness, depending on the value of ISTFC (see Section 4.2.4), and return in array FK.

(4) If ISTEP=1, form the dynamic stiffness and return in array FK. This stiffness is used to set up the dynamic structure stiffness at the beginning of a new dynamic analysis phase. The dynamic stiffness matrix is given by:

$$\underline{K}^* = CDKO \cdot \underline{K}_o + (1 + CDKT) \cdot \underline{K}_T$$

where $\underline{K}_o$ = elastic stiffness and $\underline{K}_T$ = tangent stiffness.

(5) If ISTEP>1, form either the total dynamic stiffness or the change in stiffness, depending on the value of ISTFC (see Section 4.2.4) and return in array FK.

(6) Status codes defining the previous and current states of the element, as established in subroutine STAT**, will typically be used to determine the nature of the stiffness change. Set the previous codes equal to the current codes.

4.4.3 STAT** - Element State Determination Calculations

4.4.3.1 Beginning Statements

The SUBROUTINE, COMMON, and DIMENSION statements for subroutine STAT** are typically as follows.

```
SUBROUTINE STAT** (NDF,Q,TIME)
COMMON /INFGR/ NGR,NELS,MFST,...
COMMON /INFEL/ IMEM,KST,LM(...
```

COMMON /EVNT/ EVFAC,IEVNT,IEVEL,IEVGR

COMMON /WORK/ ...

COMMON /PRECIS/ ZERO,ONE

DIMENSION Q(NDF)

4.4.3.2 Variable Types

The variables NDF, Q, and TIME are all of input type.

4.4.3.3 Purpose

Subroutine STAT** is called frequently during the analysis. Its purpose is to update the element state information in /INFEL/, given the current state and an increment of nodal displacement (array Q) set up by the base program.

4.4.3.4 Tasks to be Performed

(1) Perform the state determination computations. These vary greatly with the type of element. In general, the element deformations are determined using the Q array, and hence, all element state quantities in /INFEL/ are calculated and updated.

(2) If initial strain effects are considered, compute nodal increments of temperature (DTEM) and internal pipe pressure (DPRES). This is done with reference to the global node number (NOD) by calling subroutine GETNTP for each node with the statement:

CALL GETNTP (NOD,DTEM,DPRES)

If initial strain effects are present, or if the element contains internal degrees of freedom, the governing event factor, EVFAC, stored in /EVNT/, will be needed for the state determination calculations. In such cases, initial strain quantities will have been stored in INIT** for a full static load increment, and only EVFAC times these initial quantities must be used in the state determination.

(3) Update the envelope values of stresses and strains in /INFEL/.

(4) If the element changes such that its status is different from the status at the last stiffness update (because of yielding or unloading), set the stiffness update code KST in /INFEL/. If large displacement effects are included for the specific element, KST is usually set to 1 in all cases.

4.4.4 RINT** - Internal Resisting Loads

4.4.4.1 Beginning Statements

The SUBROUTINE, COMMON, and DIMENSION statements for subroutine RINT** are typically as follows.

```
SUBROUTINE RINT** (NDF,Q,VEL,FE,FD,TIME)
COMMON /INFGR/ NGR,NELS,MFST,...
COMMON /INFEL/ IMEM,KST,LM(...
COMMON /WORK/ ...
COMMON /PRECIS/ ZERO,ONE
DIMENSION Q(NDF),VEL(NDF),FE(NDF),FD(NDF)
```

4.4.4.2 Variable Types

The variables transferred through the argument list are of the following types.

(a) Input: NDF, Q, VEL, TIME

(b) Output: FE, FD

4.4.4.3 Purpose

Subroutine RINT** is called for each element at the beginning of the analysis and after each state determination if needed. Its purpose is to compute the nodal loads which are in equilibrium with the current state of stress.

4.4.4.4 Tasks to be Performed

(1) For static or dynamic analysis, compute the elasto-plastic nodal forces from the current state data, and return to the base program in the FE array.

(2) For dynamic analysis only (TIME > 0), compute the viscous damping nodal forces from the current state data, and return to the base program in the FD array. FD is given by:

$$f_D = -[DKO \cdot \underline{K}_o + DKT \cdot \underline{K}_T] \cdot \dot{Q}$$

where $\underline{K}_o$ = elastic stiffness; $\underline{K}_T$ = tangent stiffness; and $\dot{Q}$ = velocity array (VEL).

4.4.5 RDYN** - Initial Dynamic Load

4.4.5.1 Beginning Statements

The SUBROUTINE, COMMON, and DIMENSION statements for subroutine RDYN** are typically as follows.

```
SUBROUTINE RDYN** (NDF,Q,VEL,ACC,FZ,C7,C8)
COMMON /INFGR/ NGR,NELS,MFST,...
COMMON /INFEL/ IMEM,KST,LM(...
COMMON /WORK/ ...
COMMON /PRECIS/ ZERO,ONE
DIMENSION Q(NDF),VEL(NDF),ACC(NDF),FZ(NDF)
```

4.4.5.2 Variable Types

The variables transferred through the argument list are of the following types.

(a) Input: NDF, Q, VEL, ACC, C7, C8

(b) Output: FZ

4.4.5.3 Purpose

Subroutine RDYN** is called for each element at the beginning of each time step. Its

purpose is to compute, for the next time step, the initial loads associated with the step-by-step integration method being used.

4.4.5.4 Tasks to be Performed

Only damping contributions to the dynamic initial load vector are typically present at the element level. This is because the mass is typically lumped at the nodes, so that inertia effects are accounted for in the base program. Internal element inertia can, however, be considered. The initial forces must be calculated and returned in array FZ. The precise procedure to be followed depends on the element characteristics. FZ is given by:

$$\underline{f}_z = [DKO \cdot \underline{K}_o + DKT \cdot \underline{K}_T] \cdot [(1 + C7)\,\dot{\underline{Q}} + C8 \cdot \ddot{\underline{Q}}]$$

where $\underline{K}_o$ = elastic stiffness; $\underline{K}_T$ = tangent stiffness; $\dot{\underline{Q}}$ = velocity array (VEL); and $\ddot{\underline{Q}}$ = acceleration array (ACC).

4.4.6 INIT** - Initial Static Load

4.4.6.1 Beginning Statements

The SUBROUTINE, COMMON, and DIMENSION statements for subroutine INIT** are typically as follows.

```
SUBROUTINE INIT** (NJT,NDF,RF)
COMMON /INFGR/ NGR,NELS,MFST,...
COMMON /INFEL/ IMEM,KST,LM(...
COMMON /WORK/ ...
COMMON /PRECIS/ ZERO,ONE
DIMENSION RF(NDF)
```

4.4.6.2 Variable Types

The variables transferred through the argument list are of the following types.

(a) Input: NJT, NDF

(b) Output: RF

4.4.6.3 Purpose

Subroutine INIT** is called for each element at the beginning of each load step. Its purpose is to compute the initial forces caused by temperature and by internal pressure changes in pipes.

4.4.6.4 Tasks to be Performed

(1) Calculate increment of nodal temperature and internal pressure by calling GETNTP as described in Section 4.4.3.4.

(2) Calculate the initial nodal forces due to temperature and pressure changes and return to the base program in the RF array.

4.4.7 EVNT** - Event Factor Calculation

4.4.7.1 Beginning Statements

The SUBROUTINE, COMMON, and DIMENSION statements for subroutine EVNT** are typically as follows.

```
SUBROUTINE EVNT** (NDF,Q,VEL,ACC,EFAC,IEV)
COMMON /INFGR/ NGR,NELS,MFST,...
COMMON /INFEL/ IMEM,KST,LM(...
COMMON /WORK/ ...
COMMON /PRECIS/ ZERO,ONE
DIMENSION Q(NDF),VEL(NDF),ACC(NDF)
```

4.4.7.2 Variable Types

The variables transferred through the argument list are of the following types.

(a) Input: NDF, Q, VEL, ACC

(b) Output: EFAC, IEV

4.4.7.3 Purpose

Subroutine EVNT** is called for each element at frequent intervals during the analysis. Its purpose is to calculate the proportion of the displacement increment, Q, which can be applied to an element before a significant nonlinear event occurs. Typical events are yielding, gap closure, and unloading.

4.4.7.4 Tasks to be Performed

(1) Calculate the element strains using the displacement increment in array Q.

(2) If initial strain effects are considered, it may be necessary to obtain increments for nodal temperature and pressure by calling GETNTP as in Section 4.4.3.4.

(3) Calculate the event factor, EFAC, for the first nonlinear event for the element.

(4) Set IEV to a code which indicates the event type.

4.4.8 MDSE** - Midstep Equilibrium Error

4.4.8.1 Beginning Statements

The SUBROUTINE, COMMON, and DIMENSION statements for subroutine MDSE** are typically as follows.

```
SUBROUTINE MDSE** (DVELE,NDF)
COMMON /INFGR/ NGR,NELS,MFST,...
COMMON /INFEL/ IMEM,KST,LM(...
COMMON /WORK/ ...
```

```
COMMON /PRECIS/ ZERO,ONE
DIMENSION DVELE(NDF)
```

4.4.8.2 Variable Types

The variables transferred through the argument list are of the following types.

(a) Input: NDF

(b) Output: DVELE

4.4.8.3 Purpose

Subroutine MDSE** is called for each element at frequent intervals during dynamic analysis. Its purpose is to calculate an approximation to the midstep equilibrium error, and return it to the base program. The equilibrium errors are assembled by the base program and used to control the time step size.

4.4.8.4 Tasks to be Performed

(1) Calculate the midstep error vector as the product of the tangent static stiffness matrix and the increment of nodal velocities transferred in array DVELE.

(2) Replace DVELE by the midstep error vector.

4.4.9 OUTS** - Output Element Current State

4.4.9.1 Beginning Statements

The SUBROUTINE and COMMON statements for subroutine OUTS** are typically as follows.

```
SUBROUTINE OUTS** (KPR,TIME)
COMMON /INFGR/ NGR,NELS,MFST,...
COMMON /INFEL/ IMEM,KST,LM(...
COMMON /TAPES/ NIU,NOU,NT1,...NT16
```

4.4.9.2 Variable Types

The variables KPR and TIME are both of input type.

4.4.9.3 Purpose

Subroutine OUTS** is called for each element at specified output intervals, as controlled by the base program. Its purpose is to print the current element results (stresses and strains, status information, etc.) from the information in /INFEL/.

4.4.9.4 Tasks to be Performed

(1) For the first element in the element group (i.e. if IMEM = MFST) print an appropriate heading for the current element results output.

(2) If KPR is not zero, and if the element print code (set in INEL** and stored in /INFEL/) is also not zero, print the current element results. If KPR=0, or the element print code in /INFEL/ is 0, do not print results.

4.4.10 EOUT** - Output Element Envelope Results

4.4.10.1 Beginning Statements

The SUBROUTINE and COMMON statements for subroutine EOUT** are typically as follows.

```
SUBROUTINE EOUT**
COMMON /INFGR/ NGR,NELS,MFST....
COMMON /INFEL/ IMEM,KST,LM(...
COMMON /TAPES/ NIU,NOU,NT1,...,NT16
```

4.4.10.2 Purpose

Subroutine EOUT** is called for each element at certain time step or load step intervals. Its purpose is to print envelope values of stresses and and strains from data stored in /INFEL/.

4.4.10.3 Tasks to be Performed

(1) For the first element in the element group (i.e., if IMEM = MFST), print an appropriate envelope results heading.

(2) Print the envelope values from /INFEL/.

5. U-BAR RESTRAINT ELEMENT

5.1 INTRODUCTION

The *ubar* type element is intended primarily for modeling U-bar pipe whip restraints of General Electric type. However, the element may be suitable for modeling restraints of other types. The essential features are as follows.

(1) Idealization as an inelastic bar, arbitrarily oriented in space. Resistance along restraint axis only.

(2) Multilinear force-extension relationship in tension (up to 6 linear segments), with initial gap. Inelastic unloading. Zero stiffness in compression.

(3) Option for large displacement analysis, to allow for changes in direction of restraint axis.

(4) Anchorage to a fixed point is required.

(5) Option for pipe displacements perpendicular to U-bar plane to be ignored, to allow unrestrained axial movement of pipe.

(6) Assumed to have negligible mass.

The features of the element are described in physical terms in Section 5.2. The element theory is presented in Section 5.3.

5.2 ELEMENT PROPERTIES

5.2.1 General Characteristics

A restraining device in the form of a U-bar has been designed for use as a pipe whip restraint by the General Electric Company. The device is illustrated diagrammatically in Fig. 5.1.

For analysis, the restraint is idealized as an inelastic bar, as shown in Fig. 5.2. The bar possesses only axial stiffness and exerts a restraining force only along the axis of the restraint. This is a simplified idealization which ignores such effects as bending of the U-bar and friction between the U-bar and the pipe. The idealization may be inaccurate if the pipe displacement is not parallel to the initial U-bar axis.

A restraint element may be arbitrarily oriented in space. The orientation is defined by specifying two nodes at the element ends (Fig. 5.2). Node I is a node of the pipe system. Node J will typically be a fixed anchor point (i.e. a node with specified zero displacements). However, node J may be a structural node if desired. If node J is fixed ("single node" option), the element extension depends on the displacements of node I only. If node J is not fixed, the element extension depends on the relative displacements between I and J, and restraint-forces are exerted at both nodes. Two or more elements may be connected to a single pipe node if desired.

The relationship between axial force and axial extension is assumed to be multi-linear, with up to six linear segments (Fig. 5.3). The stiffness may increase or decrease with increasing extension. The stiffness in any segment may be negative but must always be nonzero. The unloading stiffness is assumed to be equal to the largest of the loading stiffnesses.

In practice, U-bar restraint elements may be quite short in length, so that displacements of the pipe which are not parallel to the initial restraint axis may produce substantial rotations of the element. That is, for structural analysis purposes the displacements may be large, and the influence of change in geometry must be taken into account. The user may specify either that small displacements of the restraint be assumed or that large displacements be taken into account.

5.2.2 Normal Displacement Option

Figure 5.4 shows a typical restraint, in which the axis of the pipe is normal to the plane of the U-bar. In some cases, there may be significant pipe displacements along the pipe axis as well as in the U-bar plane.

When the large displacement option is used, the element extension is the difference between the current and initial element lengths. All displacements of the restrained node, including the component along the pipe axis, thus contribute to the calculated extension. If the U-bar is placed between collars on the pipe, then the bar will rotate as the pipe moves axially,

and the calculated extension will be correct However, if there are no collars present, the U-bar will not be affected by axial movements of the pipe. In particular, closure of the initial gap will be governed only by pipe movements in the U-bar plane.

To allow for this effect, an option is provided to allow node displacements normal to the U-bar plane (i.e. axial movements of the pipe) to be ignored, up to the time the initial gap closes.

Figure 5.4 shows a length of pipe and a U-bar restraint. The restraint element is defined by nodes I and J, as before. In addition, a third node, K, may be specified, such that the IJK plane is normal to the U-bar plane. Typically, the U-bar will be at right angles to the pipe, and node K will be a node on the pipe. The angle JIK will then be a right angle, and the direction of the free movement will be along IK. More generally, K may be any point in the plane normal to the U-bar plane, as shown, in which case the free movement is along line IK', where K' is in the IJK plane and angle JIK' is a right angle. Point K' is determined automatically by the ANSR-III code.

If node K is specified to be nonzero in the input data, pipe displacements along IK' are ignored in computing the new axial length of the element. This is done until the new axial length exceeds the original length plus the clearance. The U-bar then functions to restrain the pipe. After this time, all pipe displacements are used to calculate the axial length of the element (that is, the bar is assumed not to slip along the pipe) until such time as the gap reopens.

5.2.3 Viscous Damping Option

Viscous damping may be specified for the element, if desired, to introduce a simple form of strain rate dependence. If nonzero values are specified for factors β_o and/or β_T, a viscous damper is introduced in parallel with the element. When the element gap is closed, the damping coefficient, C, is given by:

$$C = \beta_0 K_1 + \beta_T K_T$$

where K_1 = stiffness of first segment, as shown in Fig. 5.3, and K_T = stiffness of current seg-

ment (K_1, K_2, etc., depending on restraint extension). The effect is to increase both the stiffness and strength of the element by amounts which depend on the rate of extension of the element. When the restraint is not active (i.e. before the gap closes), the value of C is set to zero.

Appropriate values of β_0 and β_T must be selected using experiment and experience.

5.2.4 Stiffness Reformulation Tolerance

With the ANSR-III strategy for nonlinear analysis, the structure stiffness matrix is modified only when changes occur in one or more elements. If the behavior of the structure is piecewise linear, as is often the case for small displacement analyses, the structure stiffness is modified only at each yield event. For large displacement analyses, the element stiffnesses change continuously, and hence the structure stiffness should strictly be modified in every time step. In many cases, however, the stiffness change from one step to the next may be small, and it may be reasonable to retain the same stiffness for several steps. To allow this, some of the ANSR-III elements contain stiffness reformulation tolerances, which enable the user to control the frequency of stiffness reformulation.

For the U-bar element, stiffness changes occur when the tangent stiffness changes and as the orientation of the element changes (large displacements option). The reformulation tolerance applies to the change of orientation. If the change of orientation is small, the stiffness change will be small, and a modification of the structure stiffness will not be necessary. The reformulation tolerance is an angle. Each time the element stiffness is changed, the direction of the element is saved. If the angle between the current direction and the previous direction is less than the tolerance angle, the stiffness is not changed. An angle of about 0.1 radians is suggested.

5.3 THEORY AND COMPUTATIONAL PROCEDURE

5.3.1 Element Stiffness

The matrix of direction cosines of the line from node J to node I (Fig. 5.2) is

$$\underline{T} = < d_x \; d_y \; d_z > \tag{5.3.1}$$

where d_x, d_y, d_z = direction cosines with respect to the global X,Y,Z axes. For the large displacements option, the direction cosines are continually updated. For the small displacements option, they remain constant.

For any given state of the element, the current extensional stiffness, K, is determined (equal to zero for an open gap; a nonzero value depending on the force in the element when the gap is closed). Hence, the element stiffness matrix, $\underline{K}$, for the "single node" case (node J fixed) is:

$$\underline{K} = \underline{T}^T \cdot K \cdot \underline{T} \tag{5.3.2}$$

For the "two node" case (node J not fixed), the stiffness matrix, $\underline{K}'$, is:

$$\underline{K}' = \begin{bmatrix} \underline{K} & -\underline{K} \\ -\underline{K} & \underline{K} \end{bmatrix} \tag{5.3.3}$$

5.3.2 Effective Dynamic Stiffness

If nonzero values are specified for β_0 and/or β_T, the effective stiffness for dynamic analysis is determined following the standard ANSR-III procedure, except for the following modification. If the gap is closed, then the $\beta_0 K_0$ stiffness is determined using K_1 (Fig. 5.3). However, if the gap is open, the value $\beta_0 K_0$ is assumed to be zero. This is because it would clearly be incorrect to assume the damping is present in an inactive restraint. The element resisting forces due to damping are calculated using the same assumption.

5.3.3 Geometric Stiffness

For large displacement analysis, a geometric stiffness is included. If node J is fixed, the geometric stiffness matrix in terms of local translations (in the local coordinate system x,y,z, with axis x along the axis of the U-bar) is:

$$\underline{K}_G = \frac{N_c}{L_c} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.3.4}$$

where N_c is the current axial force and L_c is the current length of the element.

If node J is not fixed, the geometric stiffness matrix in terms of local translations at I and J is:

$$\underline{K'}_G = \begin{bmatrix} \underline{K}_G & -\underline{K}_G \\ -\underline{K}_G & \underline{K}_G \end{bmatrix} \tag{5.3.5}$$

The matrix $\underline{K'}_G$ is transformed to the global coordinate system, X,Y,Z, as follows:

$$\underline{K''}_G = \underline{T'}^T \cdot \underline{K'}_G \cdot \underline{T'} \tag{5.3.6}$$

in which

$$\underline{T'} = \begin{bmatrix} \underline{T} & \underline{0} \\ \underline{0} & \underline{T} \end{bmatrix} \tag{5.3.7}$$

and $\underline{T}$ is a 3 x 3 matrix of direction cosines of the angles between the local and global axes. That is,

$$\underline{T} = \begin{bmatrix} d_{xX} & d_{xY} & d_{xZ} \\ d_{yX} & d_{yY} & d_{yZ} \\ d_{zX} & d_{zY} & d_{zZ} \end{bmatrix} \tag{5.3.8}$$

in which d_{xX} = cosine of angle between the x and X axes, etc.

In order to compute the direction cosines in matrix $\underline{T}$, the local x-axis is directed along the U-bar (i.e., the first column in $\underline{T}$ will be the direction cosines of the U-bar). The local y-axis can be oriented in any direction in the plane perpendicular to the axis of the U-bar and is determined by fixing a node K'' (similar to node K described in Section 5.2.2). Once the local y-axis is determined, the local z-axis is found by taking the cross product of the x and y axes.

5.3.4 Stiffness Reformulation

The stiffness reformulation code is set if (a) the extensional stiffness, K, changes, or (b) the angle between the current restraint axis and the axis when the stiffness was last reformed exceeds the user-specified tolerance (the latter for the large displacements option only).

5.3.5 Displacements Normal to U-Bar Plane

If node K is specified and the gap is open, the components of nodal displacement parallel to the IK' direction (Fig. 5.4) are ignored for calculating element deformations and direction cosines.

Let $\underline{d}_{IK}$ be the direction cosine matrix of IK'. This matrix is calculated for the initial configuration and is assumed to remain constant during the analysis. Let the vector of increments in nodal translations at I be $\Delta \underline{r}_I$. The components of this displacement increment parallel to IK' are given by

$$\Delta \underline{r}_{IK} = \underline{d}_{IK} \cdot \underline{d}_{IK}^T \cdot \Delta \underline{r}_I \tag{5.3.9}$$

Hence, a modified displacement increment is calculated as

$$\Delta \underline{r}_I' = \Delta \underline{r}_I - \Delta \underline{r}_{IK} \tag{5.3.10}$$

This modified increment is used to calculate the element extension and to update the element orientation.

For the "two node" option, the procedure is applied to the displacements of both node I and node J.

5.3.6 Event Factor Correction for Large Displacements

For the event-to-event solution strategy, an event factor, α_e, is required. For large displacement analysis, the axial deformation, Δv, for the current step is calculated as:

$$\Delta v = L_c - L_p \tag{5.3.11}$$

where

L_c = current length of the bar; and

L_p = previous length of the bar.

Let v be the extension of the bar at the beginning of the step. An event occurs if $v + \Delta v$ exceeds the yield extension, v_y, plus an overshoot tolerance, Δv_t. That is, a yield event occurs when the extension is Δv_e, where:

$$\Delta v_e = v_y - v + \Delta v_t \tag{5.3.12}$$

If the calculated displacement increments at ends I and J of the bar are $\Delta \underline{r}_I$ and $\Delta \underline{r}_J$, an event factor, α_e, is required such that the bar extension due to $\alpha \Delta \underline{r}_I$ and $\alpha \Delta \underline{r}_J$ is equal to Δv_e. The value of α_e is, however, not simply $\Delta v_e / \Delta v$, because the extension is not a linear function of the nodal displacements.

Let

$$\underline{d} = \Delta \underline{r}_J - \Delta \underline{r}_I \tag{5.3.13}$$

and let

$$D = (\underline{d}^T \cdot \underline{d})^{1/2} = \text{length of } \underline{d} \tag{5.3.14}$$

Let I and J be the previous positions of nodes I and J, and I' and J' be the new positions (with an extension Δv), as shown in Fig. 5.5. The relative displacement vector, $\underline{d}$, is thus the vector $J''J'$. From the geometry of triangles $I'HJ'$ and $J''HJ'$, a quadratic relationship between Δv and D can be derived as:

$$(L_p + \Delta v)^2 = (L_p + D\cos\theta)^2 + D^2(1 - \cos^2\theta) \tag{5.3.15}$$

If $\underline{u}$ is a unit vector along IJ, then,

$$\cos\theta = \frac{d^T u}{D} \tag{5.3.16}$$

Solution of Eqn. 5.3.15 for D (taking the positive root) gives:

$$D = -L_p\cos\theta + \sqrt{L_p^2\cos^2\theta + \Delta v^2 + 2L_p\Delta v} \tag{5.3.17}$$

Similarly, if D_e is the length of vector $\underline{d}$ when the event occurs, then:

$$D_e = -L_p\cos\theta + \sqrt{L_p^2\cos^2\theta + \Delta v_e^2 + 2L_p\Delta v_e} \tag{5.3.18}$$

Because D is a linear function of the nodal displacements, it follows that:

$$D_e = \alpha_e D \tag{5.3.19}$$

where α_e is the required event factor. Hence,

$$\alpha_e = \frac{D_e}{D} \tag{5.3.20}$$

6. EXAMPLE ANALYSIS

6.1 GENERAL

Analysis results for a building frame with base uplift (Fig. 6.1) are presented in this chapter. This example has been chosen to test the features of ANSR-III for nonlinear dynamic analysis and to compare accuracy and number of steps for different solution schemes. In particular, the relative merits of the constant and variable time step strategies have been compared. No attempts have been made to compare the computed responses with experimental results. However, the results are compared with the results obtained by the DRAIN-2D2 program [13].

6.2 FRAME WITH BASE UPLIFT

6.2.1 General

The overturning moment at the base of a structure resulting from the lateral inertia forces which occur during a major earthquake can exceed the overturning resistance provided by the dead weight of the structure. Hence, the structure can uplift from its foundation. Experimental studies indicate that allowing for uplift can lead to reduced internal forces and ductility demands, and hence, permit a more economical design.

Experience has shown that nonlinear dynamic analyses of this type of behavior using a constant time step can be inaccurate unless a very small time step is specified. The accuracy of the solution is particularly questionable if there is very stiff impact between the structure and the foundation. This example has been considered to test the ANSR-III dynamic analysis options and to determine whether the variable time step algorithm is significantly more accurate or more economical than the algorithm with constant time step.

6.2.2 Three-Story Steel Frame

The three-story, one-bay steel frame shown in Fig. 6.1 has been studied both analytically and experimentally by Huckelbridge and Clough [14]. For the experimental study, a special mechanism was designed to accommodate the uplift response. Impact pads of two different

stiffnesses were provided beneath the column base (stiffnesses of 40 k/in. for Phase I and 400 k/in. for Phase II).

Figure 6.2 shows the analytical mode shapes and frequencies of the mathematical models used for Phases I and II, as calculated by Huckelbridge. Figure 6.3 shows that the pad stiffnesses have a significant effect on the natural period of the structure.

Huckelbridge investigated two cases analytically, using the DRAIN-2D program [15]. Constant time steps of .01 and .005 second were used for the Phase I and Phase II pad stiffnesses, respectively. The smaller time step for the Phase II analysis was needed to prevent instability in the numerical scheme. The correlation between analysis and experiment was only fair, although satisfactory from an engineering point of view. Only the Phase II pad stiffness was considered in the current study.

6.2.3 Analytical Model

The mathematical model shown in Fig. 6.4 was used to analyze the frame using ANSR-III. Uplift effects were modeled using the U-bar restraint element, acting as a reverse gap element with zero compressive strength. The beams and columns were modeled using the three-dimensional beam-column element developed by Chen [5]. All beams and columns were assumed to behave elastically (i.e., high positive and negative yield strengths were specified). Column extensions were permitted, but beam extensions at the second and third floors were ignored (i.e., equal displacement d.o.f. were specified for the nodes connecting these beams). The P-delta effect was ignored. Masses were associated with both the horizontal and vertical displacements, based on the dead load. Gravity load produced initial stresses, which were obtained by performing a static analysis, with loads applied at the nodes. Based on results obtained by Huckelbridge, it was decided to use original stiffness damping only, with β_o = 0.002196. Damping was introduced in the beam and column elements only, with zero damping in the U-bar elements.

The node and element numbering is shown in Fig. 6.4. The ANSR-III input data is listed

in Table 6.1.

6.2.4 Solution Strategies

The frame was analyzed for the first 7 seconds of the El Centro, 1940, N-S earthquake record shown in Fig. 6.5. The following six cases were studied.

(1) Case A1: Constant time step of .005 second, with the event-to-event strategy (yield overshoot tolerance = .05 kips, and unloading overshoot tolerance = .00005 kips).

(2) Case A2: Constant time step of .005 second, with iteration (unbalance tolerance = .01 kips).

(3) Case B1: Variable time step with the event-to-event strategy, a maximum midstep error of

$$0.05\ (\sum M_i)\ \Delta\ddot{x}_{g_{\max}} = 0.48\ kips;$$

a minimum midstep error of

$$0.005\ (\sum M_i)\ \Delta\ddot{x}_{g_{\max}} = 0.048\ kips;$$

a maximum step size of 0.04 second; and a minimum step size of 0.0004 second.

(4) Case B2: Variable time step with the event-to-event strategy, a maximum midstep error of 1.0 kips; a minimum midstep error of 0.1 kips; a maximum step size of 0.04 second; and a minimum step size of 0.0004 second.

(5) Case B3: Variable time step with the event-to-event strategy, a maximum midstep error of 4.0 kips; a minimum midstep error of 0.4 kips; a maximum step size of 0.04 second; and a minimum step size of 0.0004 second.

(6) Case C1: Variable time step with the event-to-event strategy, a maximum midstep error of 0.48 kips; a minimum midstep error of 0.048 kips; a maximum step size of 0.005 second; and a minimum step size of 0.0003 second.

The maximum step size of 0.04 second in Cases B1, B2, and B3 was chosen equal to the minimum time difference between two consecutive points on the earthquake record, with the

aim of avoiding inaccurate modeling of the ground motion. Energy quantities were calculated.

6.2.5 Results

6.2.5.1 General

The numbers of time steps for the six cases analyzed are given in Table 6.2. Results from the analyses are shown in Figs. 6.6 through 6.33.

6.2.5.2 Third Floor Displacements

Time histories of third floor horizontal displacements for all cases are compared in Figs. 6.6 through 6.11. The following observations may be made.

(1) Figs. 6.6 and 6.11 show that there is virtually no difference between the results obtained for Cases A1, A2, and C1.

(2) Figures 6.7 and 6.8 show that the agreement between Cases B1 and B2 is very close, although the number of time steps for Case B1 was 1141 compared to only 845 for Case B2.

(3) Figures 6.9 and 6.10 show a substantial disagreement between Cases A1 and B3, and between Cases B1 and B3. The reason is considered below.

6.2.5.3 Column Uplift

Time histories of right and left column uplift for all cases, except A2, are compared in Figs. 6.12 through 6.19. The following observations may be made.

(1) Figures 6.15 and 6.19 show that there is virtually no difference between the results obtained in Cases A1 and C1.

(2) Figures 6.13 and 6.17 show that the agreement between Cases B1 and B2 is very close.

(3) Figures 6.12, 6.14, 6.16, and 6.18 show a substantial disagreement between Cases B3 and A1, and a noticeable disagreement between Cases B1 and B3.

6.2.5.4 Reactions in Restraint Element

Time histories of the left restraint force for Cases A1, B1, B3, and C1 are compared in Figs. 6.20 through 6.22. The following observations may be made.

(1) Figure 6.22 shows that there is virtually no difference between the results obtained in Cases A1 and C1.

(2) Figures 6.20 and 6.21 show a substantial disagreement between Cases B1 and A1, and between Cases B3 and B1.

6.2.5.5 Time Step Size

The variations of the time step for Cases B1, B2, B3, and C1 are shown in Figs. 6.23 through 6.26. It can be seen that the time step is greatly reduced at the times of gap closure, with much larger values for the remainder of the response.

6.2.5.6 Energy Quantities

Time histories of accumulated energies for all cases, except A2, are shown in Figs. 6.27 through 6.31. Time histories of accumulated energy unbalances are compared in Figs. 6.32 and 6.33. The following observations may be made.

(1) Figures 6.27 and 6.31 show that there is virtually no difference between the results obtained in Cases A1 and C1.

(2) Figures 6.28 and 6.29 show that the agreement between Cases B1 and B2 is very close.

(3) Figure 6.30 shows a substantial disagreement between Cases B3 and B1, and between Cases B3 and A1.

(4) Figures 6.32 and 6.33 show that the energy unbalances obtained in Cases A1, B1, B2, and C1 are almost identical. However, there is a substantial difference between Case B3 and all other cases.

6.2.6 Conclusions

The following conclusions can be drawn from this example.

(1) The constant time step schemes with a step size of 0.005 second (Cases A1 and A2) calculated the response accurately. This is confirmed by comparing the variable time step scheme with an upper limit of 0.005 second for the step size (Case C1).

(2) The variable time step scheme with an upper limit of 0.04 second for the step size (Cases B1, B2, and B3) failed to calculate the response accurately. The reason is the poor discretization of the acceleration record which results when the time step is allowed to become large, even though this was taken into account in selecting the 0.04 second upper limit.

(3) The agreement between the variable time step schemes with different midstep error tolerances was close. A maximum midstep tolerance of about 5% of the peak nominal inertia force appears to be necessary for an accurate solution. However, it appears that acceptable results can be obtained with a tolerance as large as 40% of the peak nominal inertia forces.

(4) The calculated energy errors are small compared with the total energies for all cases. That is, energy error does not appear to give indication of the accuracy of the solution.

(5) The results obtained by ANSR-III for this example were compared with the ones obtained by the DRAIN-2D2 program [13]. Very close agreement between these sets of results was observed. This further confirms the correctness of the ANSR-III analyses.

(6) More investigation and study is needed to determine the most appropriate value for the upper limit of the time step and for the maximum tolerance on the midstep error. The purpose of the study described in this chapter has been to check the ANSR-III algorithms, not to provide a detailed study of solution strategies.

7. CONCLUSION

7.1 PURPOSE OF STUDY

This report has described the computational steps, features, and options required for a general purpose analysis scheme and computer program for the static, dynamic, visco-static, and creep analysis of nonlinear structures. The purpose of this study has been to implement the scheme in an improved and extended version of the ANSR computer program. The resulting program is believed to be a valuable tool for research and practical analysis, eliminating most of the limitations of the two previous versions of ANSR.

7.2 ACHIEVEMENTS

The following achievements have been accomplished.

(1) The computational operations have been analyzed for different types of structural analysis (linear and nonlinear, static and dynamic).

(2) The computational and data modules required by different analysis algorithms have been identified.

(3) An improved dynamic solution strategy has been developed, based on an event-to-event strategy with automatic time step control.

(4) The existing ANSR dynamic analysis algorithm (an iteration scheme) has been improved and made more flexible compared with ANSR-II.

(5) A complete new version of ANSR has been constructed, incorporating essentially all previous features plus the new dynamic option, improved data handling, improved overall logic, and major improvements developed separately by Simons [2], Riggs [3], and Mosaddad [4]. The overall architecture of the program has been made more modular to permit the addition of new analysis strategies in the future.

(6) Detailed documentation on the element modules has been developed, with instructions on how to add elements to the program. The procedure for adding elements to the base pro-

gram has been simplified compared with ANSR-II, by separating the tasks into more clearly defined modules.

(7) An energy balance option has been added for dynamic analysis.

(8) A sample element (the U-bar restraint element) has been developed, checked, and documented to illustrate the addition of elements to the program.

7.3 FUTURE DEVELOPMENTS

The ANSR-III program is believed to be a powerful general purpose program, suitable for both research and practice. However, more developments are always possible and useful. Some possible major improvements are:

(1) A black-box data base manager could be added to simplify data storage and manipulation. The data handling techniques currently used are still rather cumbersome.

(2) A post-processor capability is needed, with production of a results file and development of post-processing modules.

(3) The element library has been much improved compared with ANSR-II, but a great deal of work remains to develop new elements, particularly with a variety of material models.

It is anticipated that these improvements will be implemented in the future.

REFERENCES

1. Mondkar, D. P. and Powell, G. H., "ANSR-I - Static and Dynamic Analysis of Nonlinear Structures," *EERC Report No. 75-10,* December 1975; and "ANSR-II - Analysis of Nonlinear Structural Response, User's Manual," *EERC Report No. 79-17,* Earthquake Engineering Research Center, University of California, Berkeley, July 1979.

2. Simons, J., "Solution Strategies for Statically Loaded Nonlinear Structures," *Ph.D. Thesis,* Department of Civil Engineering, University of California, Berkeley, June 1982.

3. Riggs, R., "A Crack Model for Finite Element Analysis of Concrete," *Ph.D. Thesis,* Department of Civil Engineering, University of California, Berkeley, December 1981.

4. Mosaddad, B., "Computational Models for Cyclic Plasticity, Rate Dependence and Creep in Finite Element Analysis," *Ph.D. Thesis,* Department of Civil Engineering, University of California, Berkeley, June 1982.

5. Chen, P., "Generalized Plastic Hinge Concept for 3-D Beam-Column Element," *Ph.D. Thesis,* Department of Civil Engineering, University of California, Berkeley, December 1981.

6. Mahasuverachai, M., "Inelastic Analysis of 3-D Pipelines and Tubular Structures," *Ph.D. Thesis,* Department of Civil Engineering, University of California, Berkeley, December 1982.

7. Nour-Omid, S., "General Inelastic Large Deformation Shell Element for the ANSR-II Program," *CE 299 Report,* Department of Civil Engineering, University of California, Berkeley, 1980.

8. Hilber, H. M., Hughes, T.J.R., and Taylor, R. L., "Improved Numerical Dissipation for Time Integration Algorithms in Structural Dynamics," *Earthquake Engineering and Structural Dynamics,* Vol. 5, 283-292, 1977.

9. Hibbitt, H. D. and Karlsson, B. I., "Analysis of Pipe Whip," A.S.M.E. Pressure Vessels and Piping Conference, San Francisco, California, June 25-29, 1979.

10. Mattheis, H. and Strang, G., "The Solution of Nonlinear Finite Element Equations," *International Journal for Numerical Methods in Engineering,* Vol. 14, 1613-1626, 1979.

11. Belytschko, T. and Schoeberle, D. F., "On the Unconditional Stability of an Implicit Algorithm for Nonlinear Structure Dynamics," *Journal of Applied Mechanics,* Vol. 97, 865-936, 1971.

12. Hughes, T.J.R., "Stability, Convergence, and Growth and Decay of Energy of the Average Acceleration Method in Nonlinear Structural Dynamics," *Computers and Structures,* Vol. 6, 313-324, 1976.

13. Golafshani, A., "DRAIN-2D2 - A Program for Seismic Inelastic Response of Structures," *Ph.D. Thesis,* Department of Civil Engineering, University of California, Berkeley, December 1981.

14. Clough, R. W. and Huckelbridge, A. A., "Preliminary Experimental Study of Seismic Uplift of a Steel Frame," *Report No. UCB/EERC-77/22,* Earthquake Engineering Research Center, University of California, Berkeley, August 1977.

15. Kanaan, A. E. and Powell, G. H., "General Purpose Computer Program for Inelastic Dynamic Response of Plane Structures," *Report No. EERC 73-6,* Earthquake Engineering Research Center, University of California, Berkeley, April 1973.

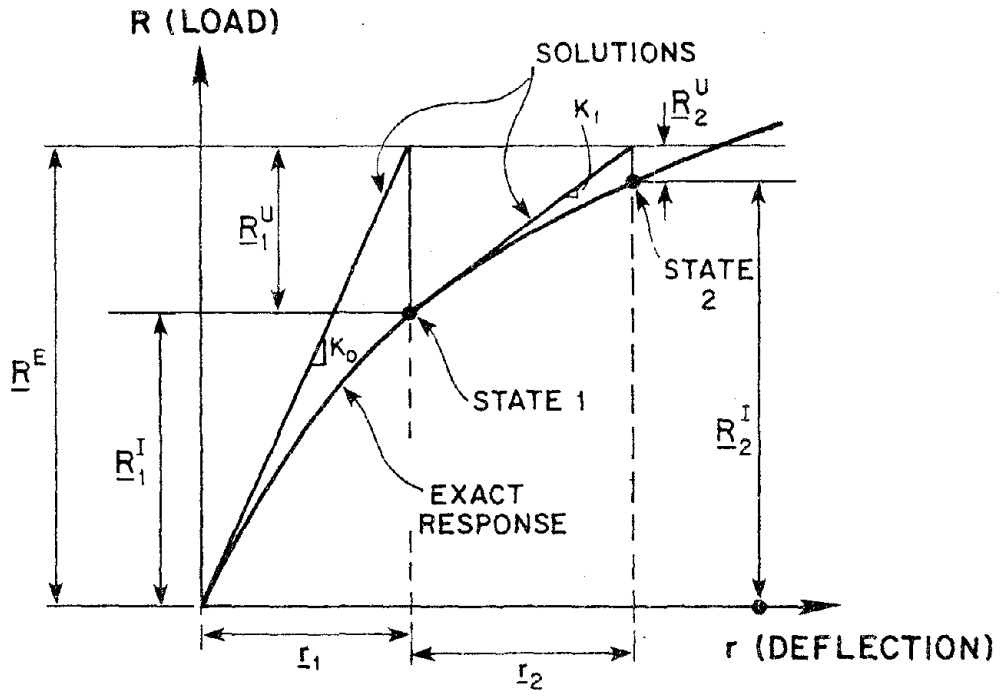


FIG. 2.1 NEWTON-RAPHSON ITERATION

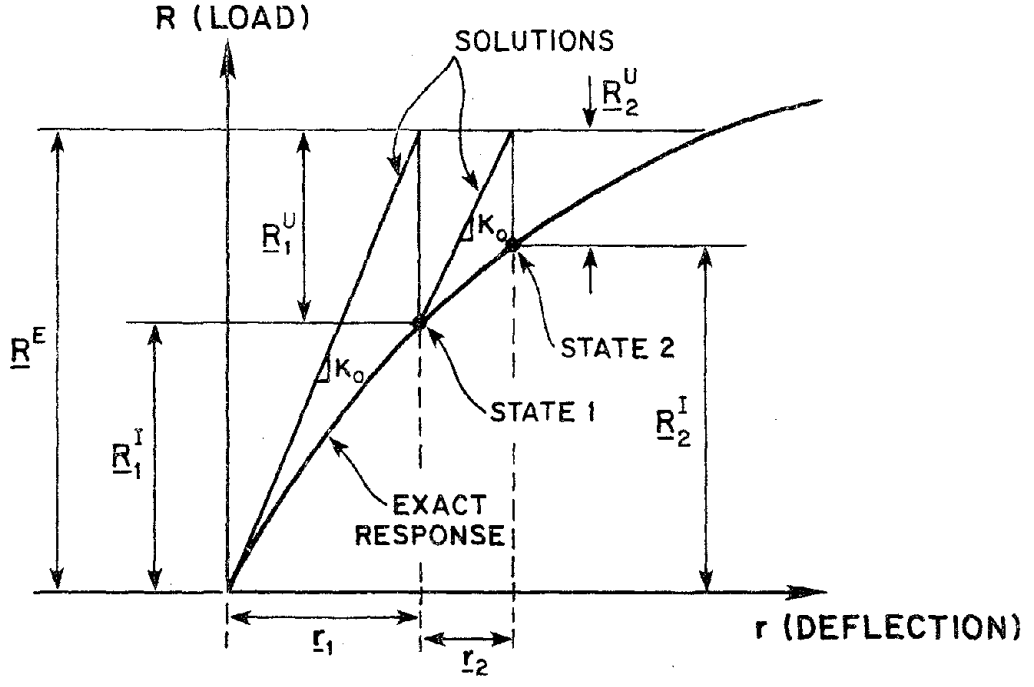


FIG. 2.2 CONSTANT-STIFFNESS ITERATION

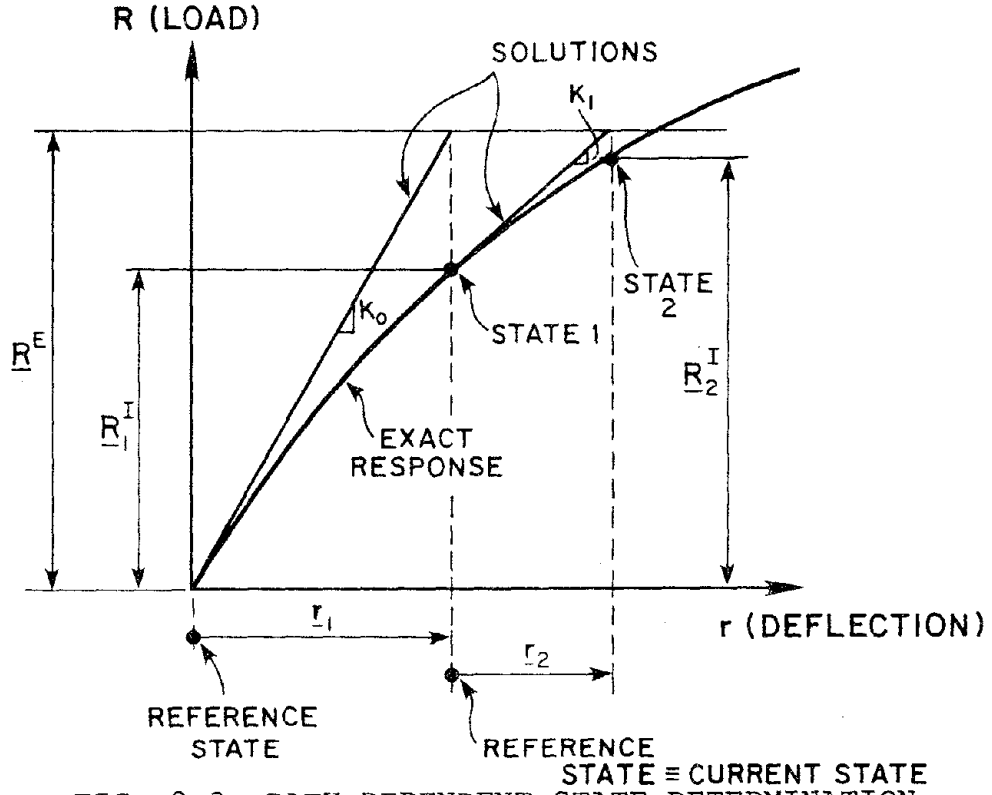


FIG. 2.3 PATH-DEPENDENT STATE DETERMINATION

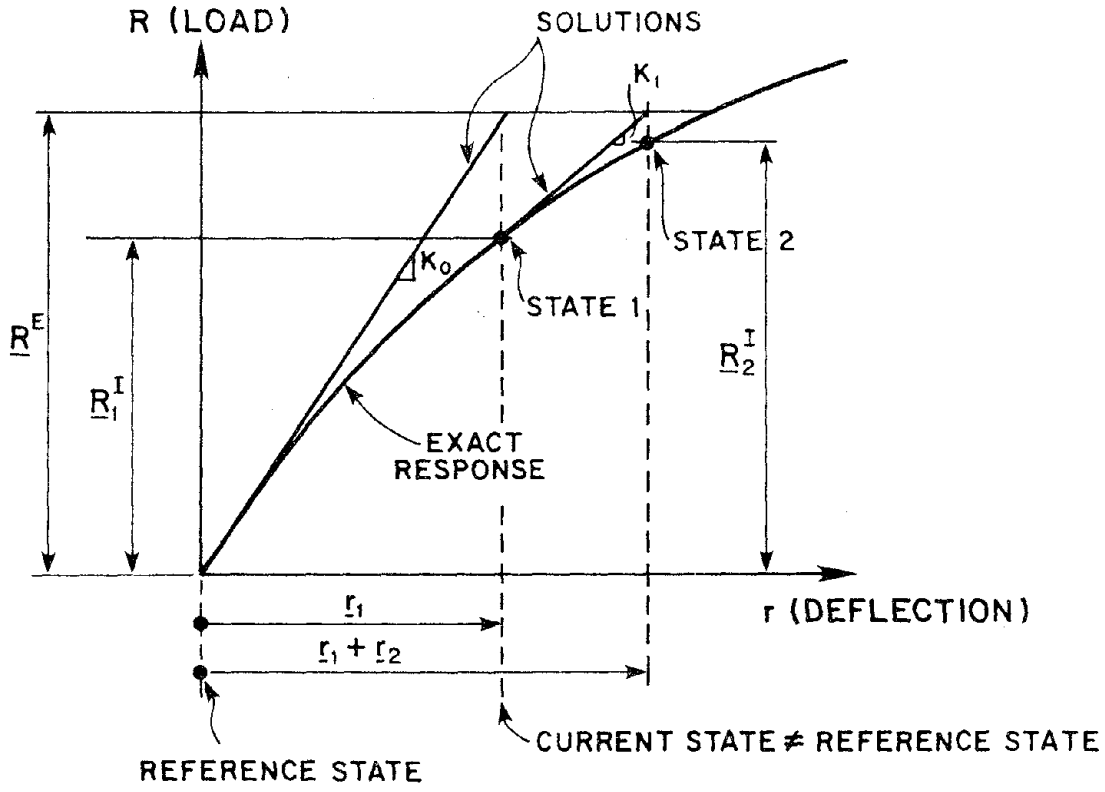


FIG. 2.4 PATH-INDEPENDENT STATE DETERMINATION

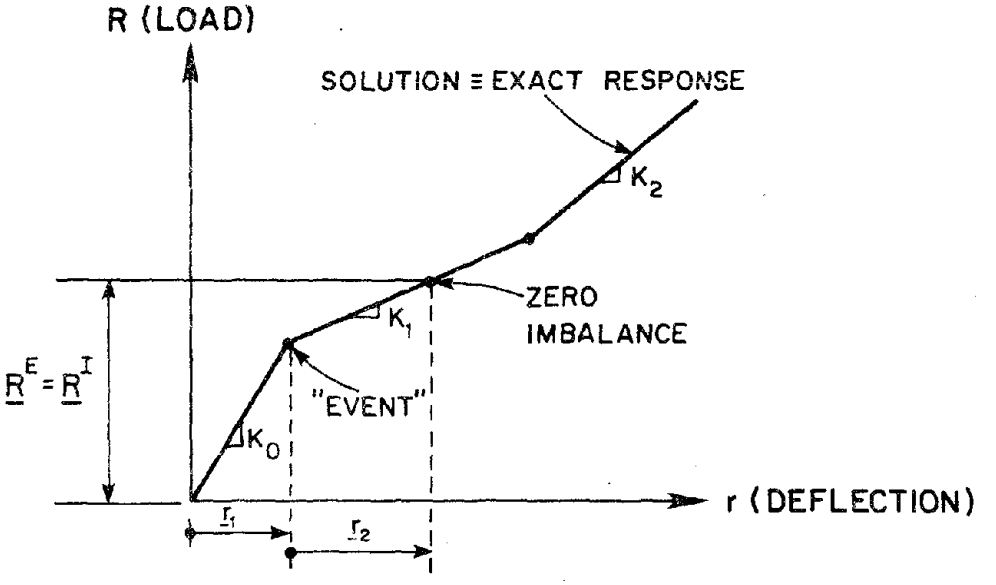


FIG. 2.5 EVENT-TO-EVENT STRATEGY FOR PIECE-WISE LINEAR RESPONSE

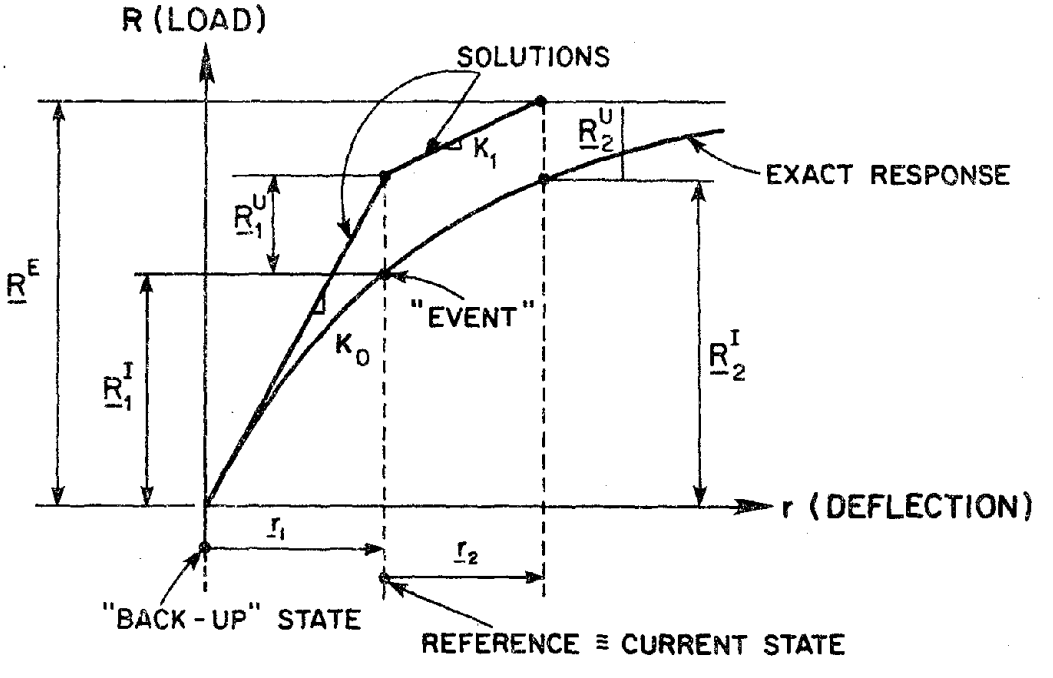


FIG. 2.6 EVENT-TO-EVENT STRATEGY FOR NONLINEAR RESPONSE

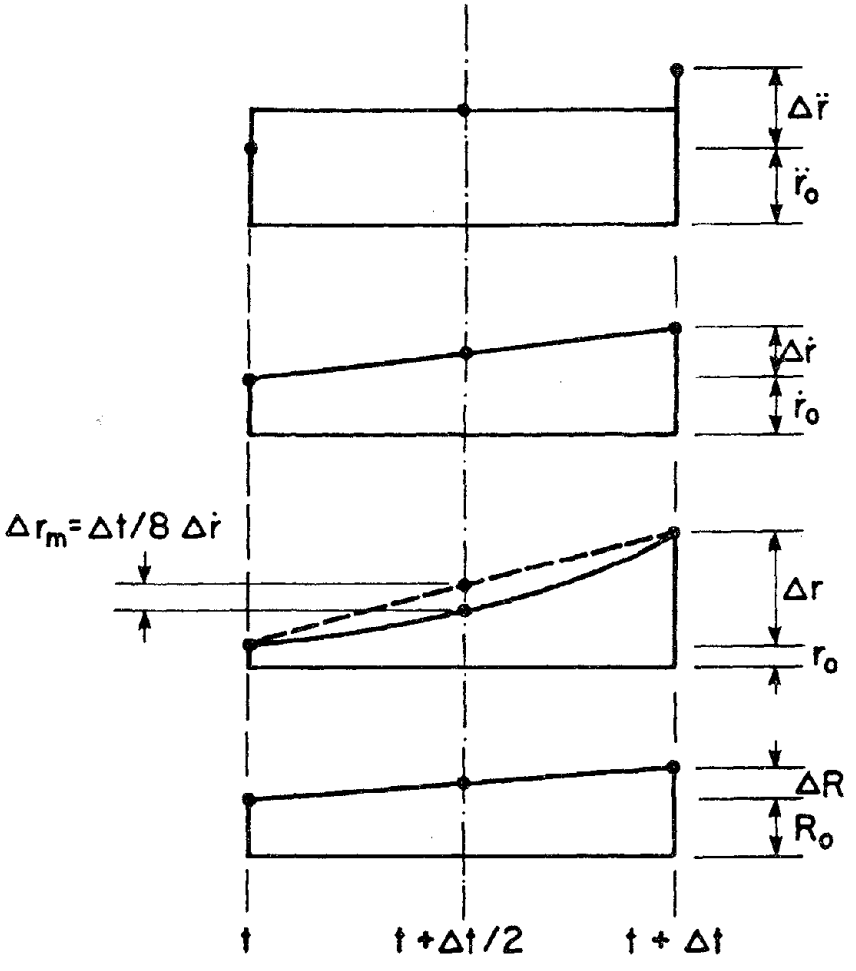


FIG. 3.1 CONSTANT AVERAGE ACCELERATION SCHEME

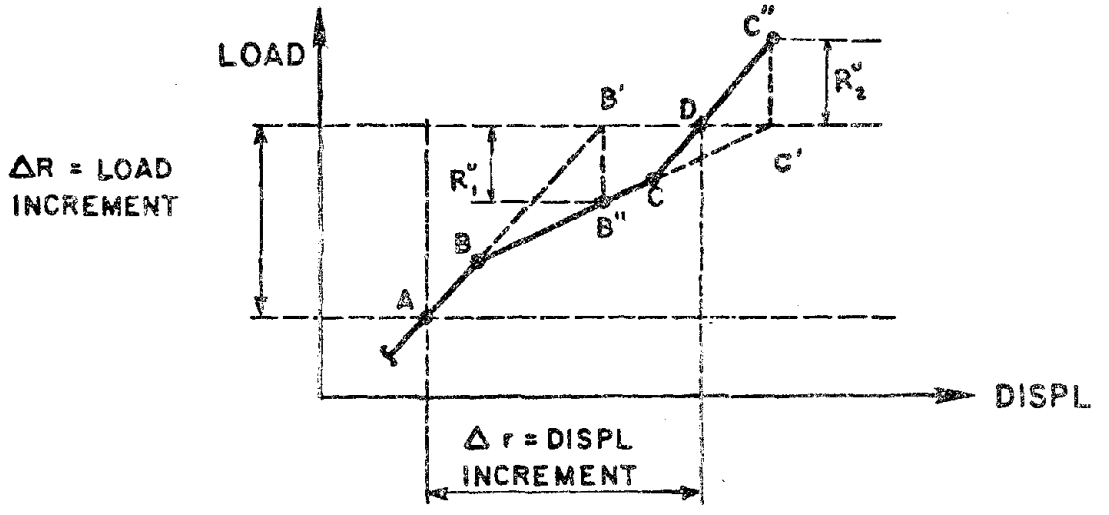


FIG. 3.2 ITERATION AND EVENT-TO-EVENT SOLUTIONS

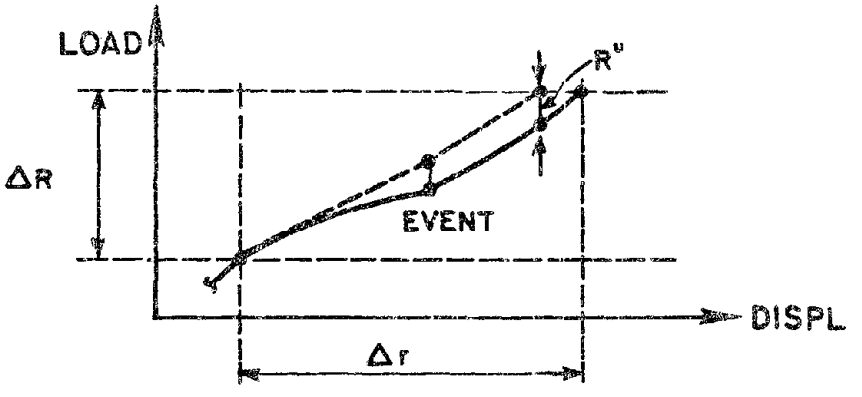


FIG. 3.3 UNBALANCE IN EVENT-TO-EVENT SOLUTION

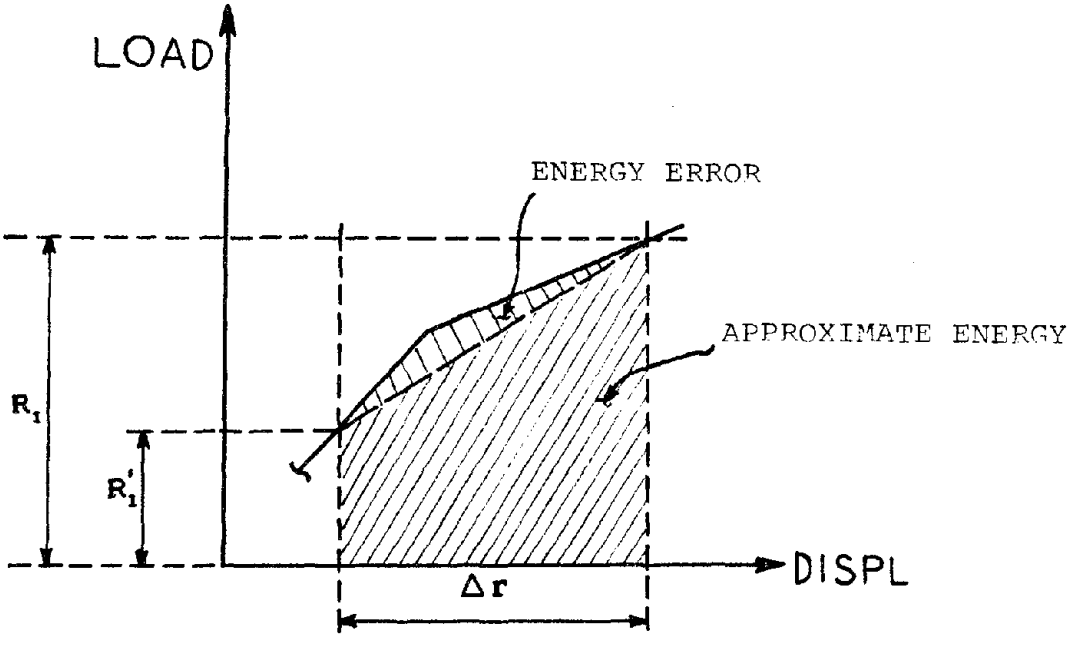


FIG. 3.4 EXACT AND APPROXIMATE INTERNAL ENERGIES

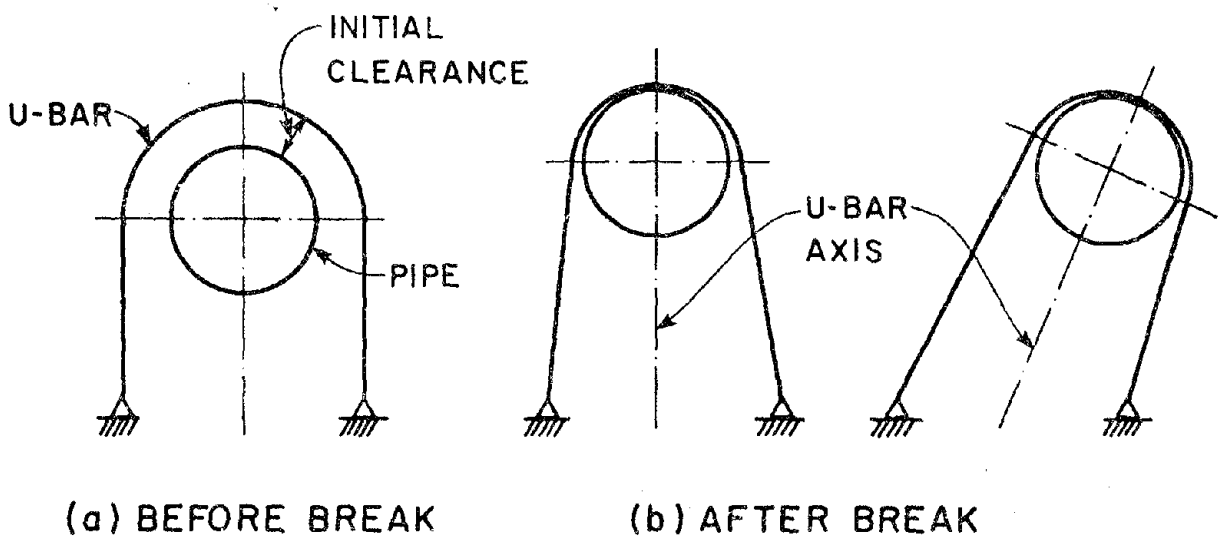


FIG. 5.1 U-BAR RESTRAINT (DIAGRAMMATIC)

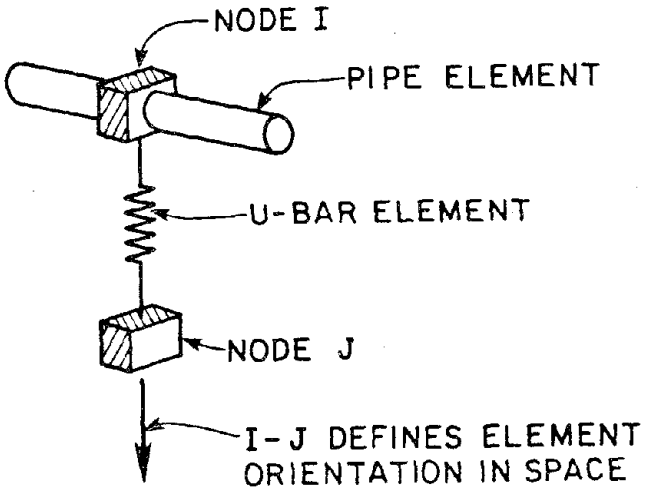


FIG. 5.2 IDEALIZATION FOR ANALYSIS

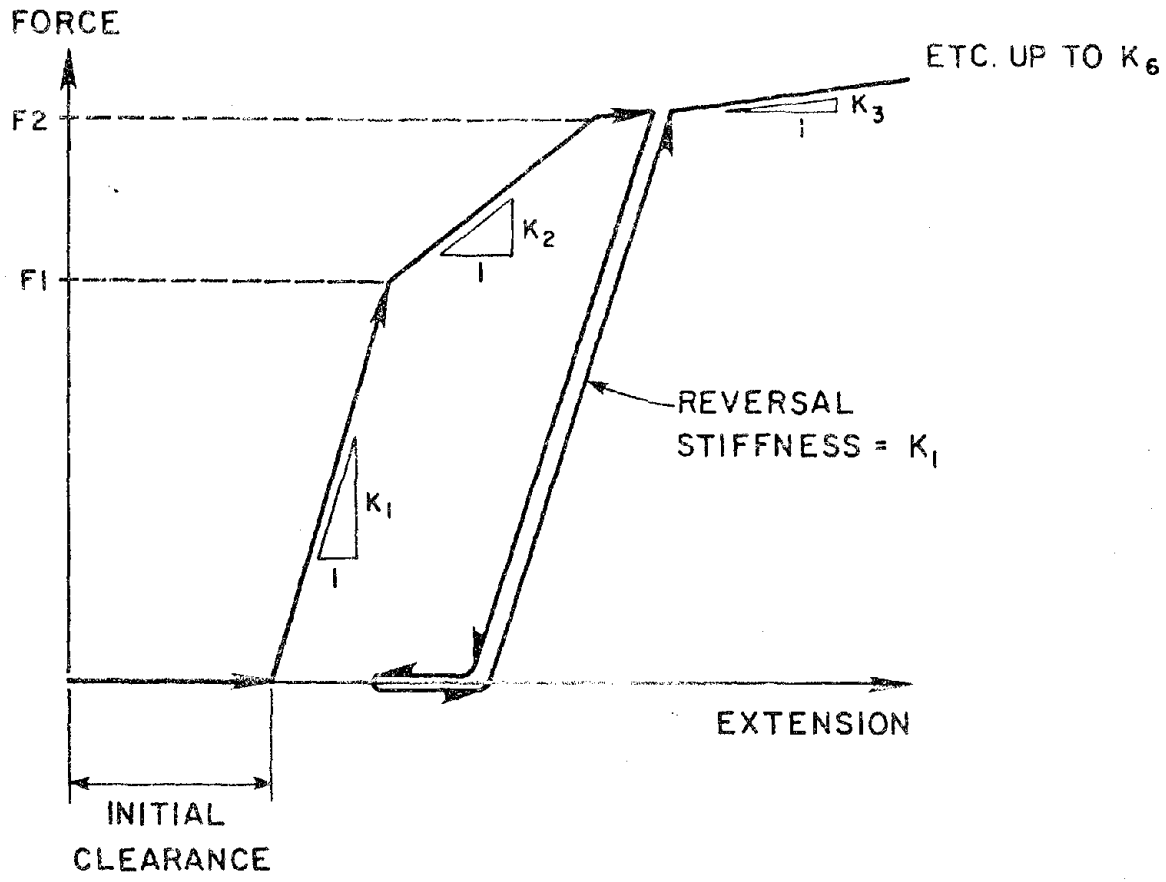
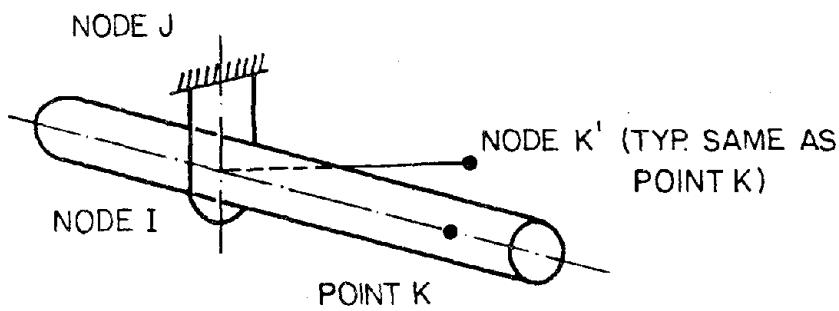


FIG. 5.3 STATIC FORCE-EXTENSION RELATIONSHIP



I J K' DEFINES NORMAL PLANE
I K PERPENDICULAR TO I J

FIG. 5.4 NORMAL DISPLACEMENT OPTION

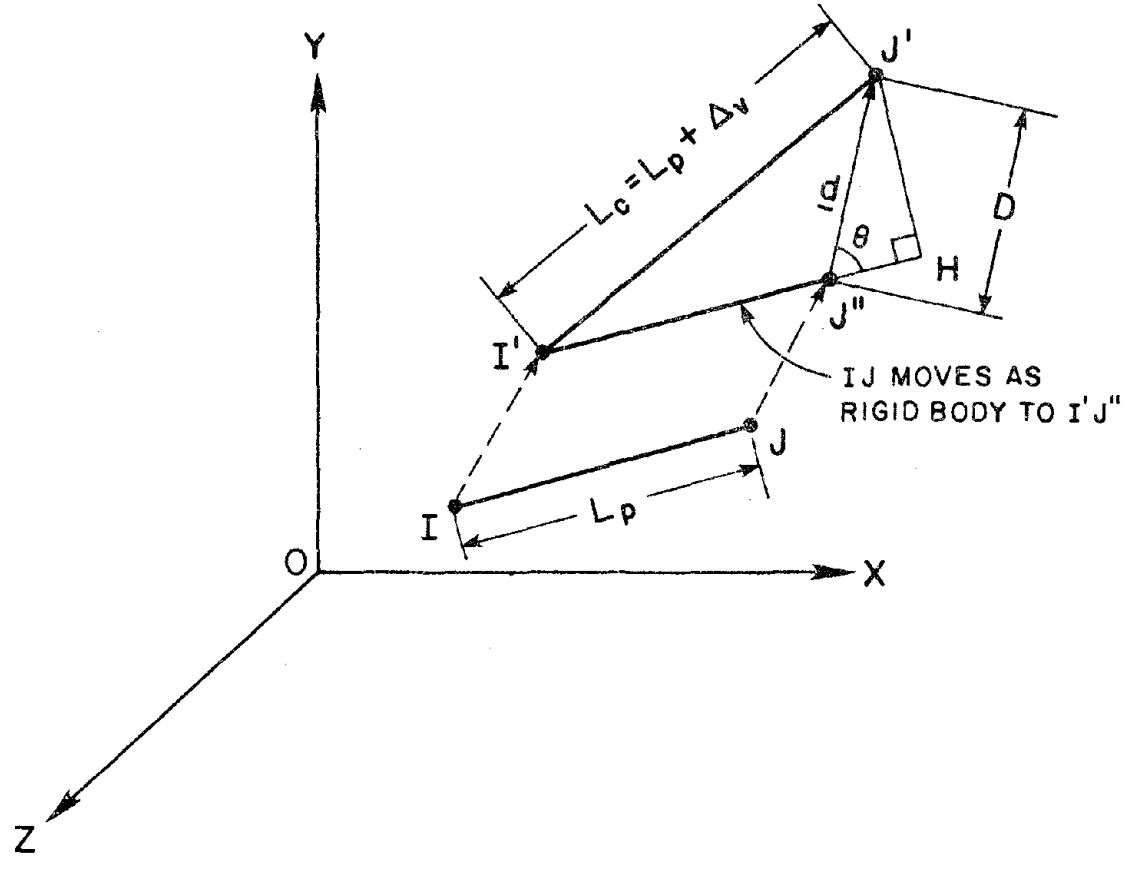


FIG. 5.5 RELATIONSHIP BETWEEN AXIAL DEFORMATION AND NODAL DISPLACEMENTS

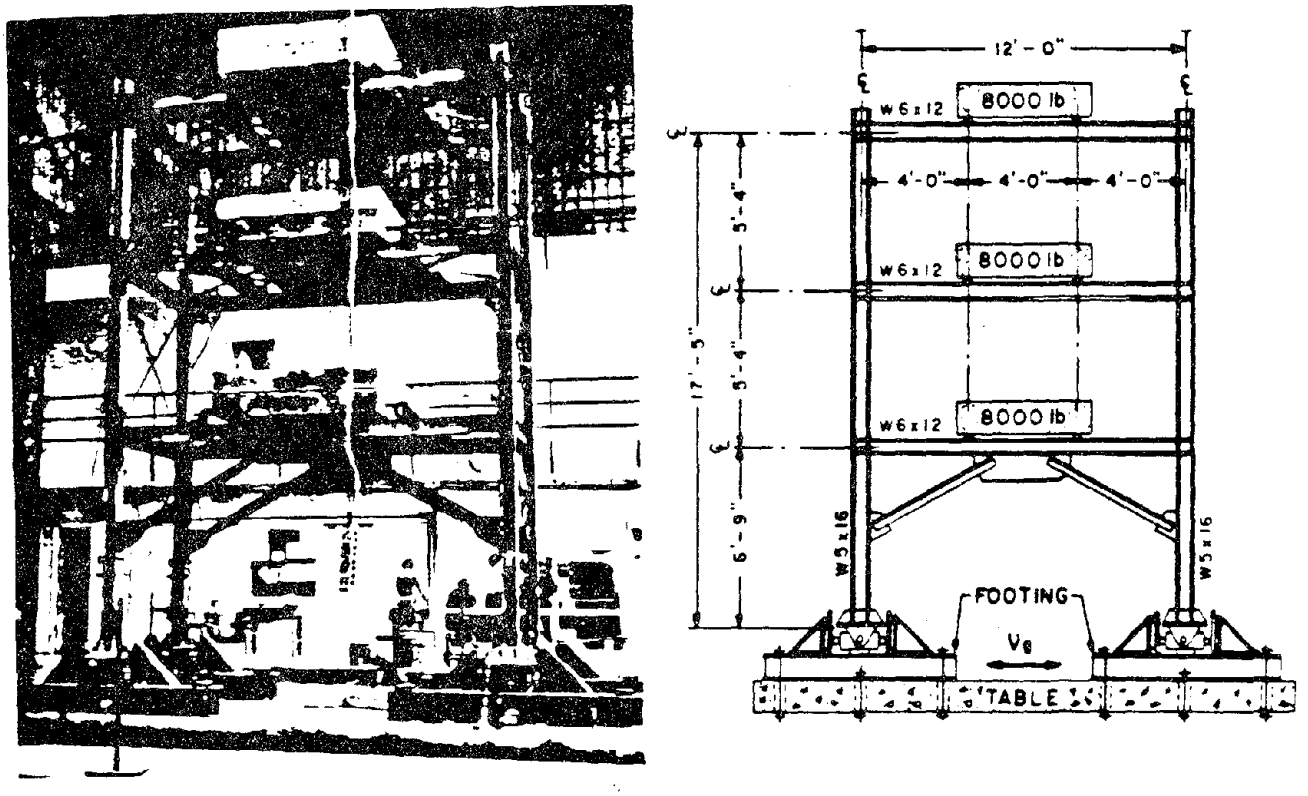


FIG.6.1 TEST MODEL (FROM HUCKELBRIDGE)

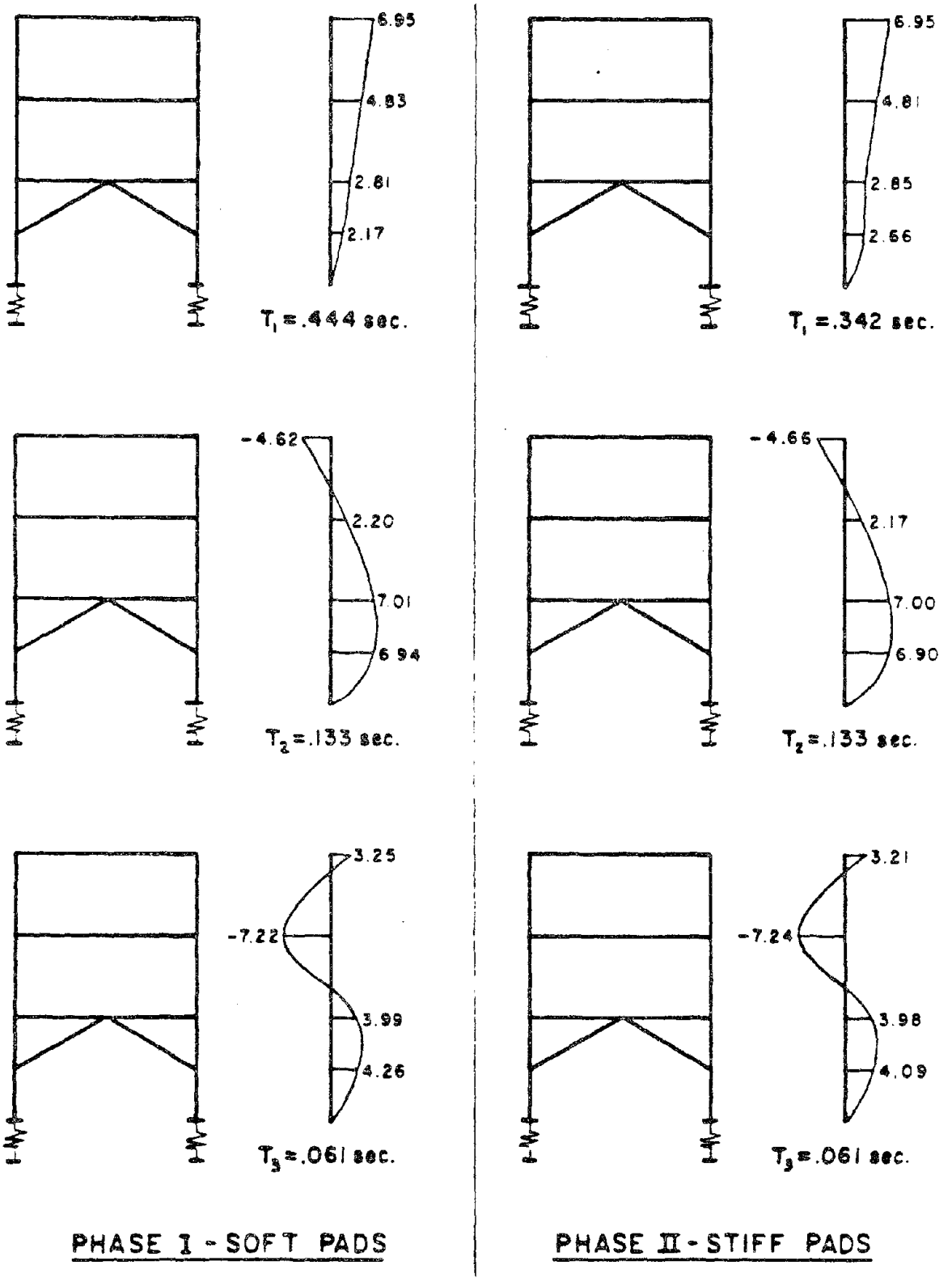


FIG. 6.2 CALCULATED FREQUENCIES AND MODAL SHAPES.

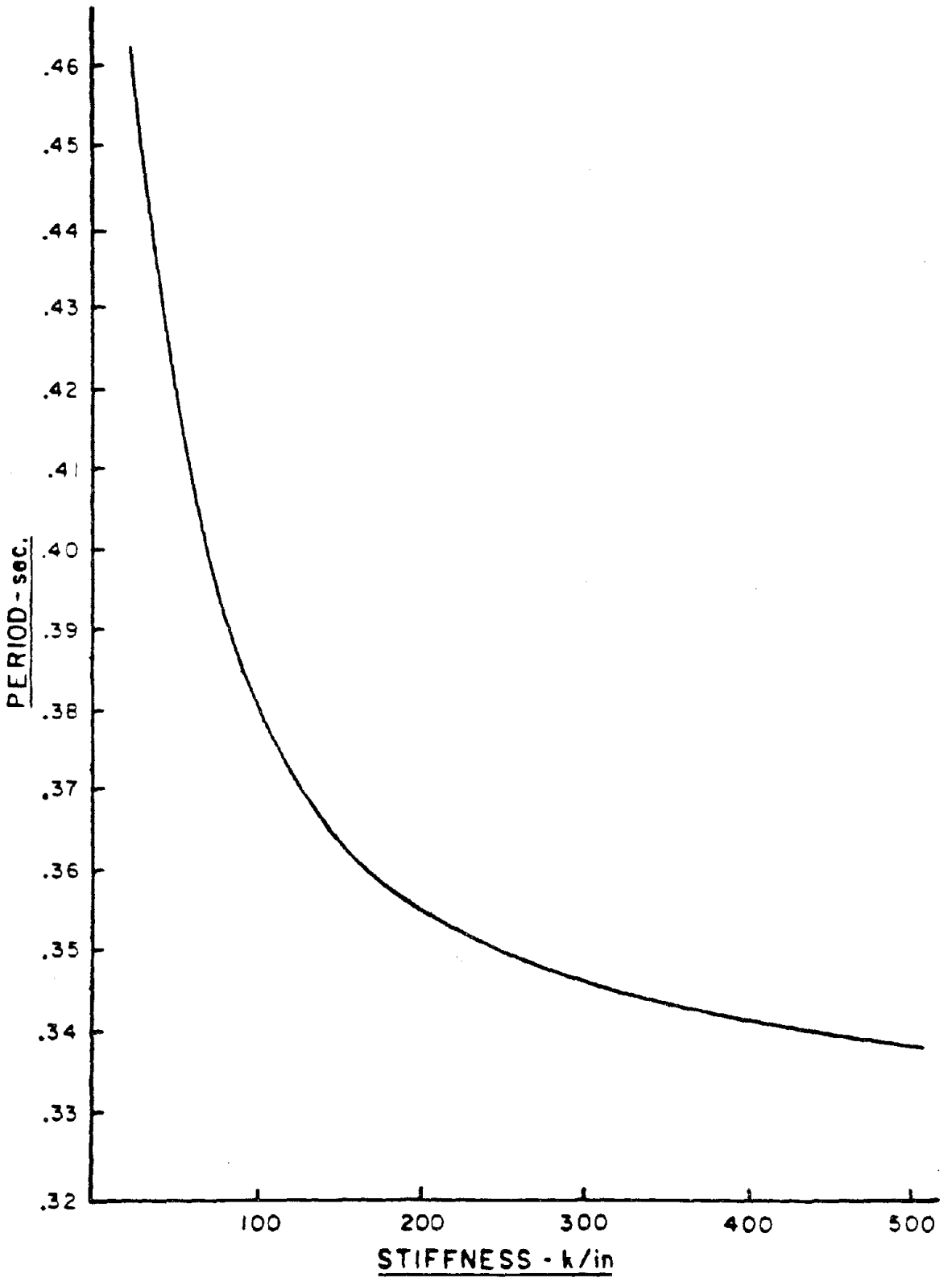


FIG. 6.3 CALCULATED 1st MODE PERIOD V/S IMPACT PAD STIFFNESS.

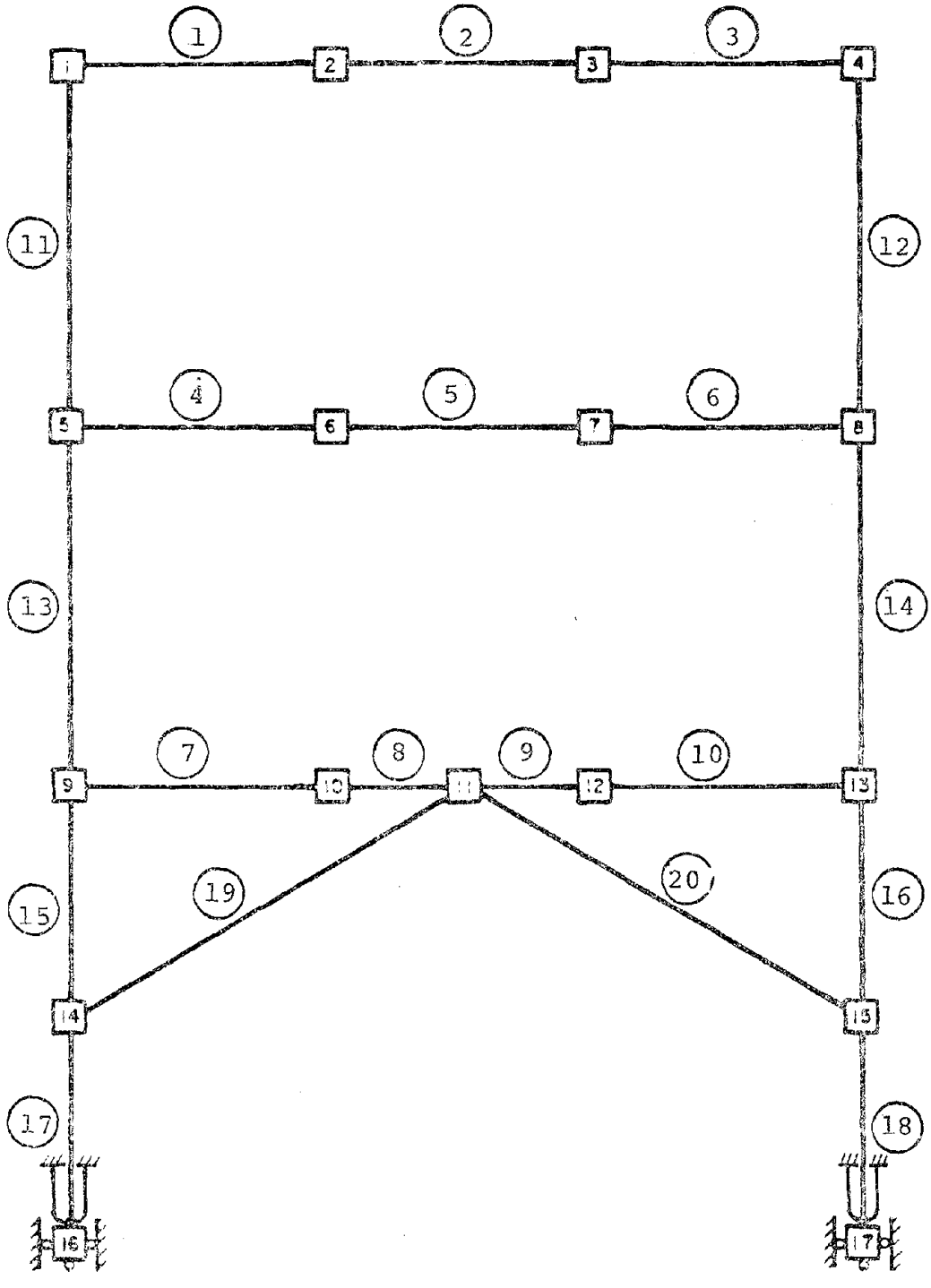


FIG. 6.4 ELEMENT AND NODE NUMBERING.

FIG. 6.5 EL CENTRO, 1940, N-S ACCELERATION RECORD.

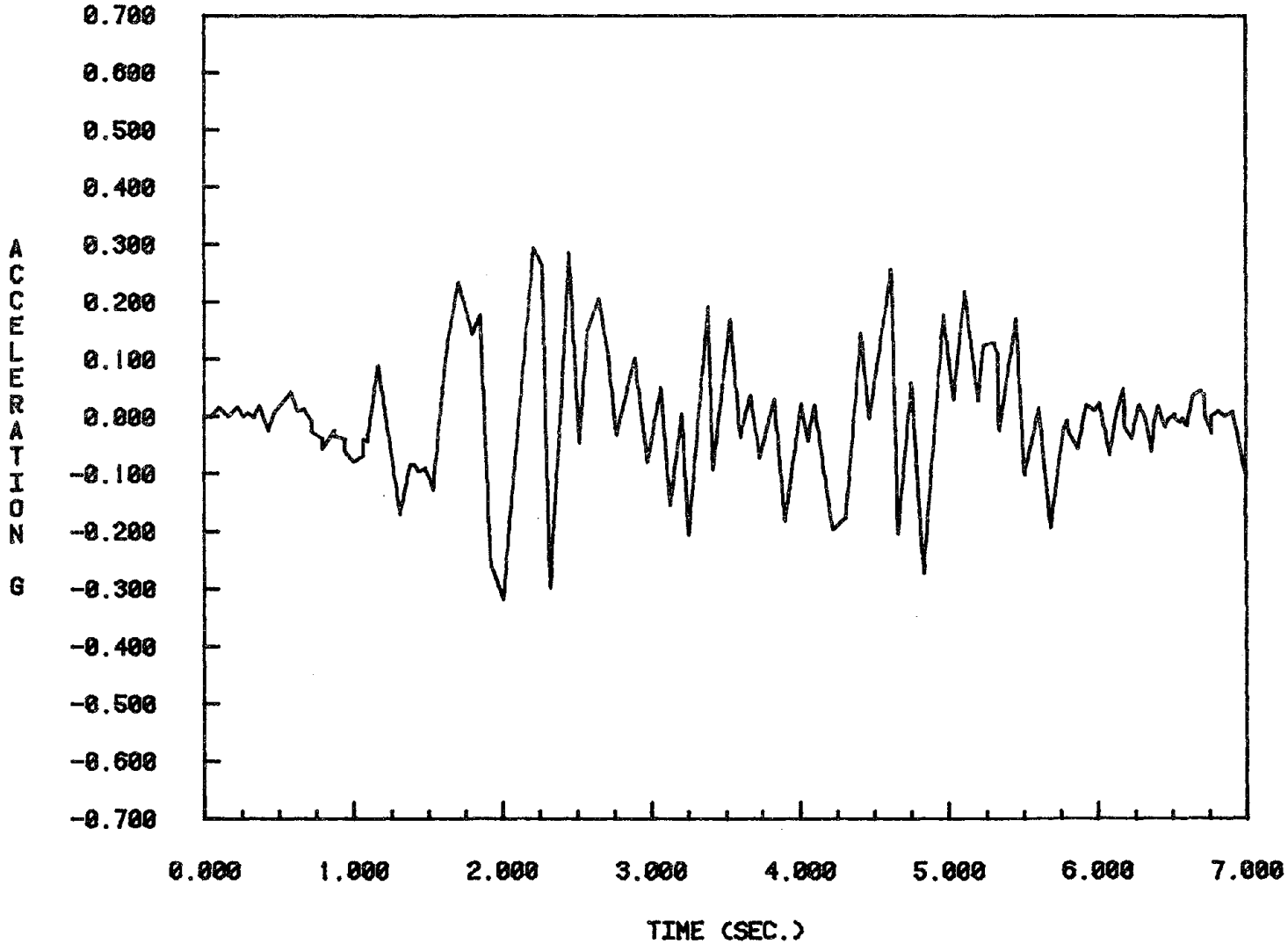


FIG. 6.6

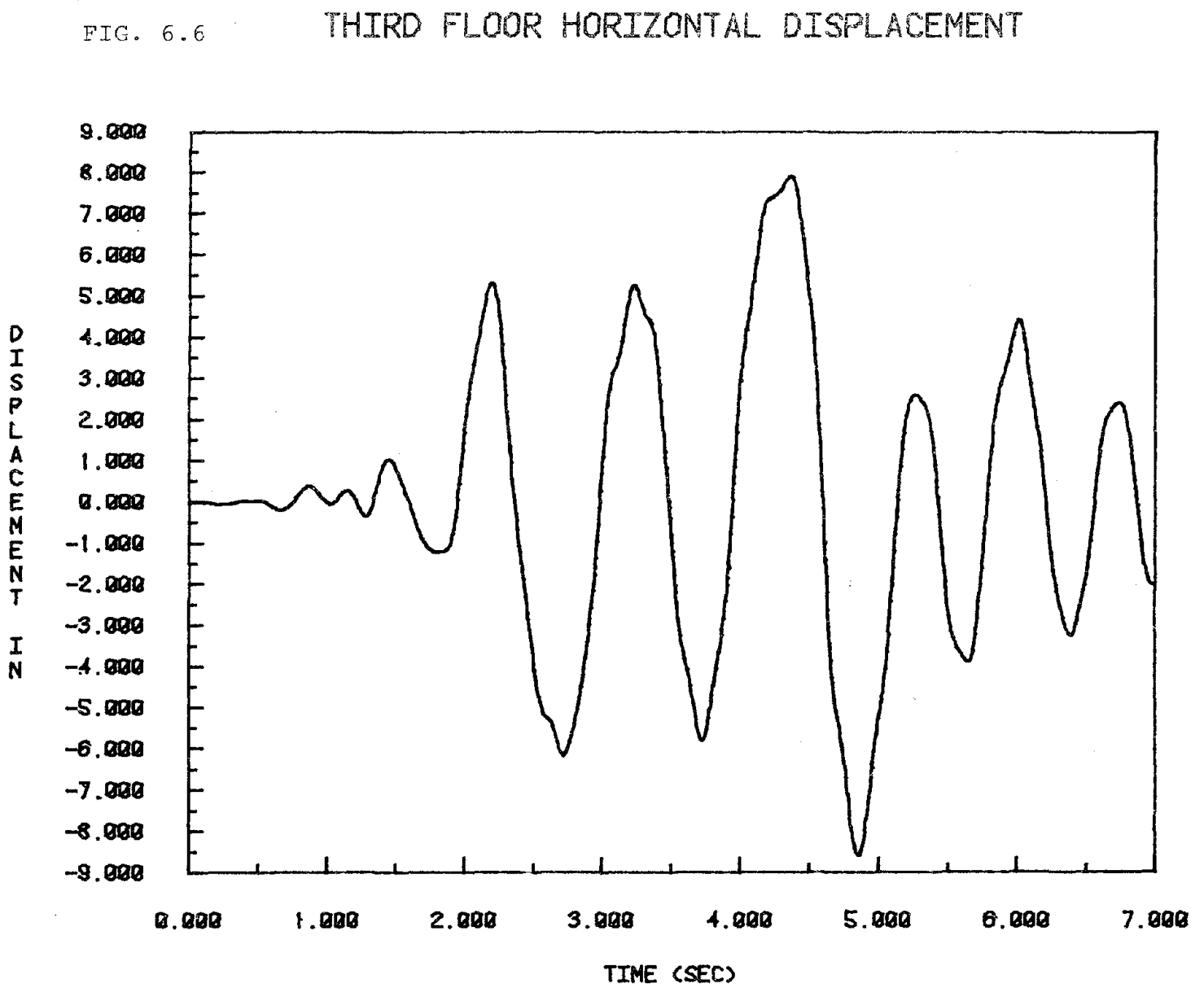

THIRD FLOOR HORIZONTAL DISPLACEMENT
DISPLACEMENT IN
9.000
8.000
7.000
6.000
5.000
4.000
3.000
2.000
1.000
0.000
-1.000
-2.000
-3.000
-4.000
-5.000
-6.000
-7.000
-8.000
-9.000
0.000
1.000
2.000
3.000
4.000
5.000
6.000
7.000
TIME (SEC)
CASE A1
CASE A2

FIG. 6.7 THIRD FLOOR HORIZONTAL DISPLACEMENT

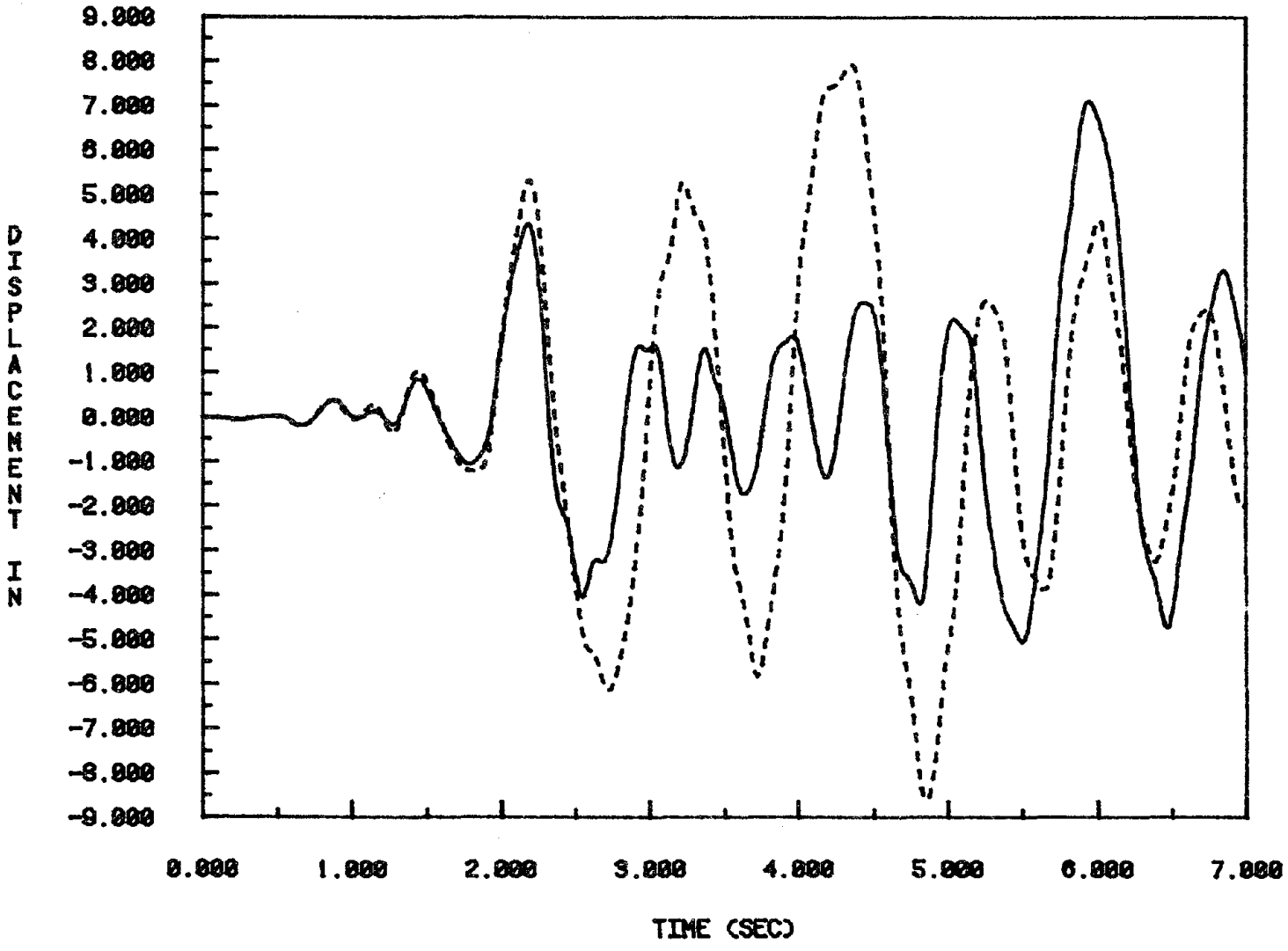


FIG. 6.8 THIRD FLOOR HORIZONTAL DISPLACEMENT

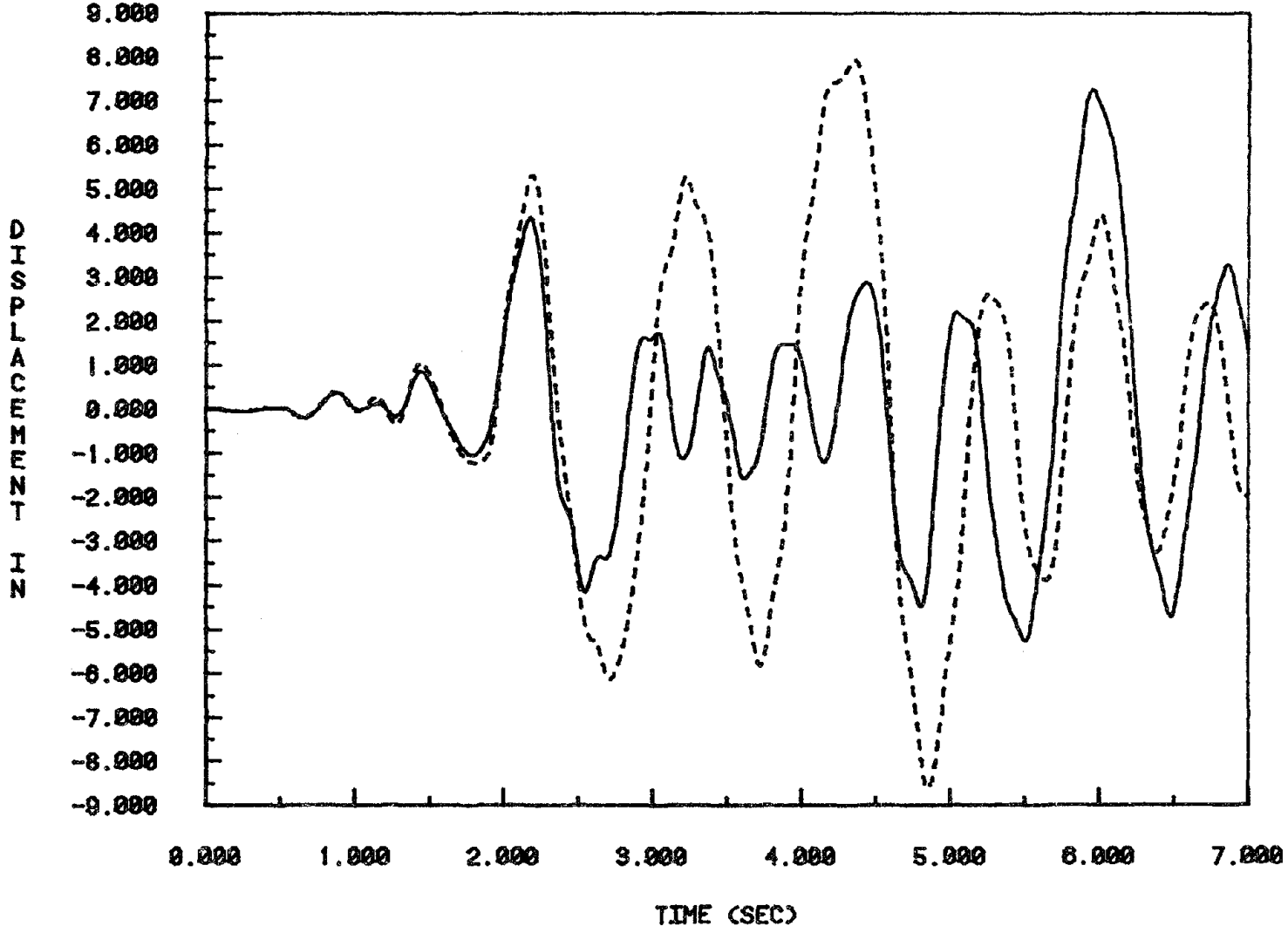


FIG. 6.9 THIRD FLOOR HORIZONTAL DISPLACEMENT

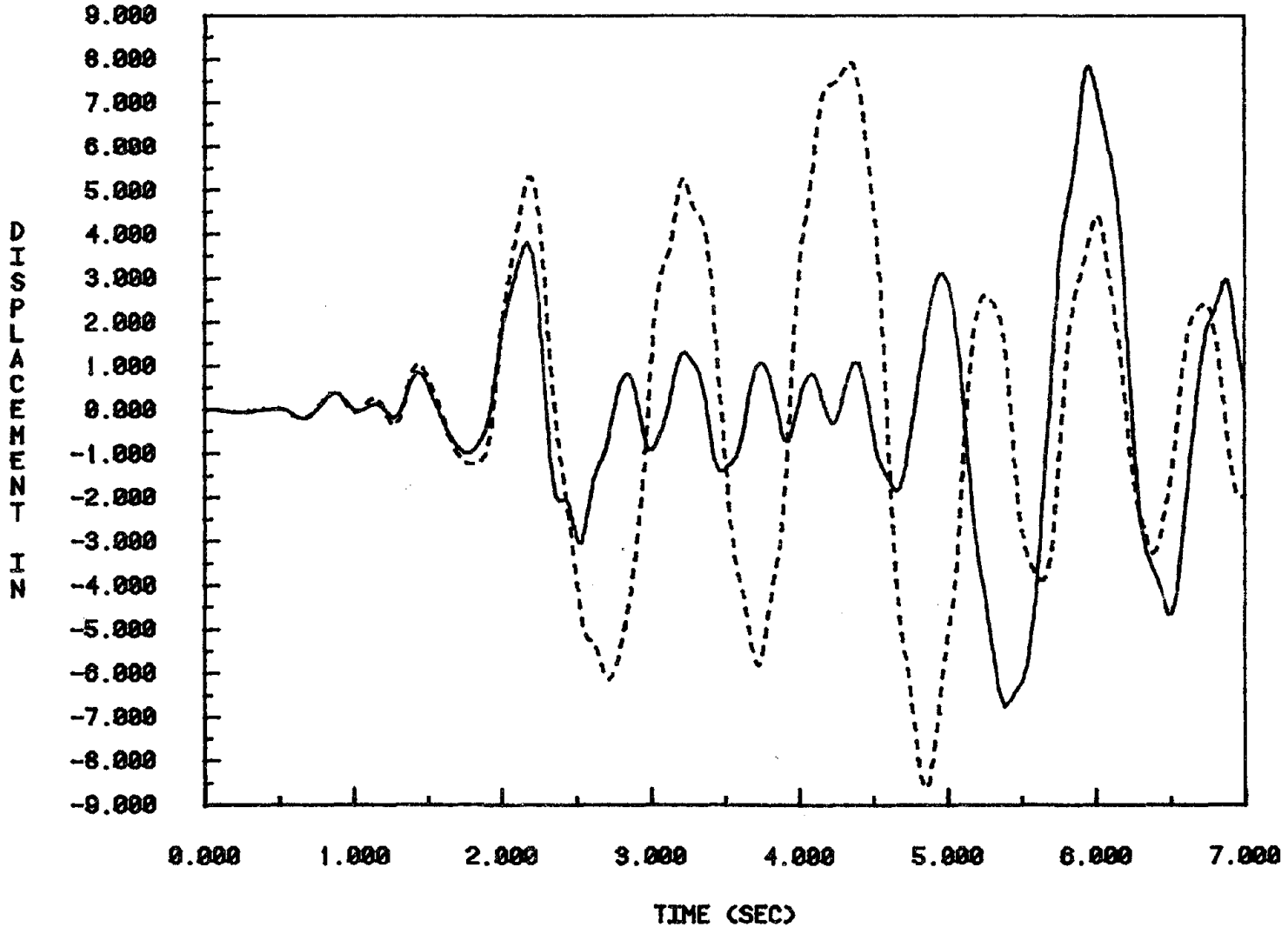


FIG. 6.10 THIRD FLOOR HORIZONTAL DISPLACEMENT

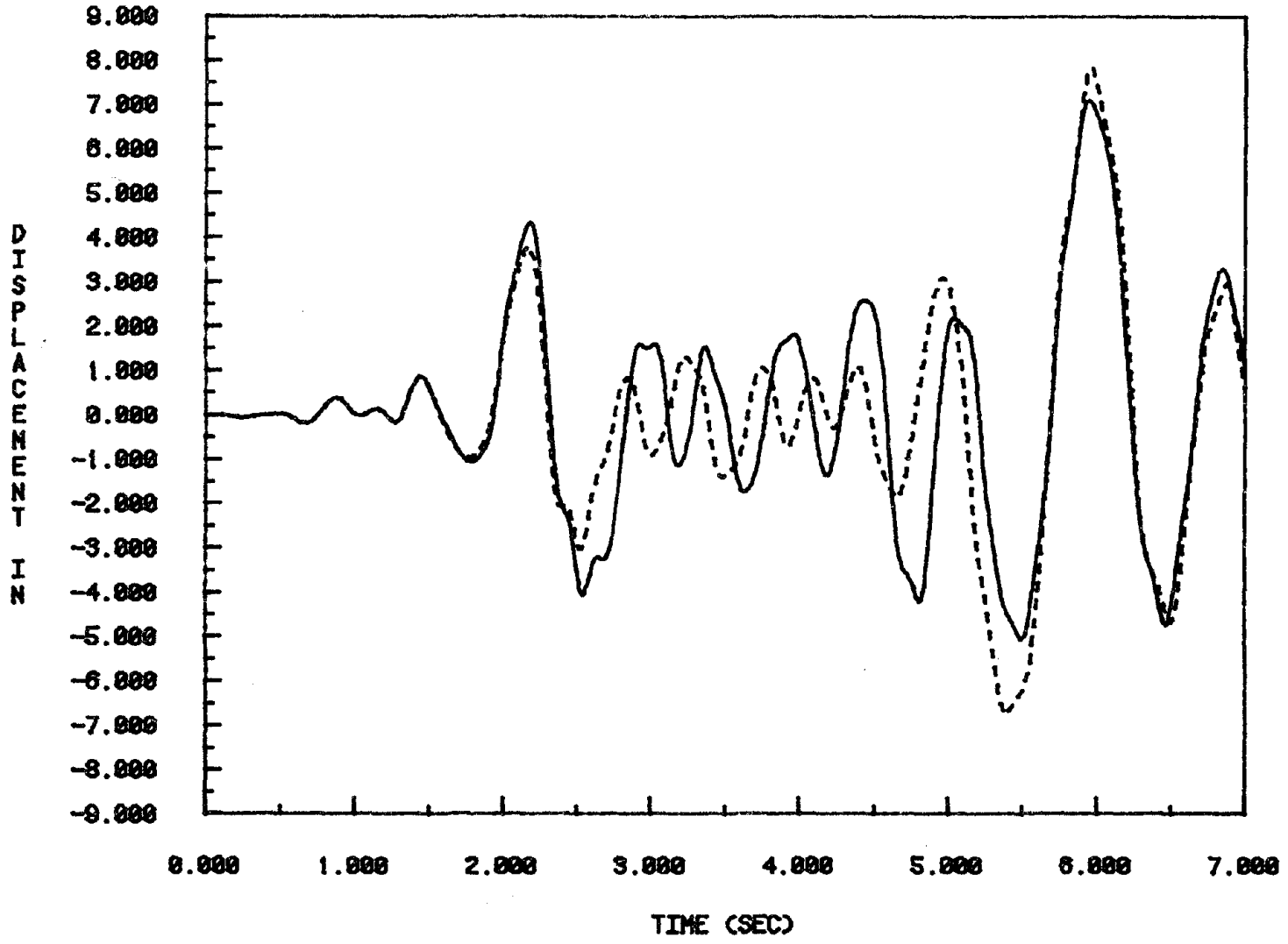


FIG. 6.11 THIRD FLOOR HORIZONTAL DISPLACEMENT

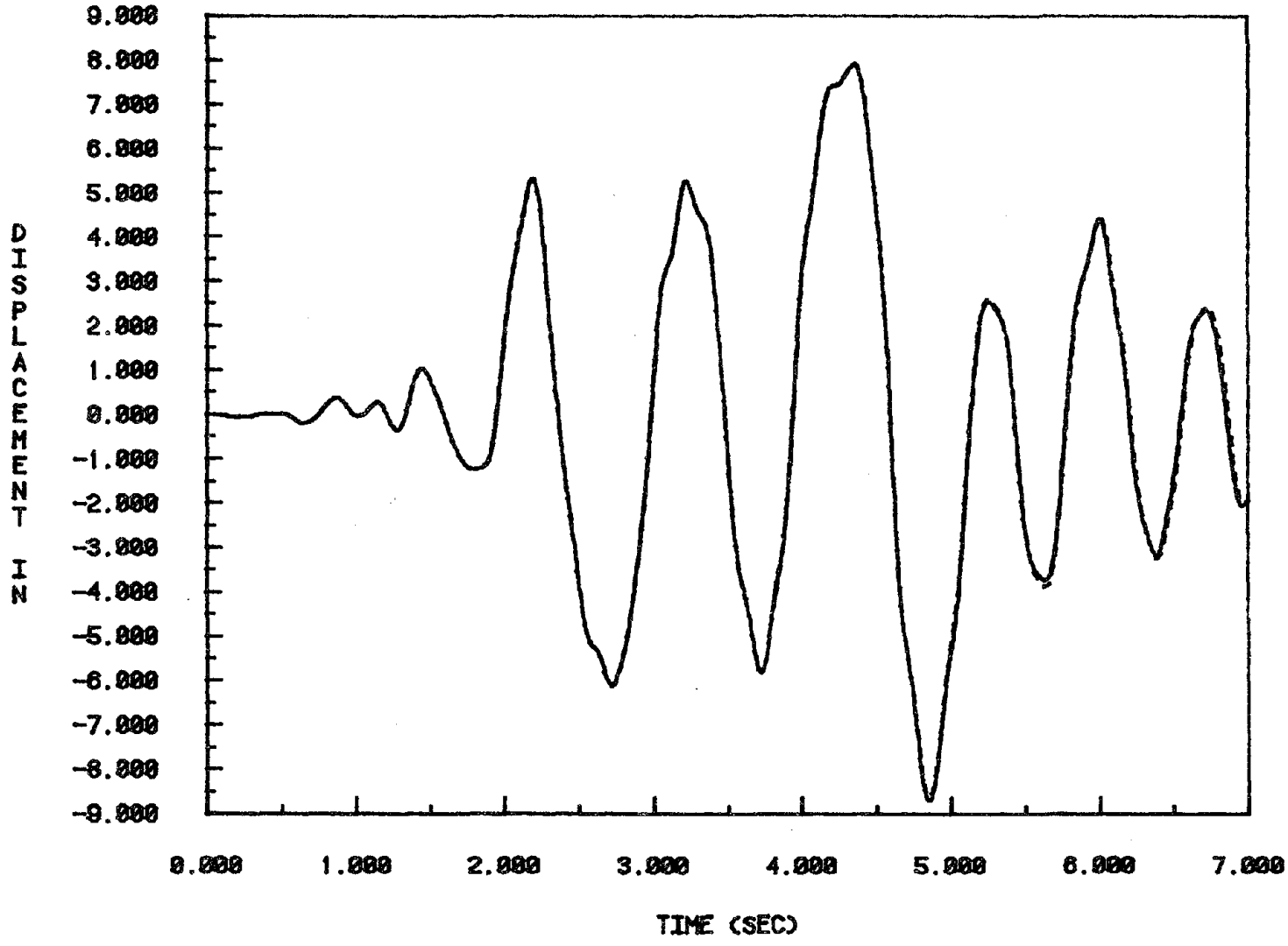


FIG. 6.12

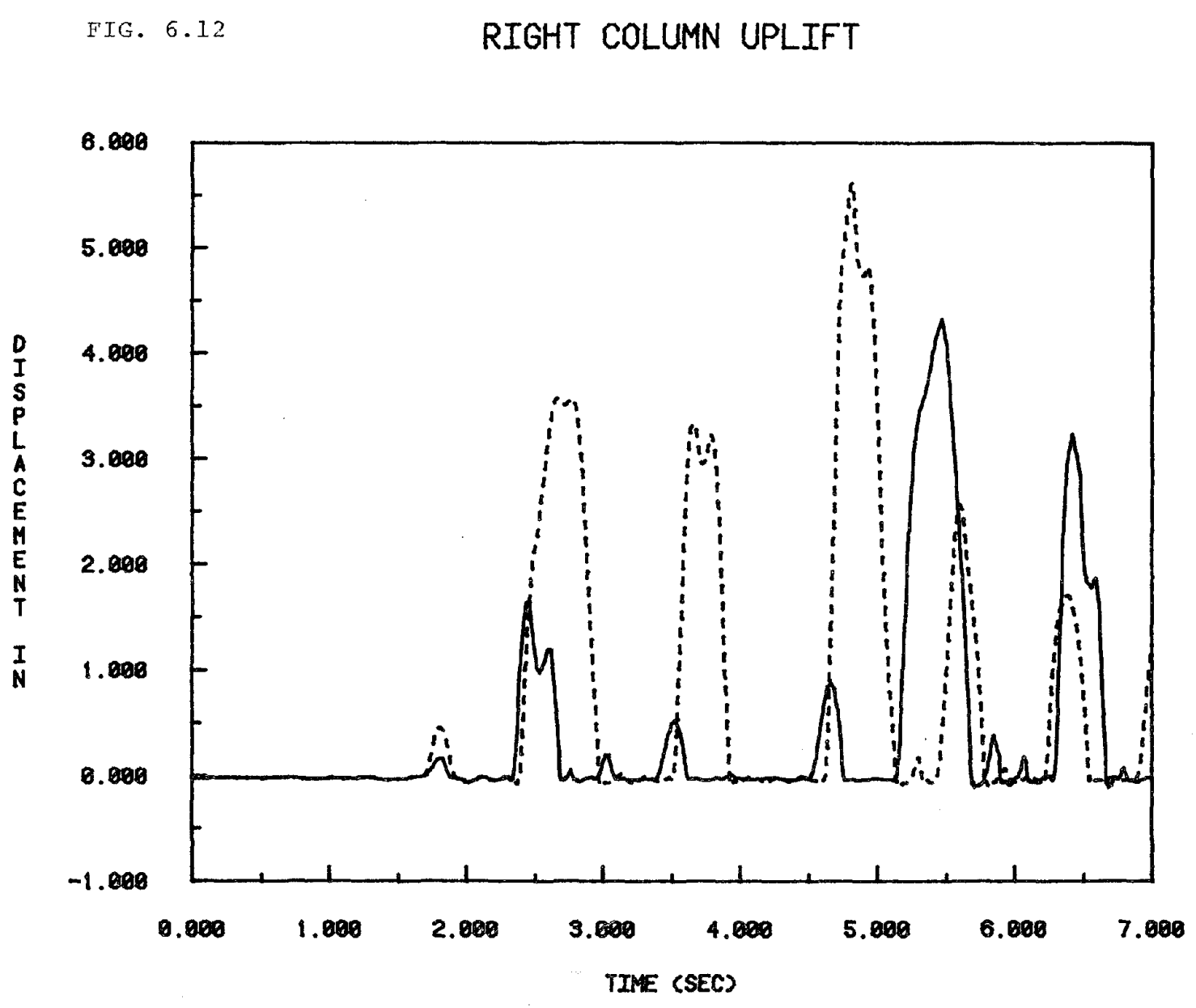

RIGHT COLUMN UPLIFT
6.000
5.000
4.000
3.000
2.000
1.000
0.000
-1.000
DISPLACEMENT IN
0.000
1.000
2.000
3.000
4.000
5.000
6.000
7.000
TIME (SEC)
CASE B3
CASE A1

FIG. 6.13

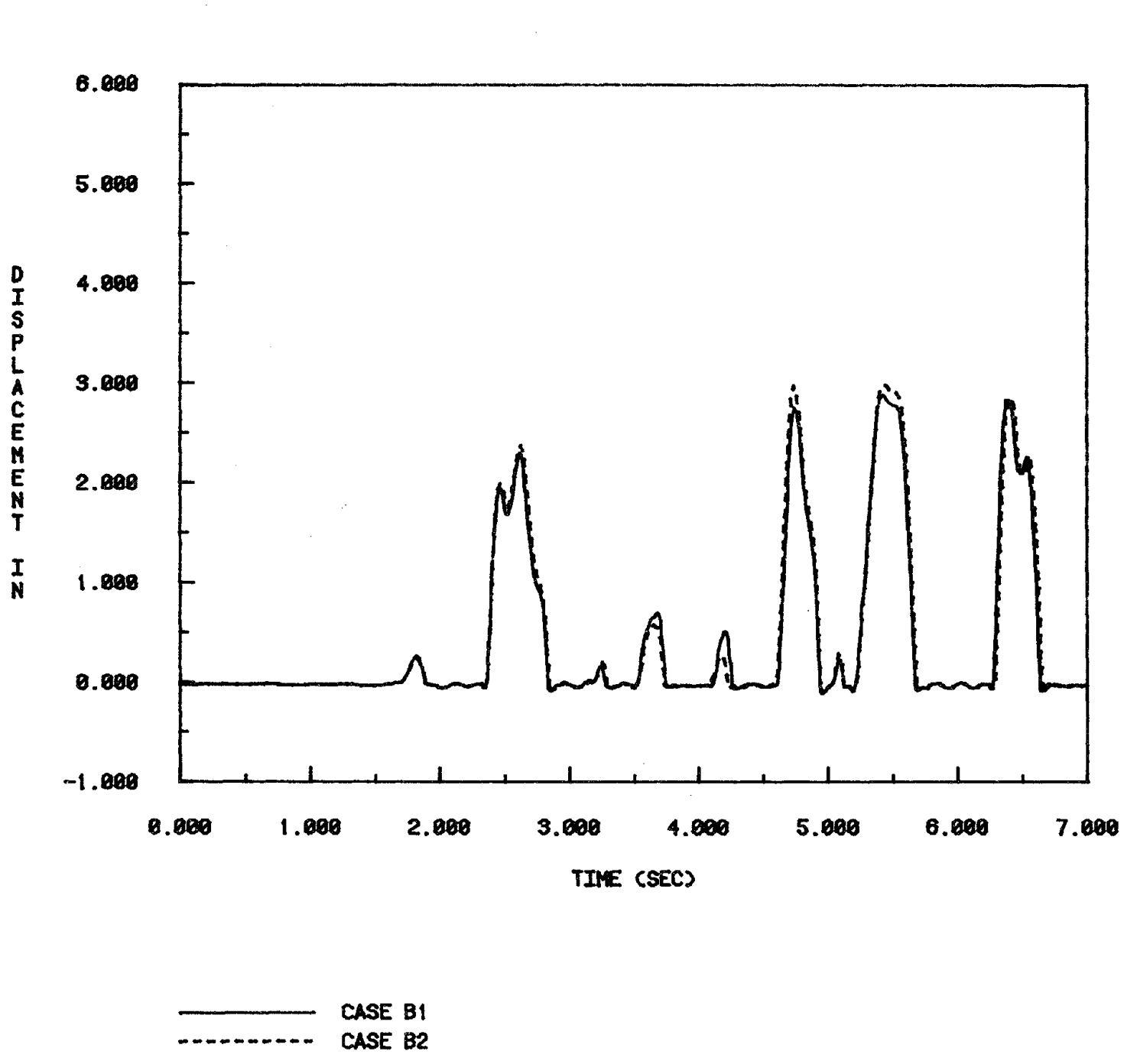

RIGHT COLUMN UPLIFT
6.000
5.000
4.000
3.000
2.000
1.000
0.000
-1.000
DISPLACEMENT IN
0.000
1.000
2.000
3.000
4.000
5.000
6.000
7.000
TIME (SEC)
CASE B1
CASE B2

FIG. 6.14

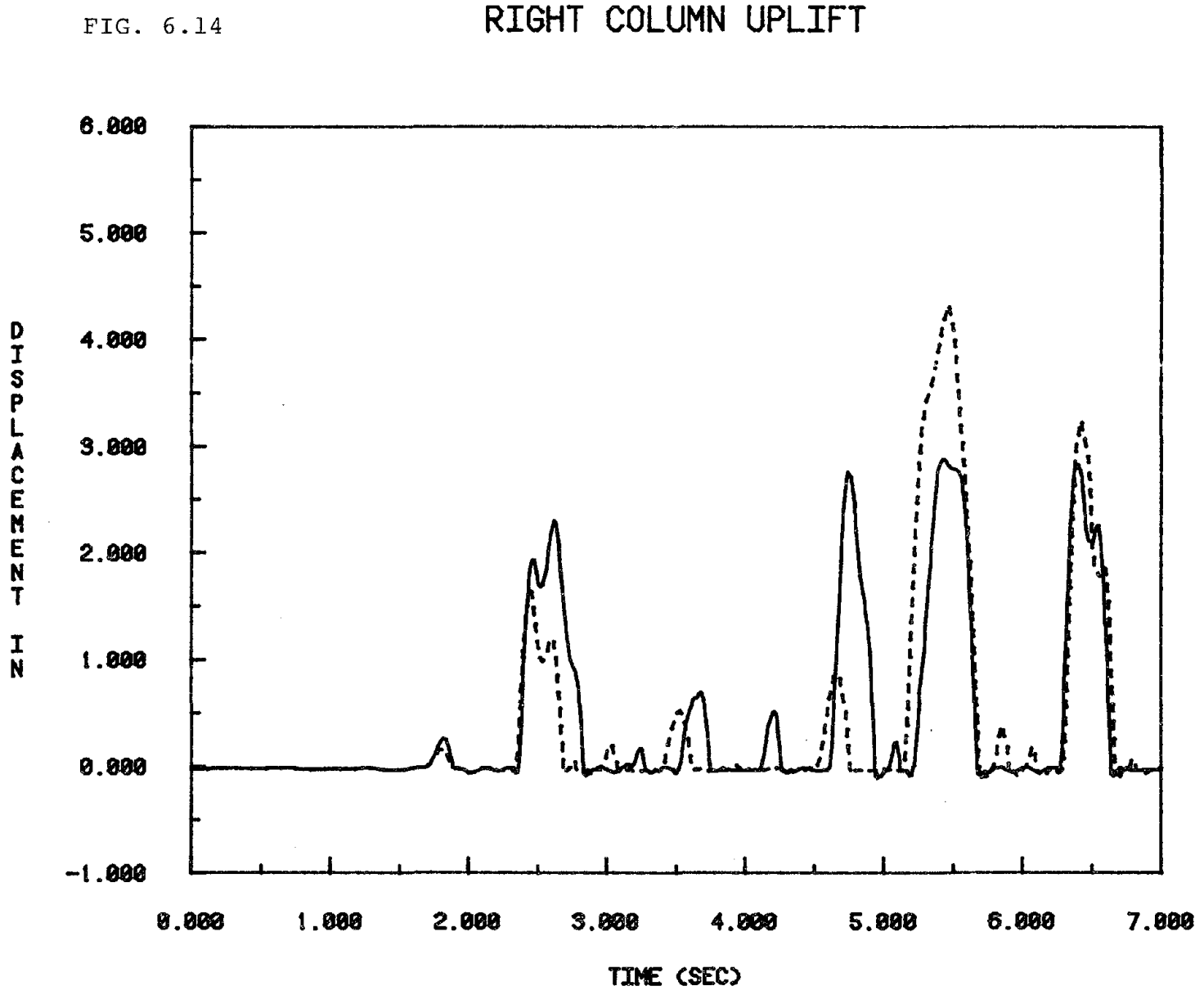

RIGHT COLUMN UPLIFT
6.000
5.000
4.000
3.000
2.000
1.000
0.000
-1.000
DISPLACEMENT IN
0.000
1.000
2.000
3.000
4.000
5.000
6.000
7.000
TIME (SEC)
CASE B1
CASE B3

FIG. 6.15

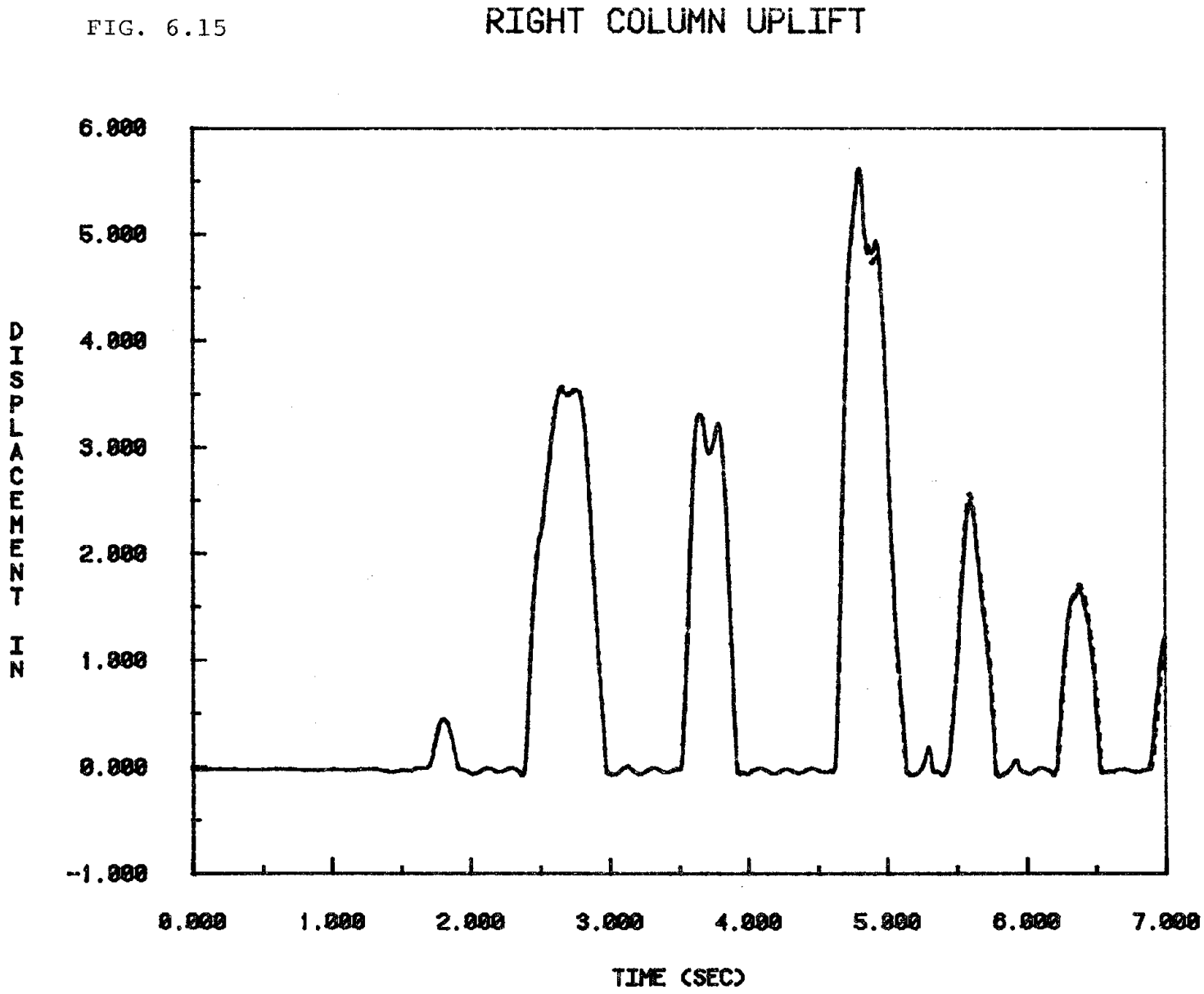
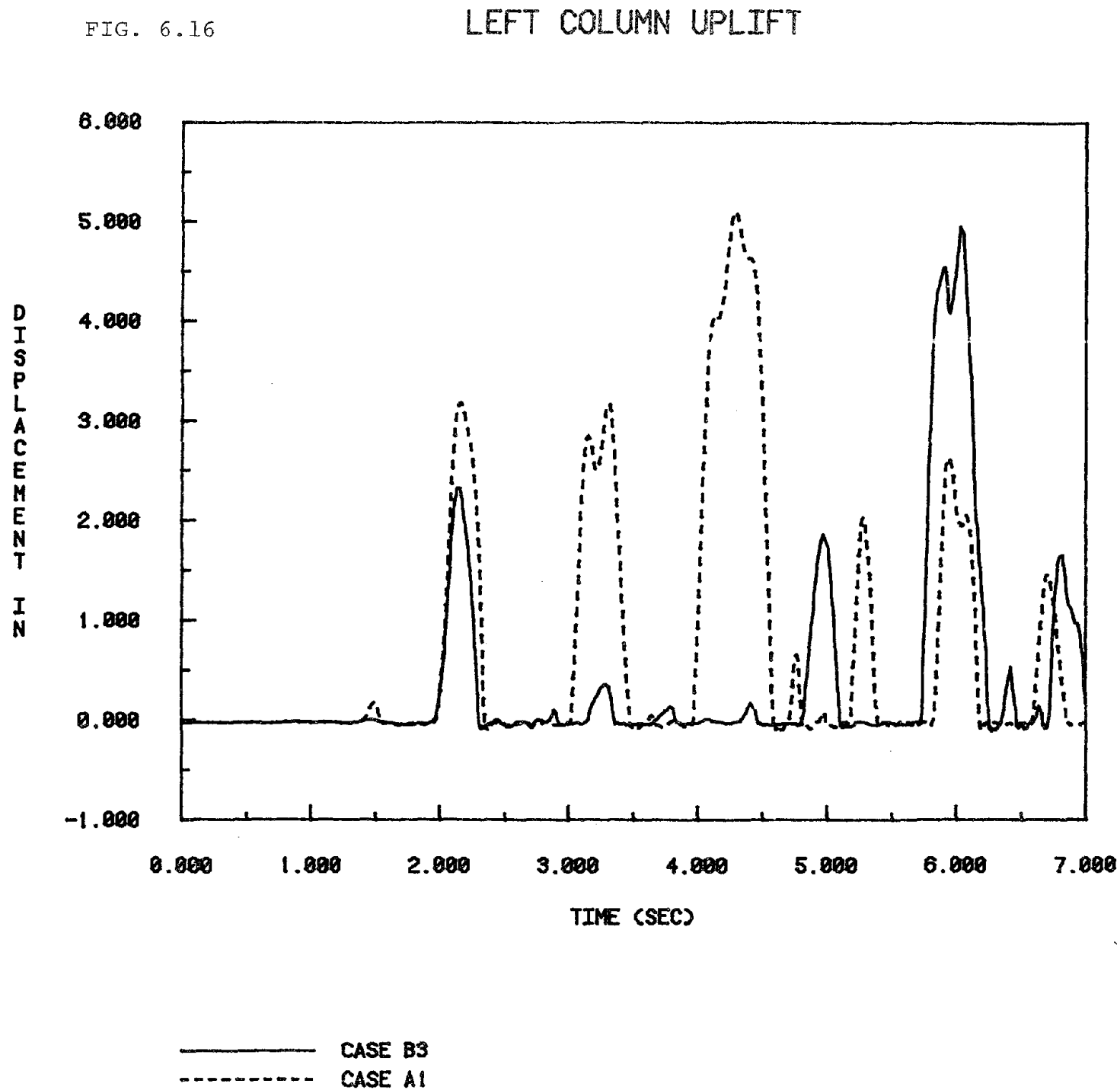

RIGHT COLUMN UPLIFT
6.000
5.000
4.000
3.000
2.000
1.000
0.000
-1.000
DISPLACEMENT IN
0.000
1.000
2.000
3.000
4.000
5.000
6.000
7.000
TIME (SEC)
CASE C1
CASE A1

FIG. 6.16


LEFT COLUMN UPLIFT
6.000
5.000
4.000
3.000
2.000
1.000
0.000
-1.000
DISPLACEMENT IN
0.000
1.000
2.000
3.000
4.000
5.000
6.000
7.000
TIME (SEC)
CASE B3
CASE A1

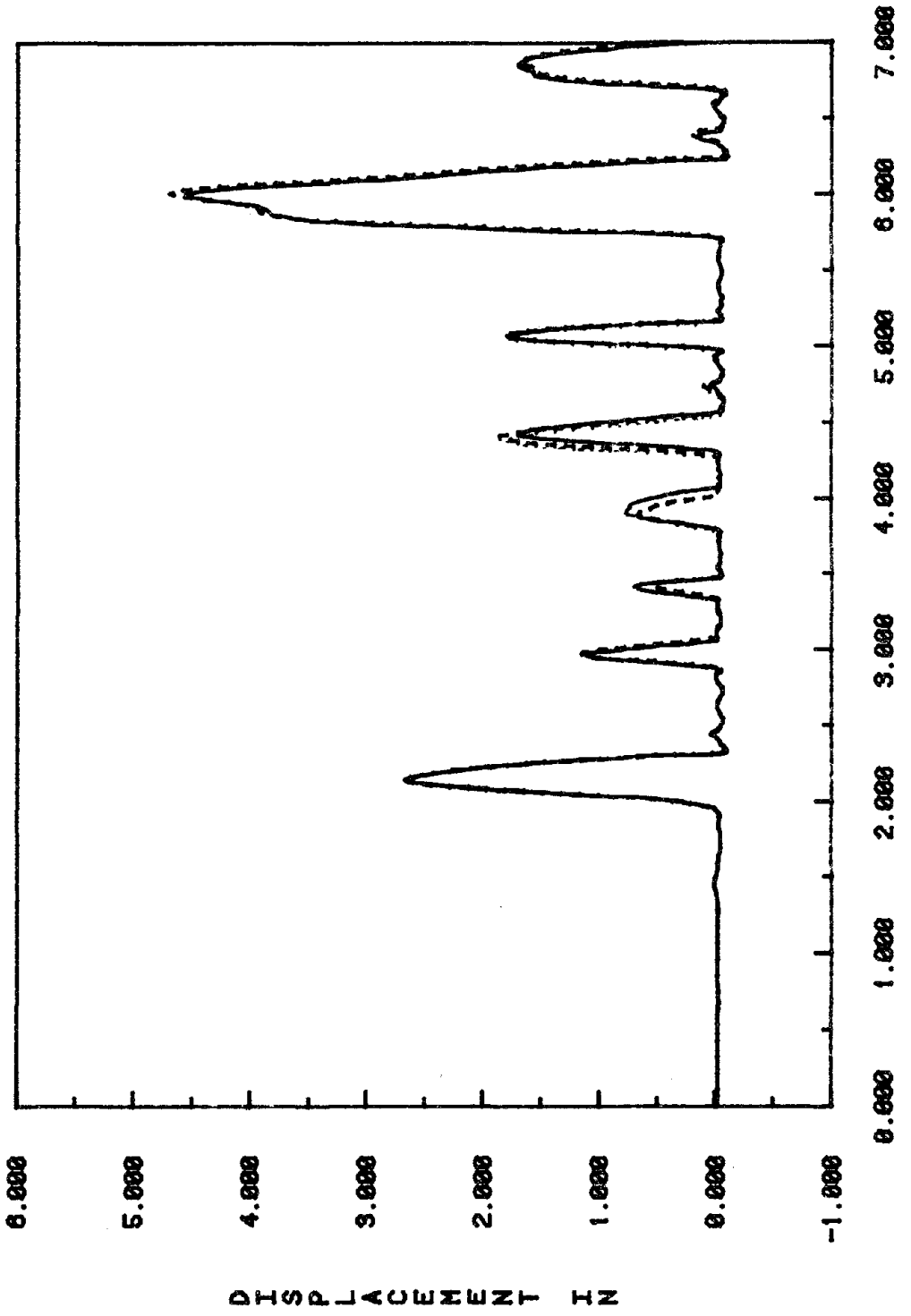

LEFT COLUMN UPLIFT
DISPLACEMENT IN
6.0000
5.0000
4.0000
3.0000
2.0000
1.0000
0.0000
-1.0000
0.0000
1.0000
2.0000
3.0000
4.0000
5.0000
6.0000
7.0000
TIME (SEC)
CASE B1
CASE B2

FIG. 6.17

FIG. 6.18

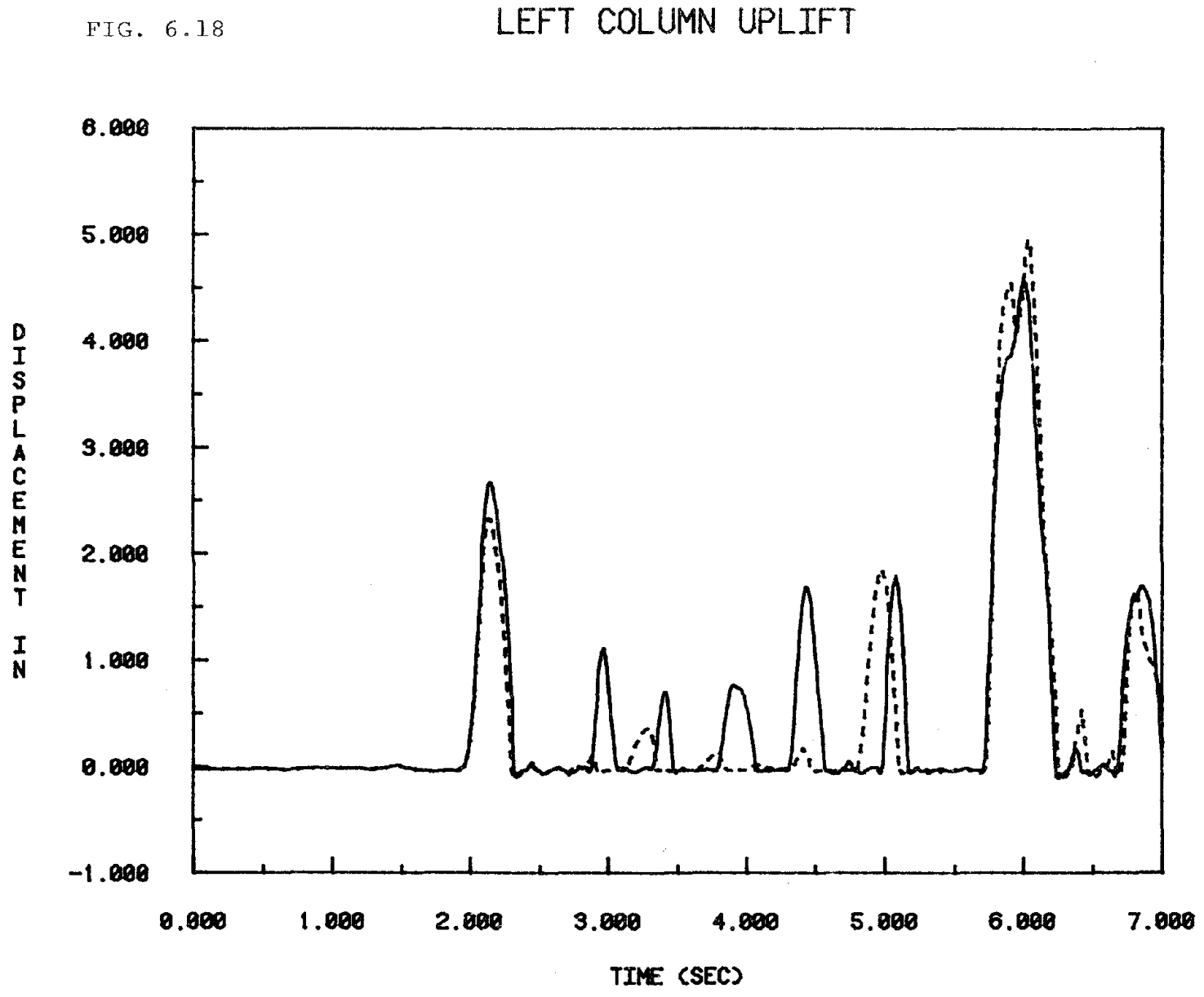

LEFT COLUMN UPLIFT
6.000
5.000
4.000
3.000
2.000
1.000
0.000
-1.000
DISPLACEMENT IN
0.000
1.000
2.000
3.000
4.000
5.000
6.000
7.000
TIME (SEC)
CASE B1
CASE B3

FIG. 6.19

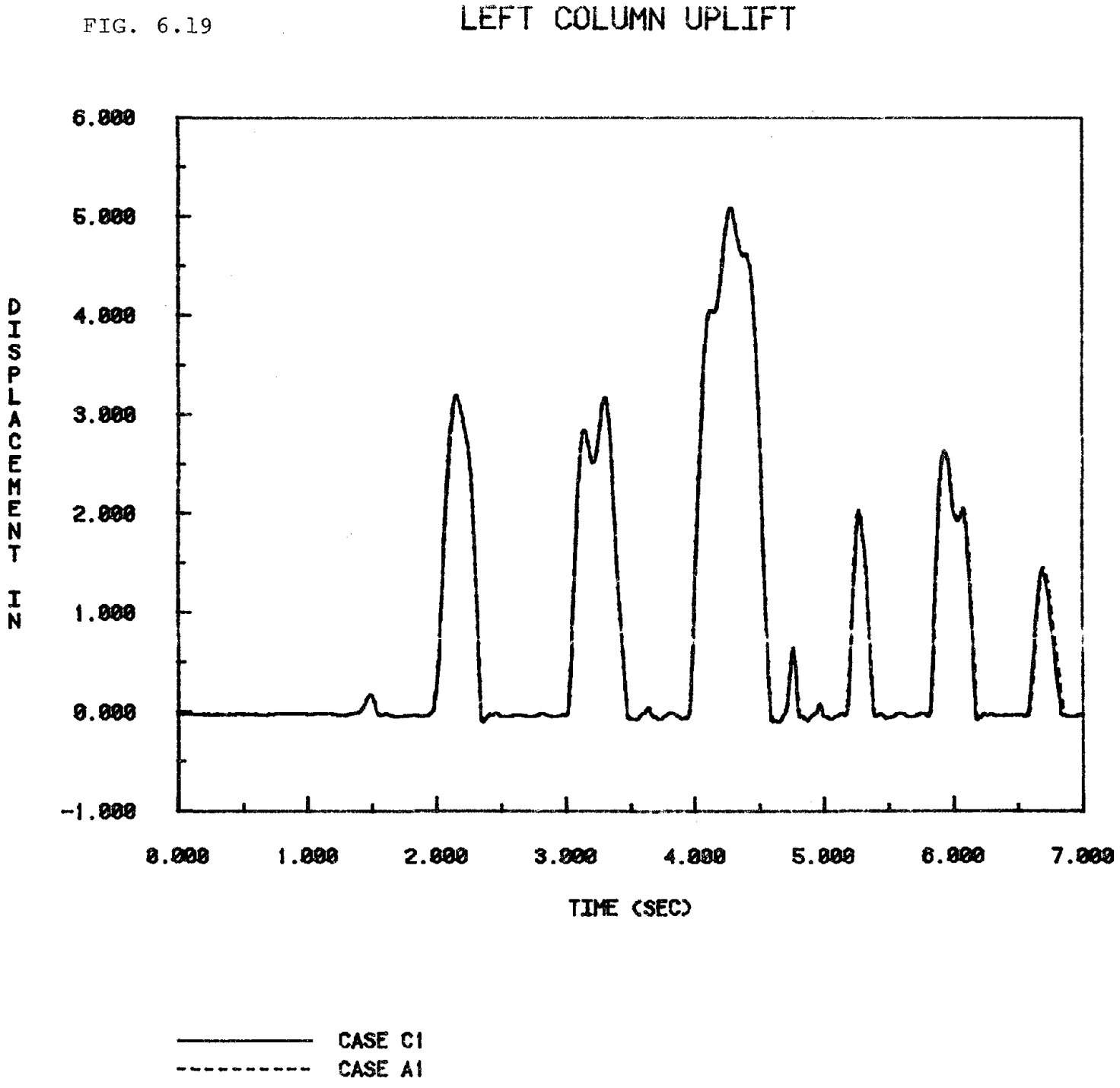

LEFT COLUMN UPLIFT
6.000
5.000
4.000
3.000
2.000
1.000
0.000
-1.000
DISPLACEMENT IN
0.000
1.000
2.000
3.000
4.000
5.000
6.000
7.000
TIME (SEC)
CASE C1
CASE A1

FIG. 6.20

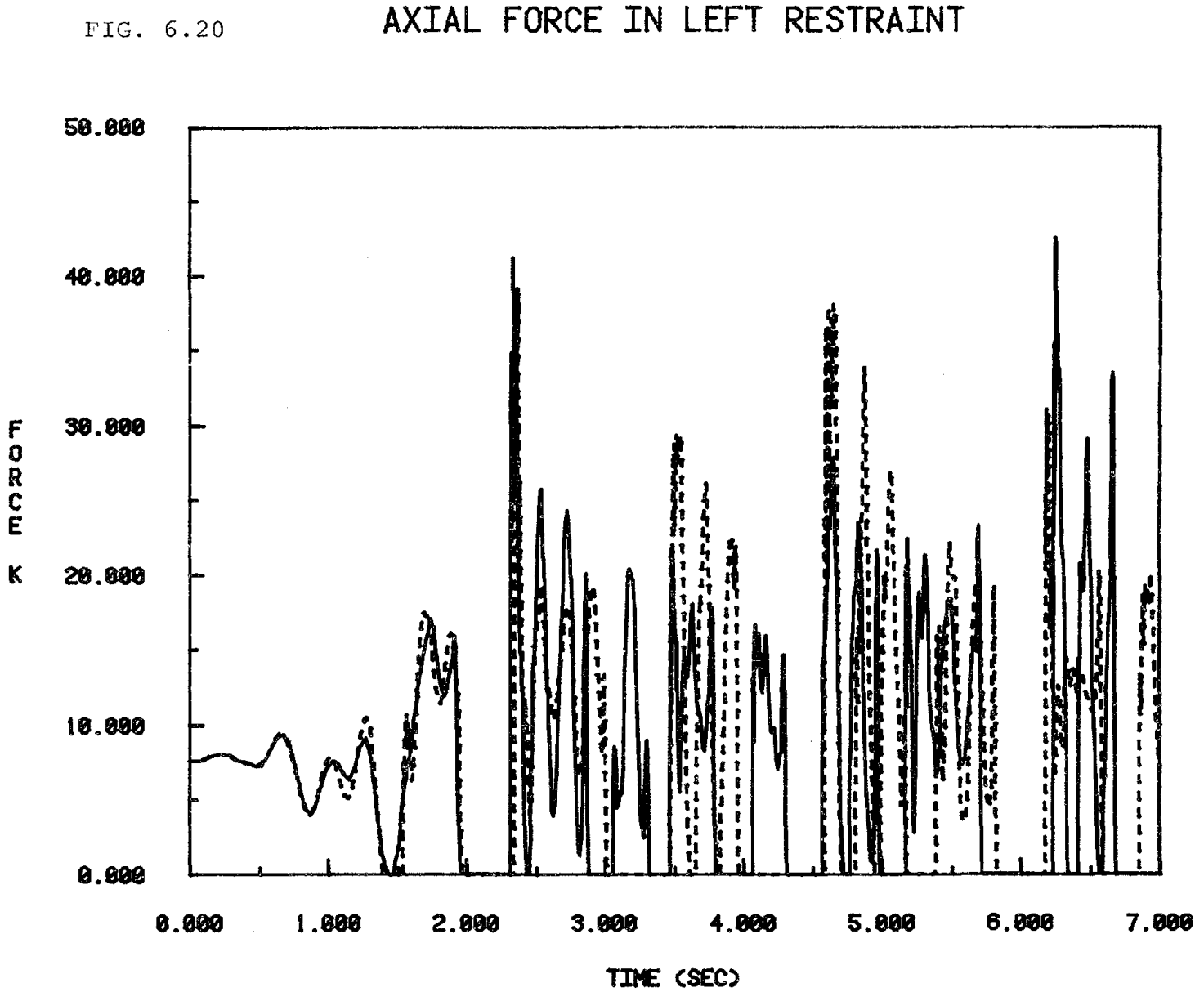

AXIAL FORCE IN LEFT RESTRAINT
50.000
40.000
30.000
20.000
10.000
0.000
FORCE K
0.000
1.000
2.000
3.000
4.000
5.000
6.000
7.000
TIME (SEC)
CASE B1
CASE A1

FIG. 6.21

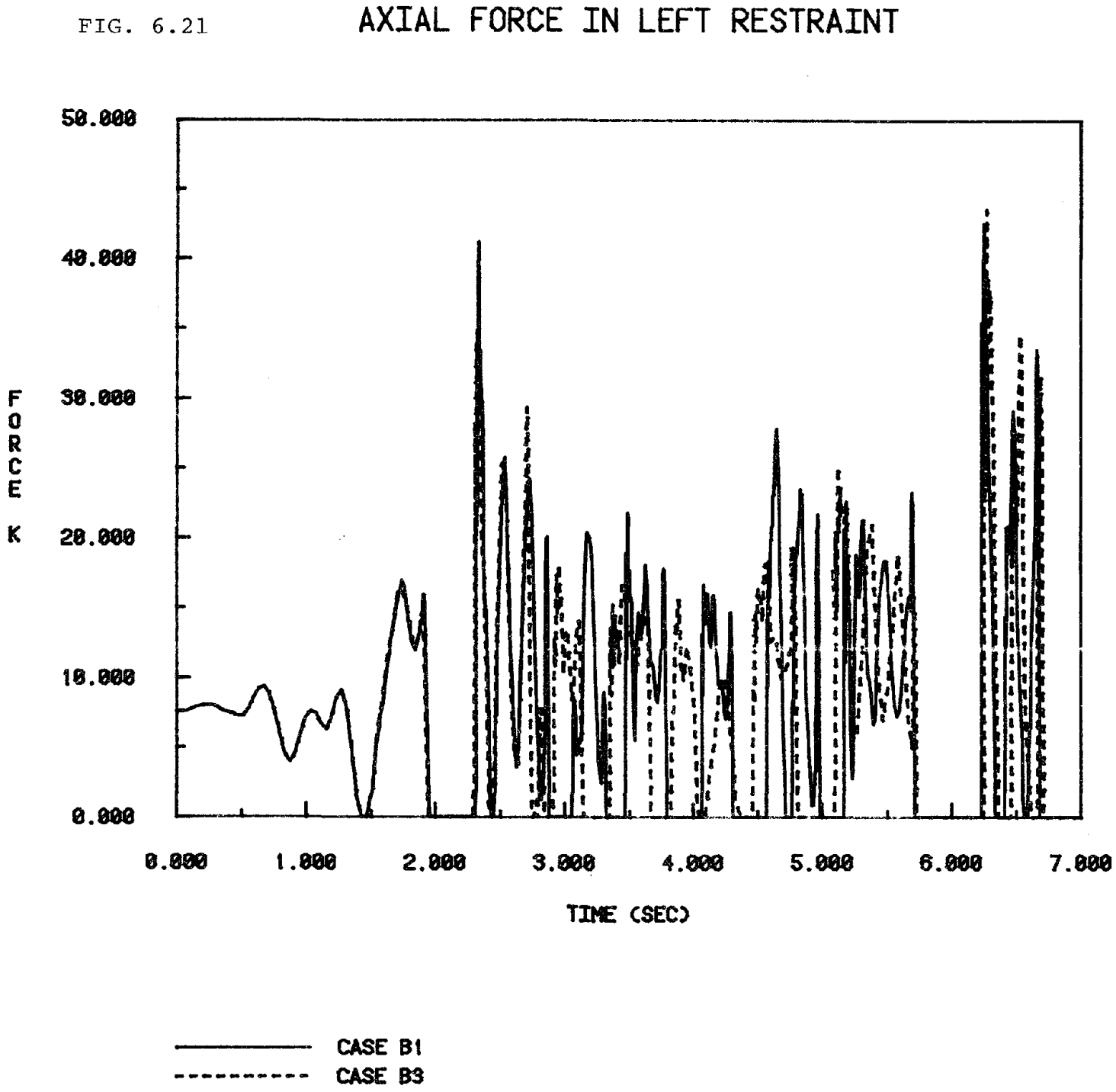

AXIAL FORCE IN LEFT RESTRAINT
50.000
40.000
30.000
20.000
10.000
0.000
FORCE K
0.000
1.000
2.000
3.000
4.000
5.000
6.000
7.000
TIME (SEC)
CASE B1
CASE B3

FIG. 6.22

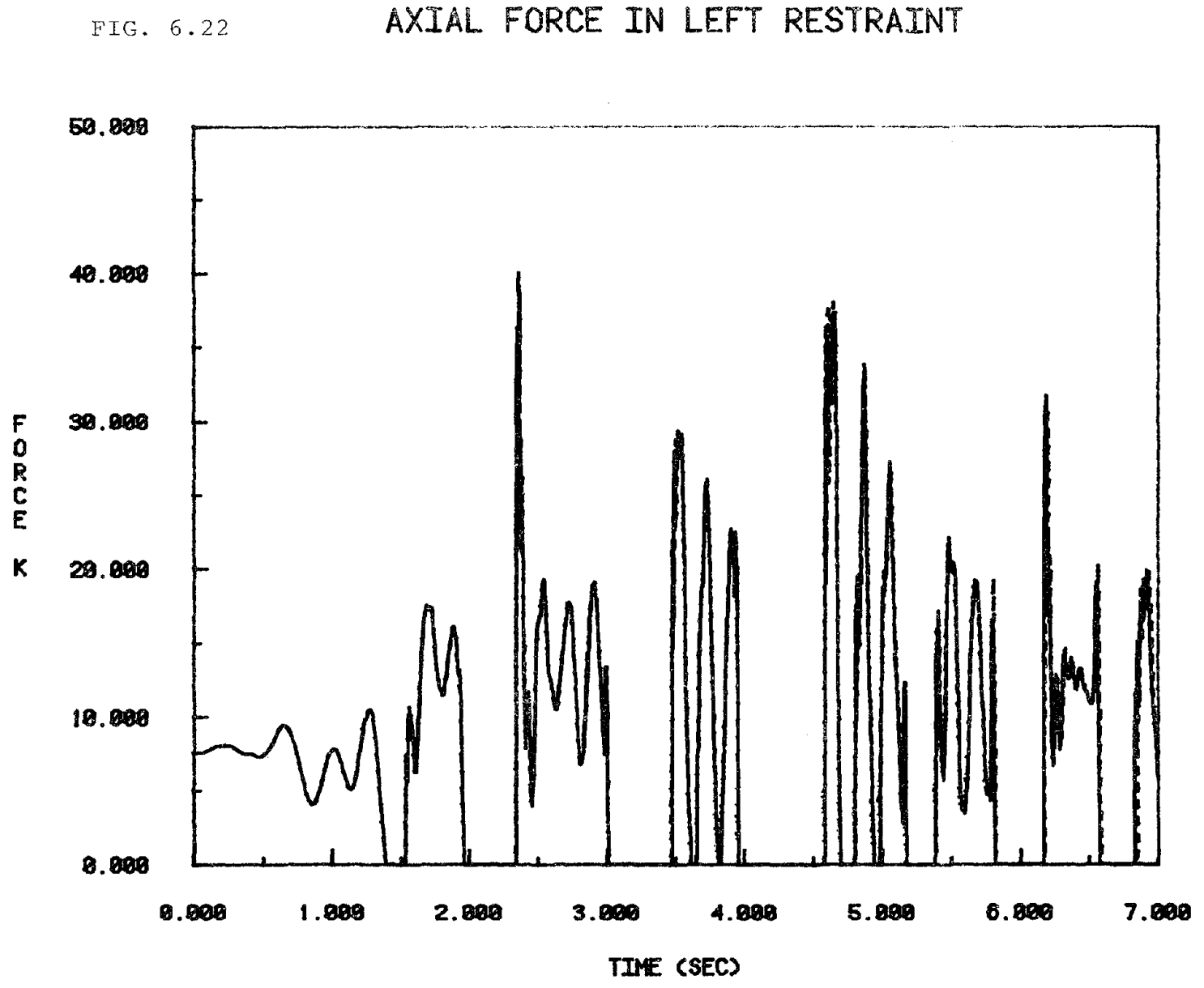

AXIAL FORCE IN LEFT RESTRAINT
50.000
40.000
30.000
20.000
10.000
0.000
FORCE K
0.000
1.000
2.000
3.000
4.000
5.000
6.000
7.000
TIME (SEC)
CASE C1
CASE A1

FIG. 6.23

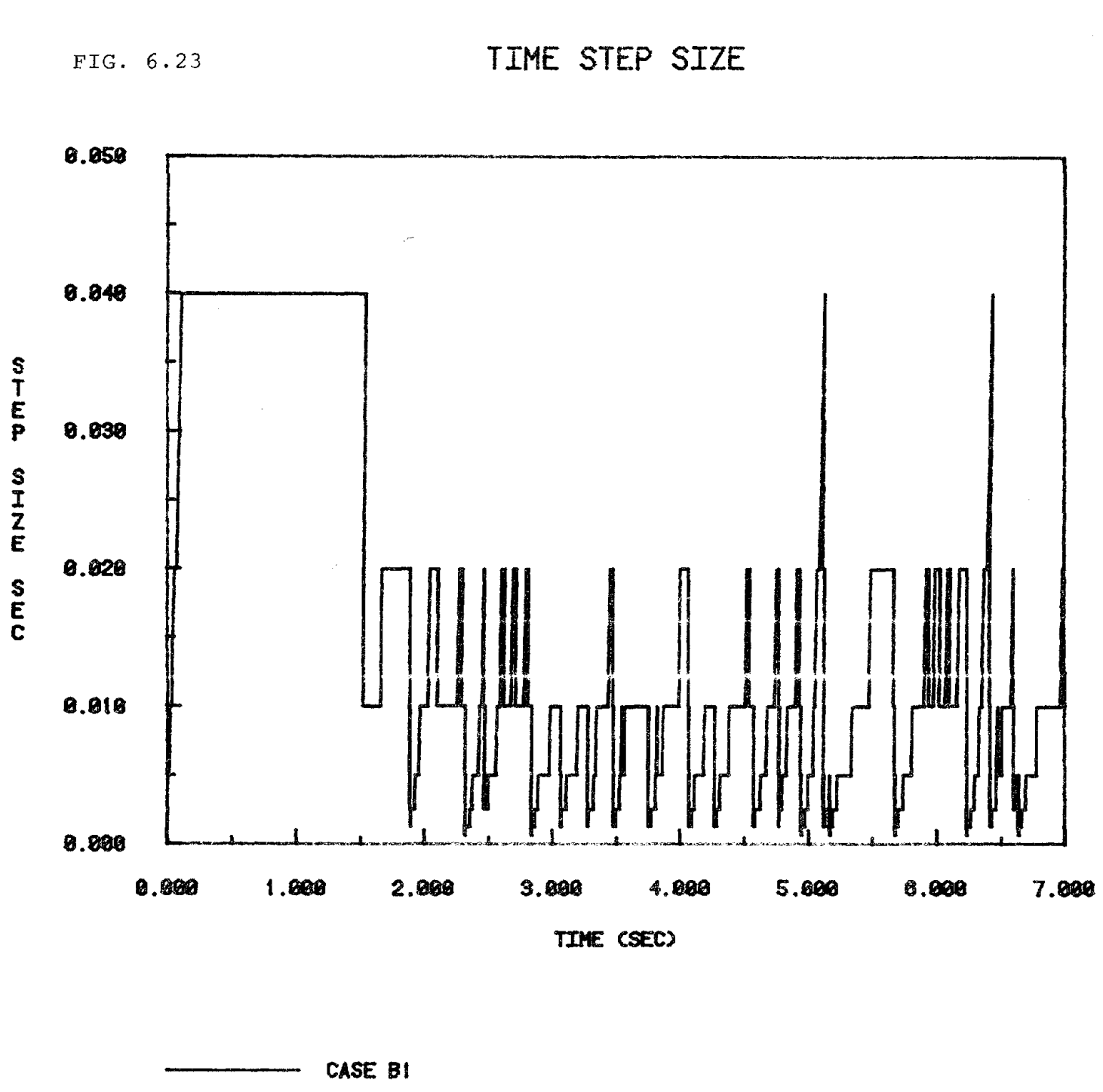

TIME STEP SIZE
0.050
0.040
0.030
0.020
0.010
0.000
STEP SIZE SEC
0.000
1.000
2.000
3.000
4.000
5.000
6.000
7.000
TIME (SEC)
CASE B1

FIG. 6.24

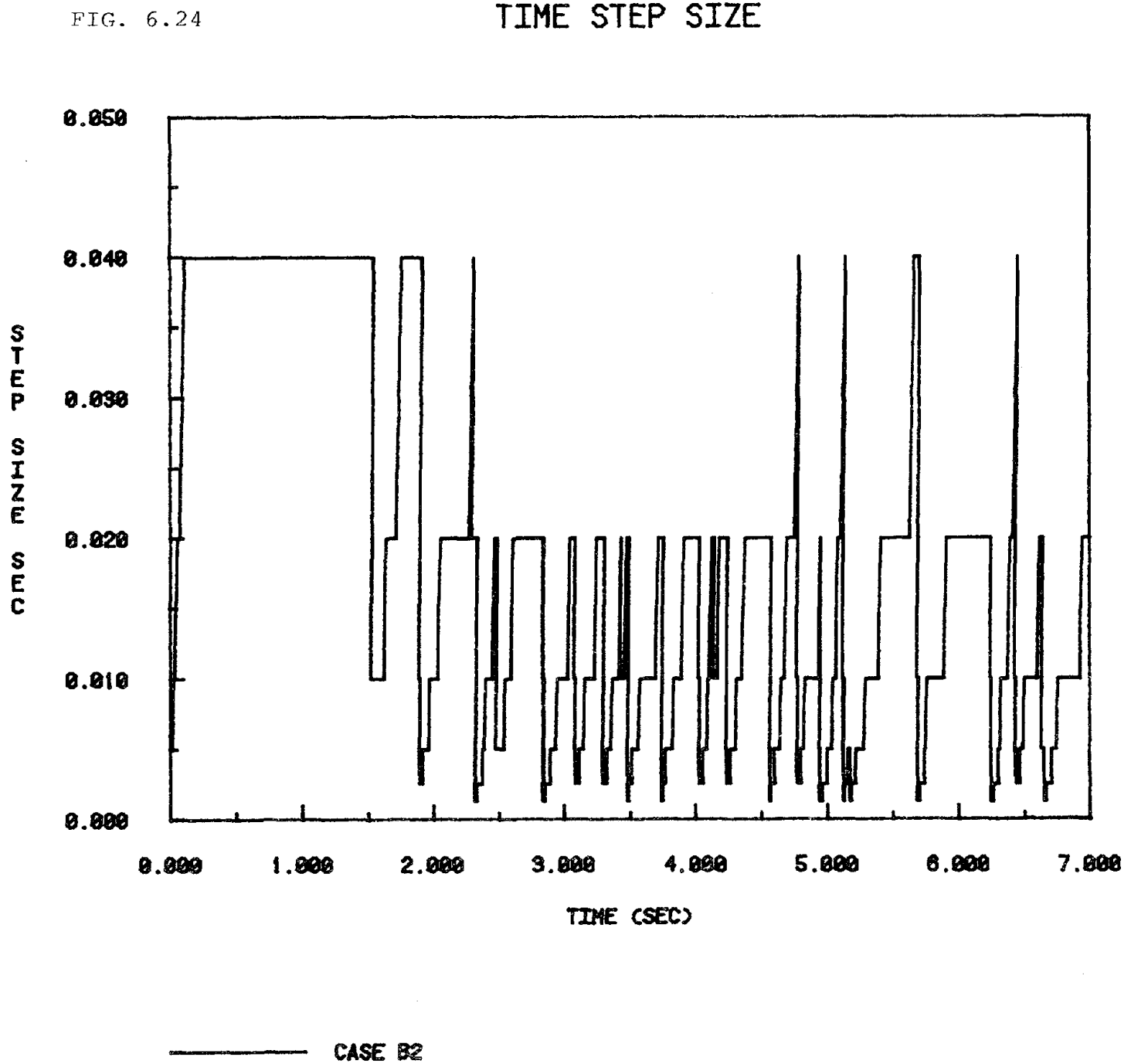

TIME STEP SIZE
0.050
0.040
0.030
0.020
0.010
0.000
STEP SIZE SEC
0.000
1.000
2.000
3.000
4.000
5.000
6.000
7.000
TIME (SEC)
CASE B2

FIG. 6.25

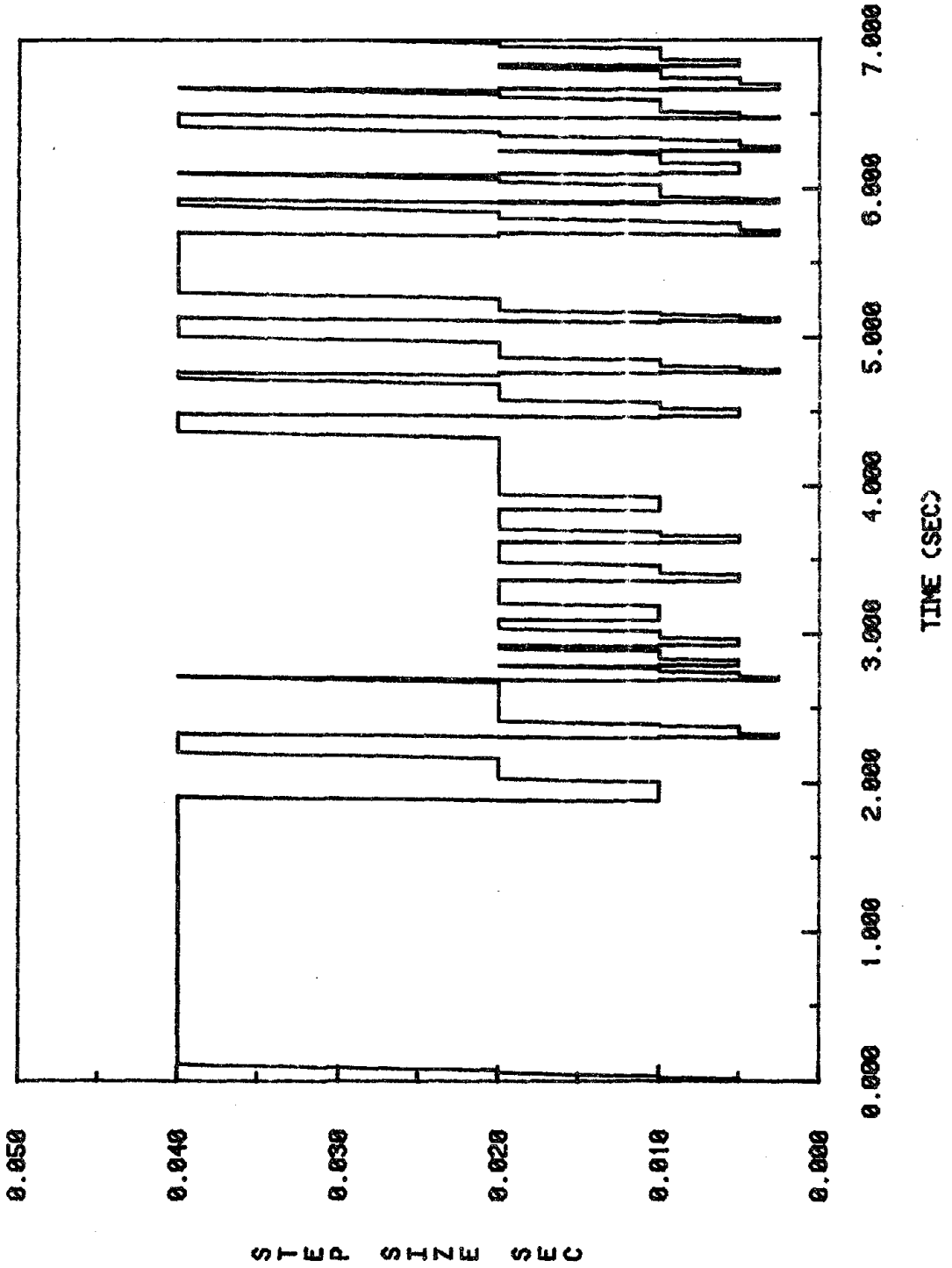

TIME STEP SIZE
0.050
0.040
0.030
0.020
0.010
0.000
STEP SIZE SEC
0.000
1.000
2.000
3.000
4.000
5.000
6.000
7.000
TIME (SEC)
CASE B3

FIG. 6.26

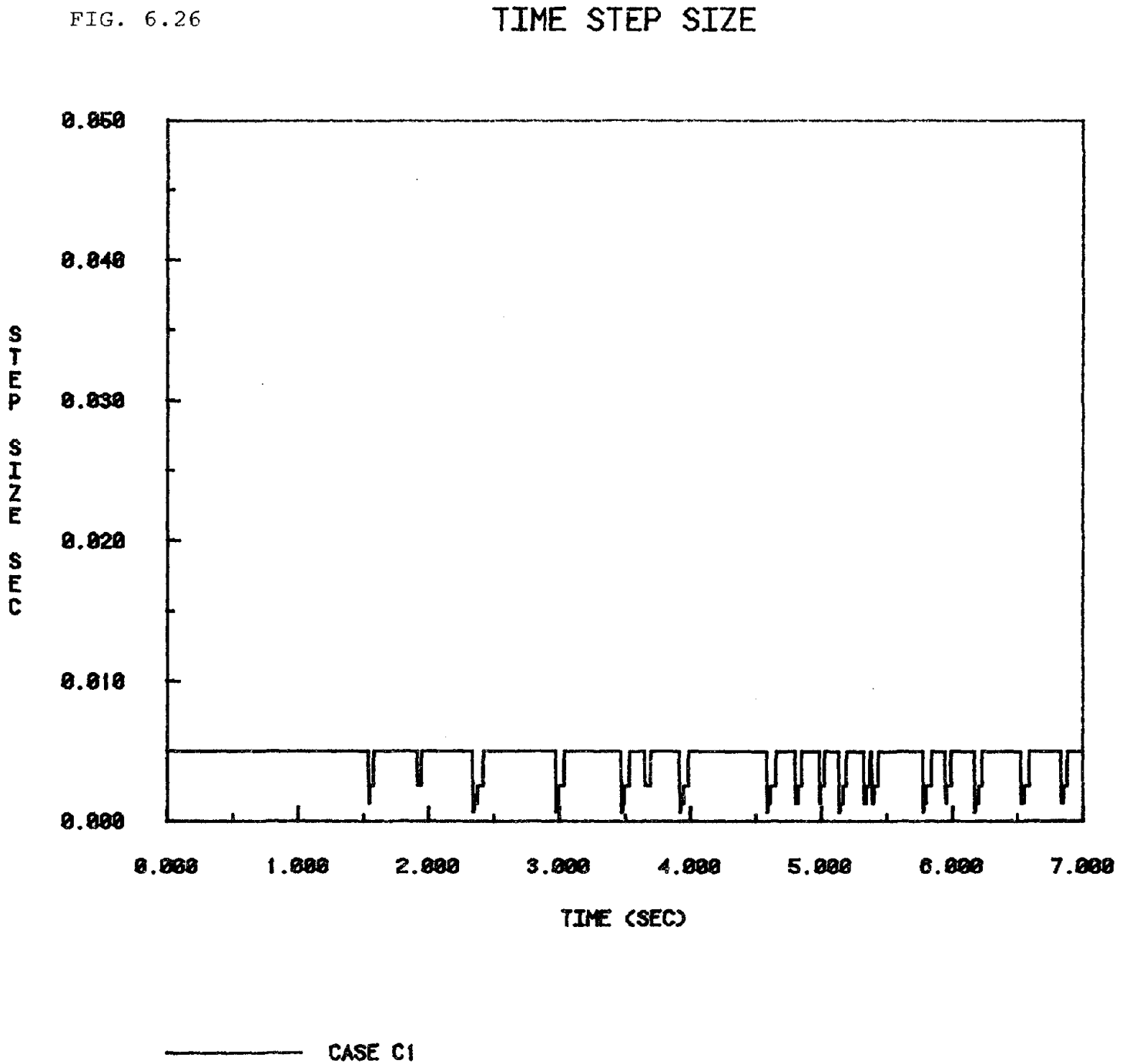

TIME STEP SIZE
0.050
0.040
0.030
0.020
0.010
0.000
STEP SIZE SEC
0.000
1.000
2.000
3.000
4.000
5.000
6.000
7.000
TIME (SEC)
CASE C1

FIG. 6.27

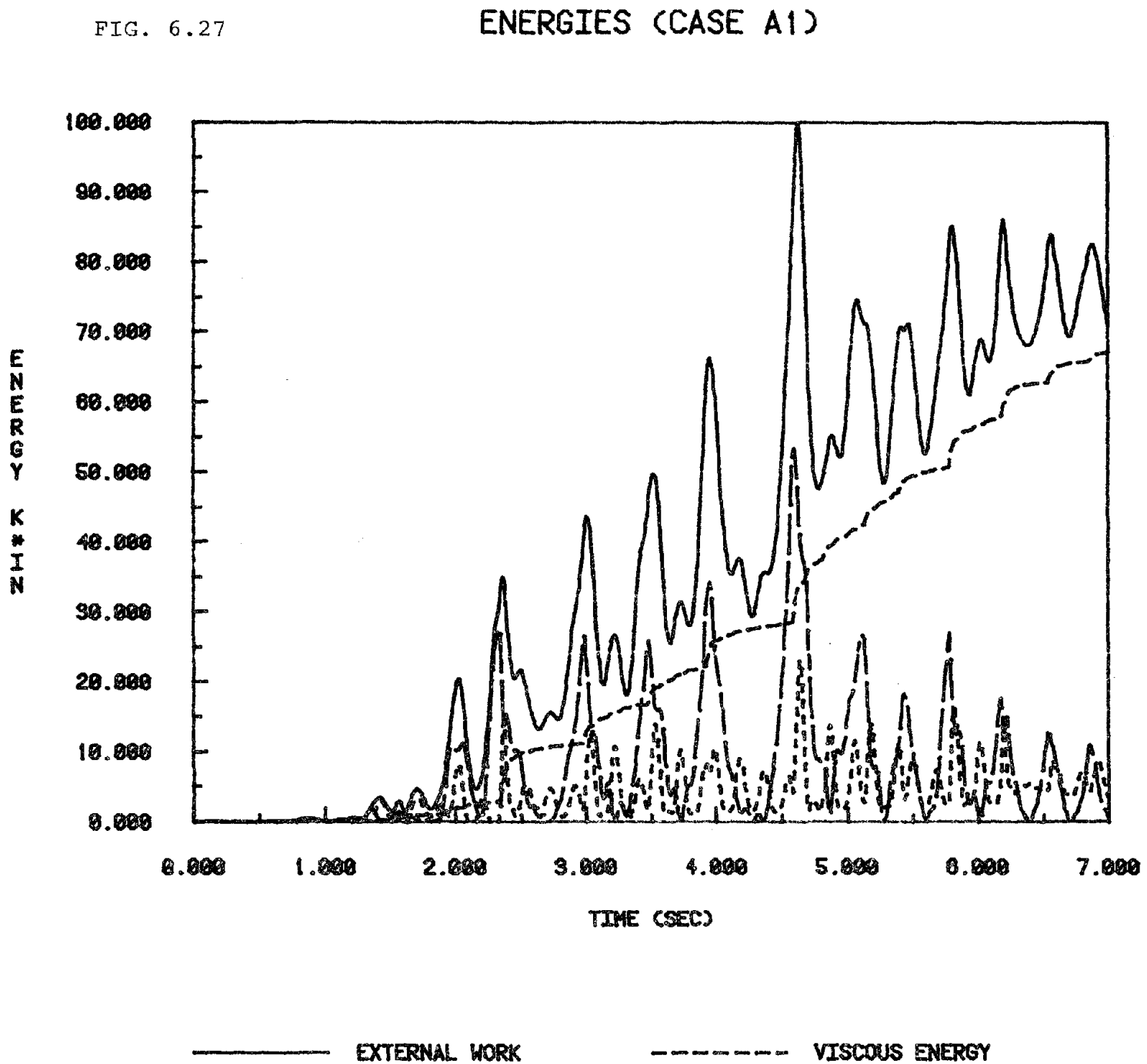

ENERGIES (CASE A1)
ENERGY K*IN
100.000
90.000
80.000
70.000
60.000
50.000
40.000
30.000
20.000
10.000
0.000
0.000
1.000
2.000
3.000
4.000
5.000
6.000
7.000
TIME (SEC)
EXTERNAL WORK
KINETIC ENERGY
VISCOUS ENERGY
ELASTO-PLASTIC ENERGY

FIG. 6.28

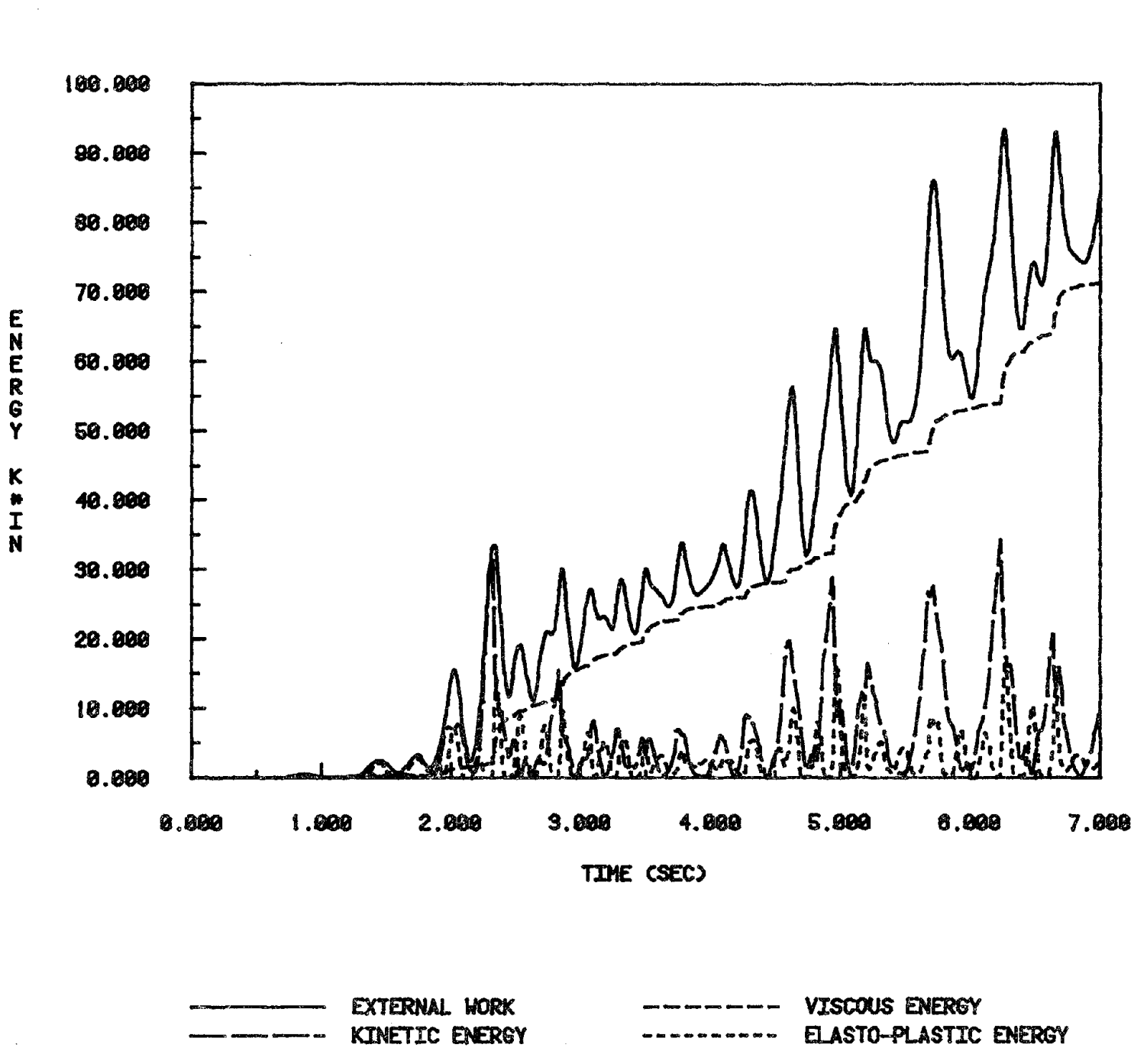

ENERGIES (CASE B1)
ENERGY K*IN
100.000
90.000
80.000
70.000
60.000
50.000
40.000
30.000
20.000
10.000
0.000
0.000
1.000
2.000
3.000
4.000
5.000
6.000
7.000
TIME (SEC)
EXTERNAL WORK
KINETIC ENERGY
VISCOUS ENERGY
ELASTO-PLASTIC ENERGY

FIG. 6.29

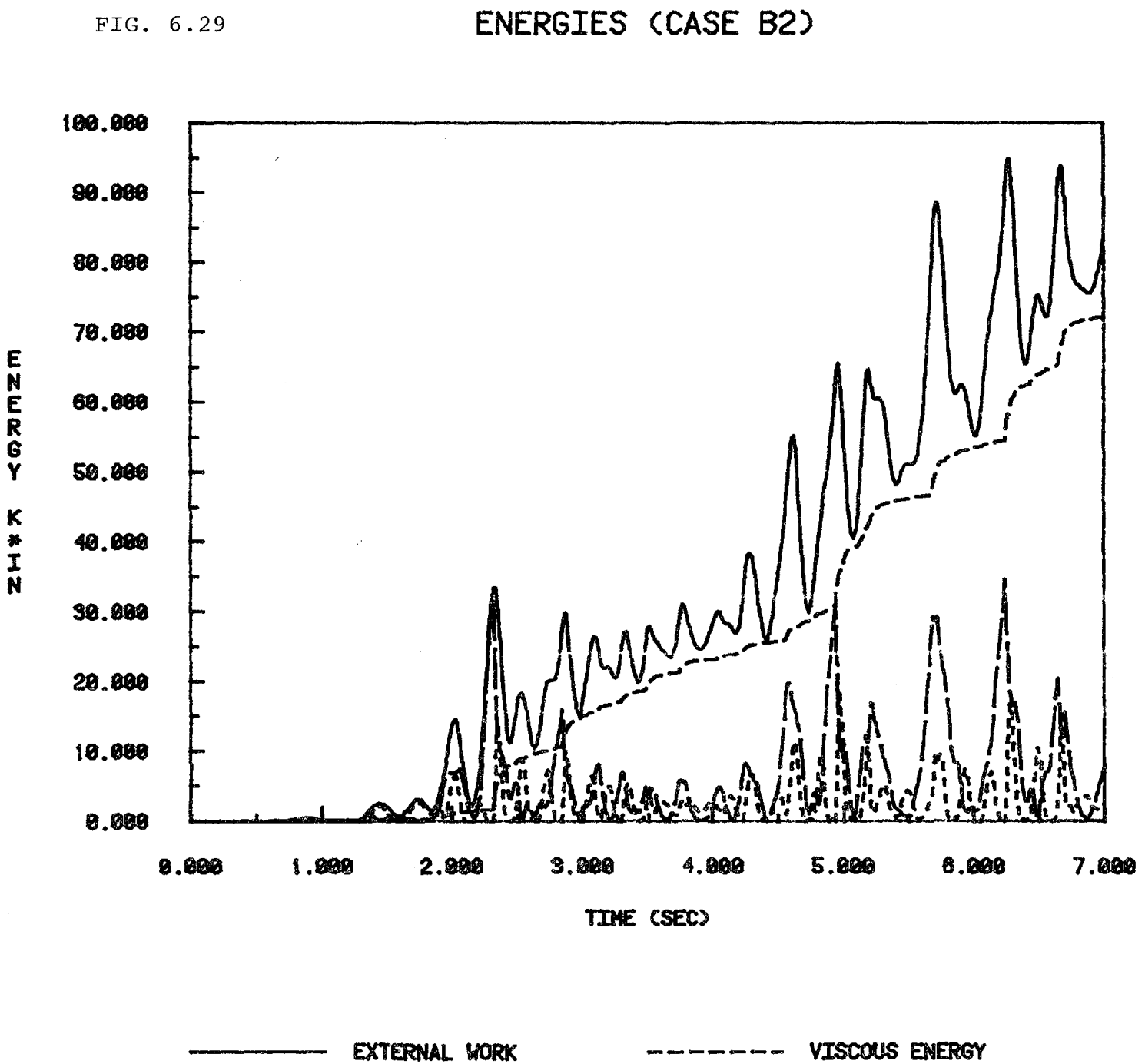

ENERGIES (CASE B2)
ENERGY K*IN
100.000
90.000
80.000
70.000
60.000
50.000
40.000
30.000
20.000
10.000
0.000
0.000
1.000
2.000
3.000
4.000
5.000
6.000
7.000
TIME (SEC)
EXTERNAL WORK
KINETIC ENERGY
VISCOUS ENERGY
ELASTO-PLASTIC ENERGY

FIG. 6.30

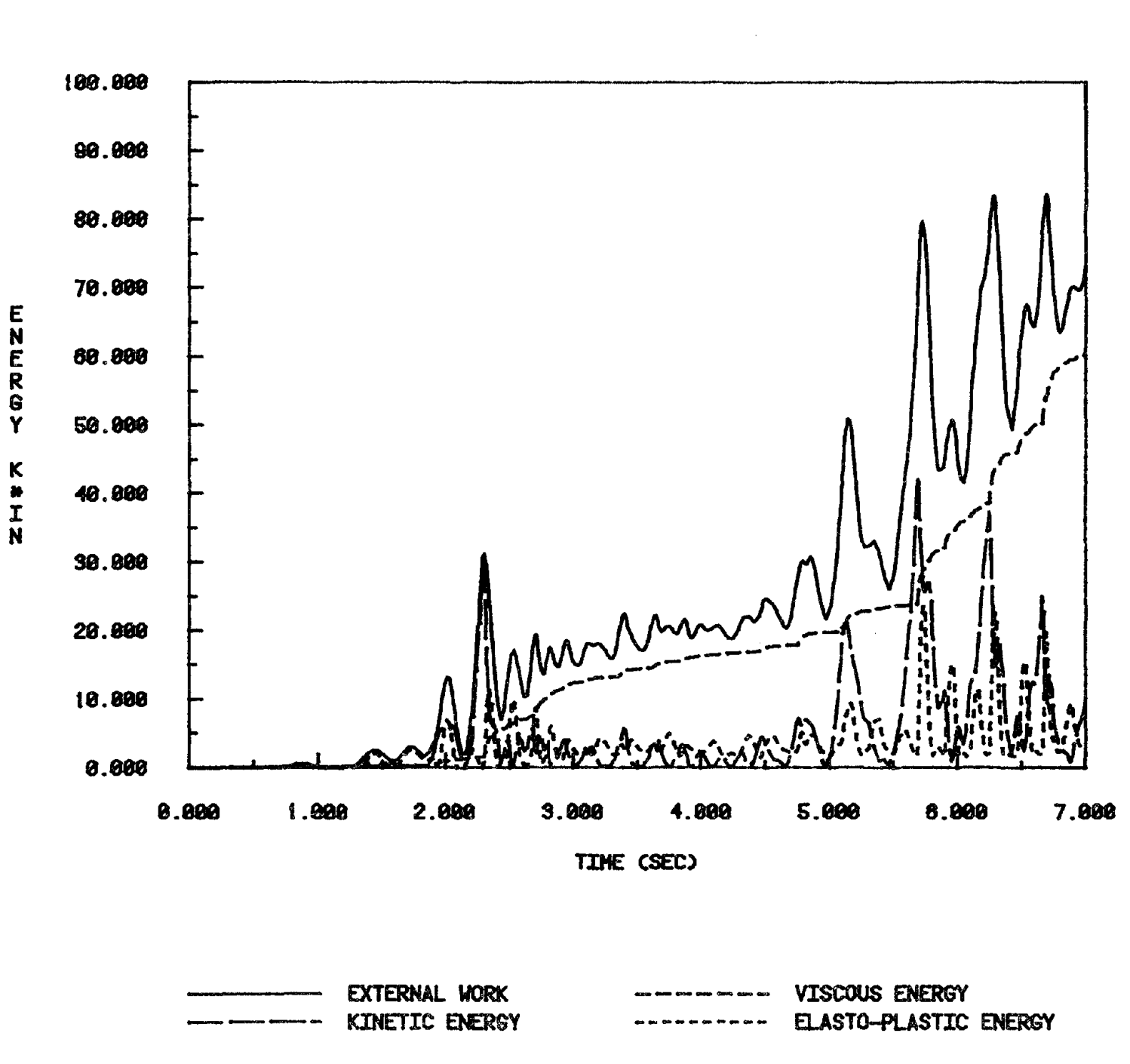

ENERGIES (CASE B3)
100.000
90.000
80.000
70.000
60.000
50.000
40.000
30.000
20.000
10.000
0.000
ENERGY K*IN
0.000
1.000
2.000
3.000
4.000
5.000
6.000
7.000
TIME (SEC)
EXTERNAL WORK
KINETIC ENERGY
VISCOUS ENERGY
ELASTO-PLASTIC ENERGY

FIG. 6.31

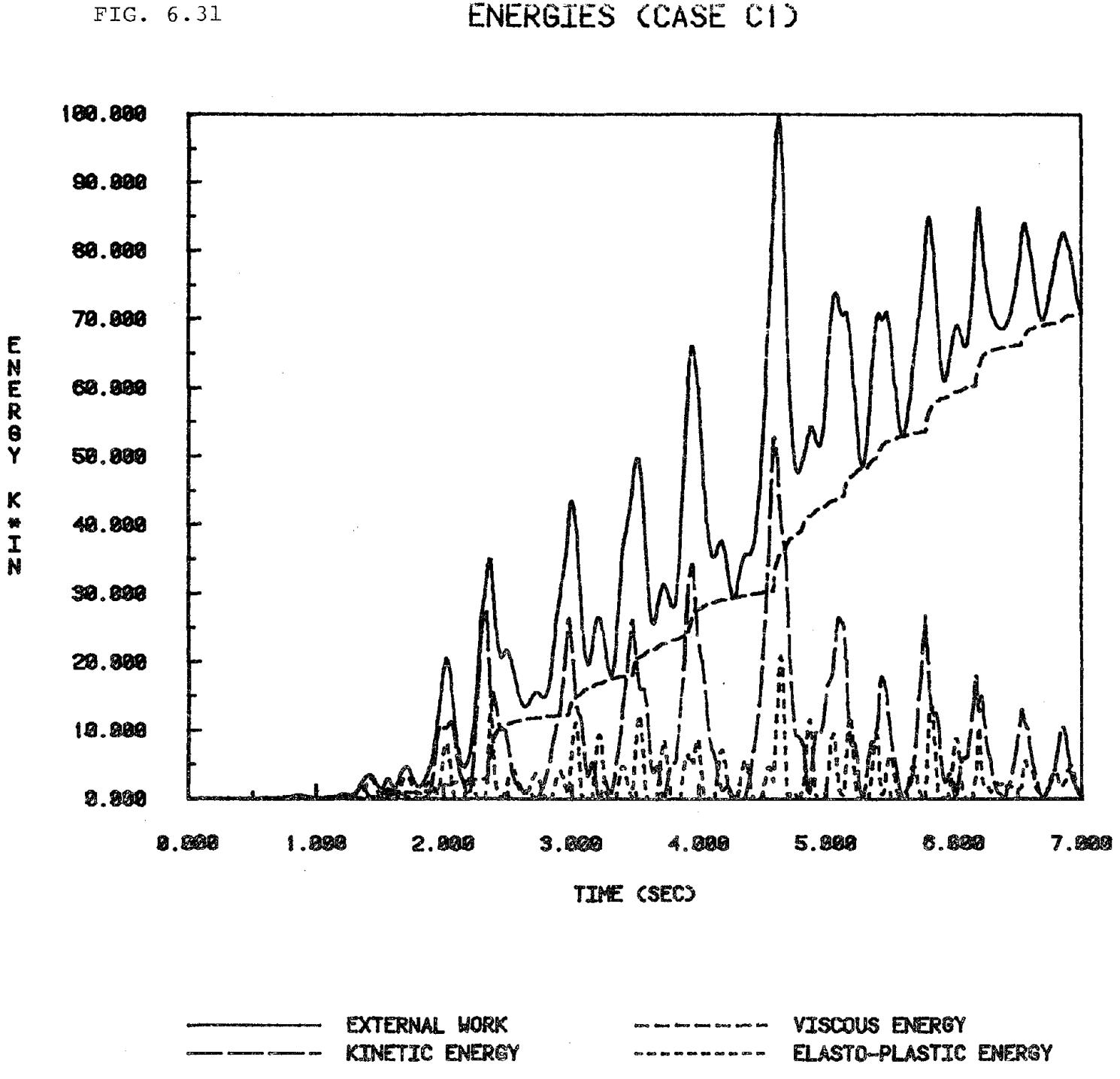

ENERGIES (CASE C1)
100.000
90.000
80.000
70.000
60.000
50.000
40.000
30.000
20.000
10.000
0.000
ENERGY K*IN
0.000
1.000
2.000
3.000
4.000
5.000
6.000
7.000
TIME (SEC)
EXTERNAL WORK
KINETIC ENERGY
VISCOUS ENERGY
ELASTO-PLASTIC ENERGY

FIG. 6.32

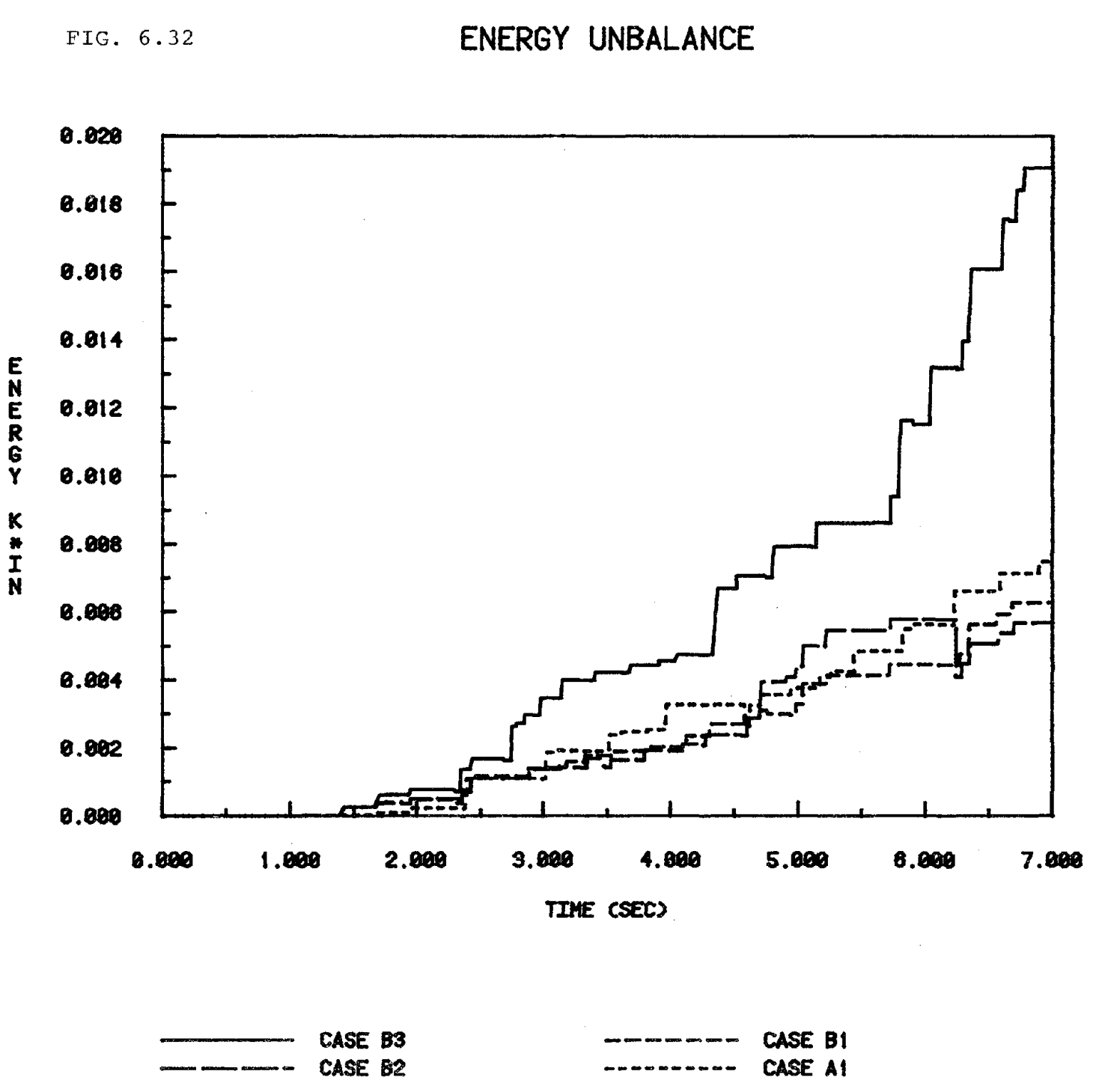

ENERGY UNBALANCE
0.020
0.018
0.016
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
ENERGY K*IN
0.000
1.000
2.000
3.000
4.000
5.000
6.000
7.000
TIME (SEC)
CASE B3
CASE B2
CASE B1
CASE A1

FIG. 6.33

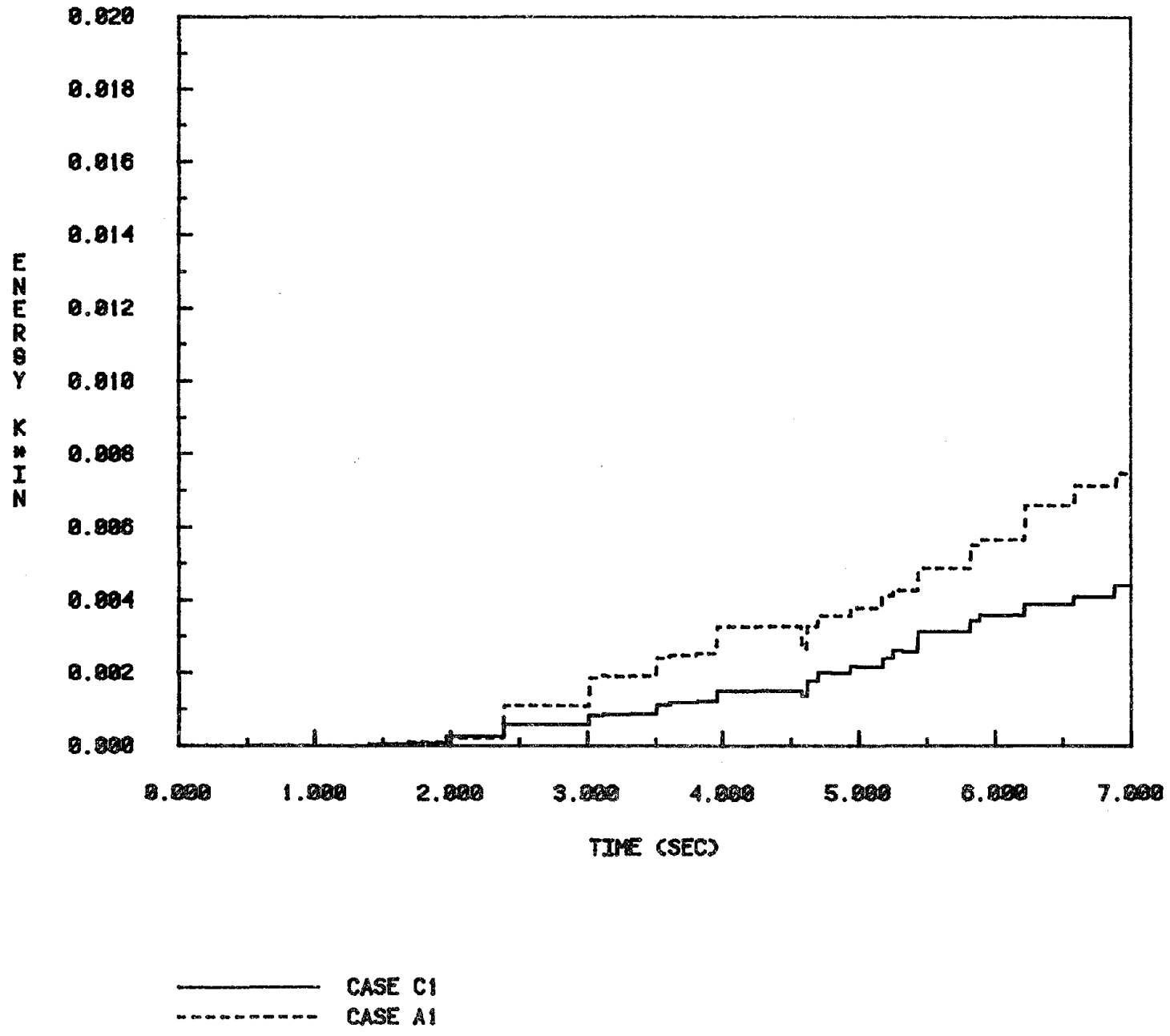

ENERGY UNBALANCE
0.020
0.018
0.016
0.014
0.012
0.010
0.008
0.006
0.004
0.002
0.000
ENERGY KMIN
0.000
1.000
2.000
3.000
4.000
5.000
6.000
7.000
TIME (SEC)
CASE C1
CASE A1

TABLE 3.1 - OVERALL LOGIC OF ANSR-III

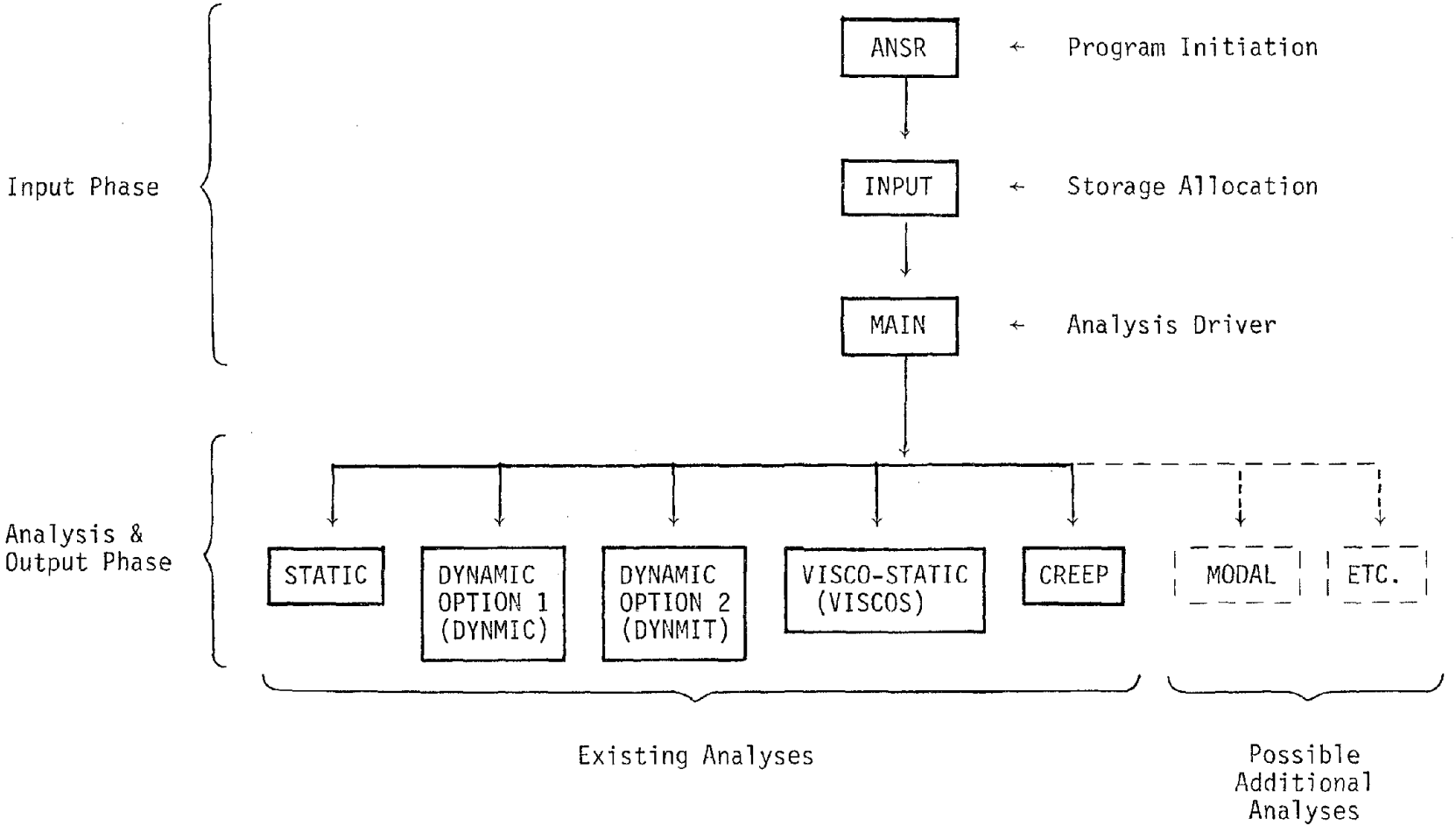


TABLE 3.2

LOGIC FOR EVENT-TO-EVENT DYNAMIC ANALYSIS (DYNMIC OPTION)

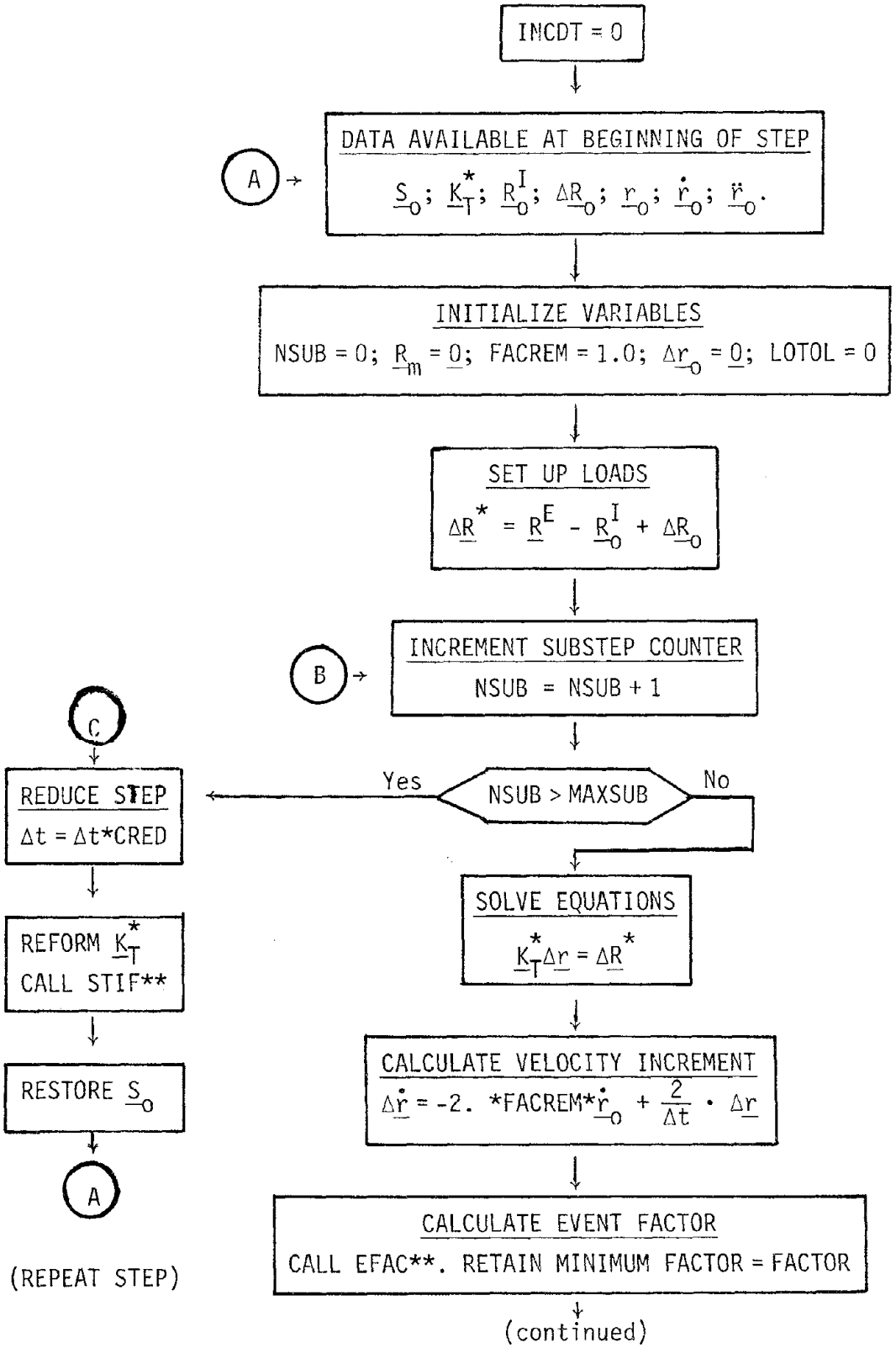
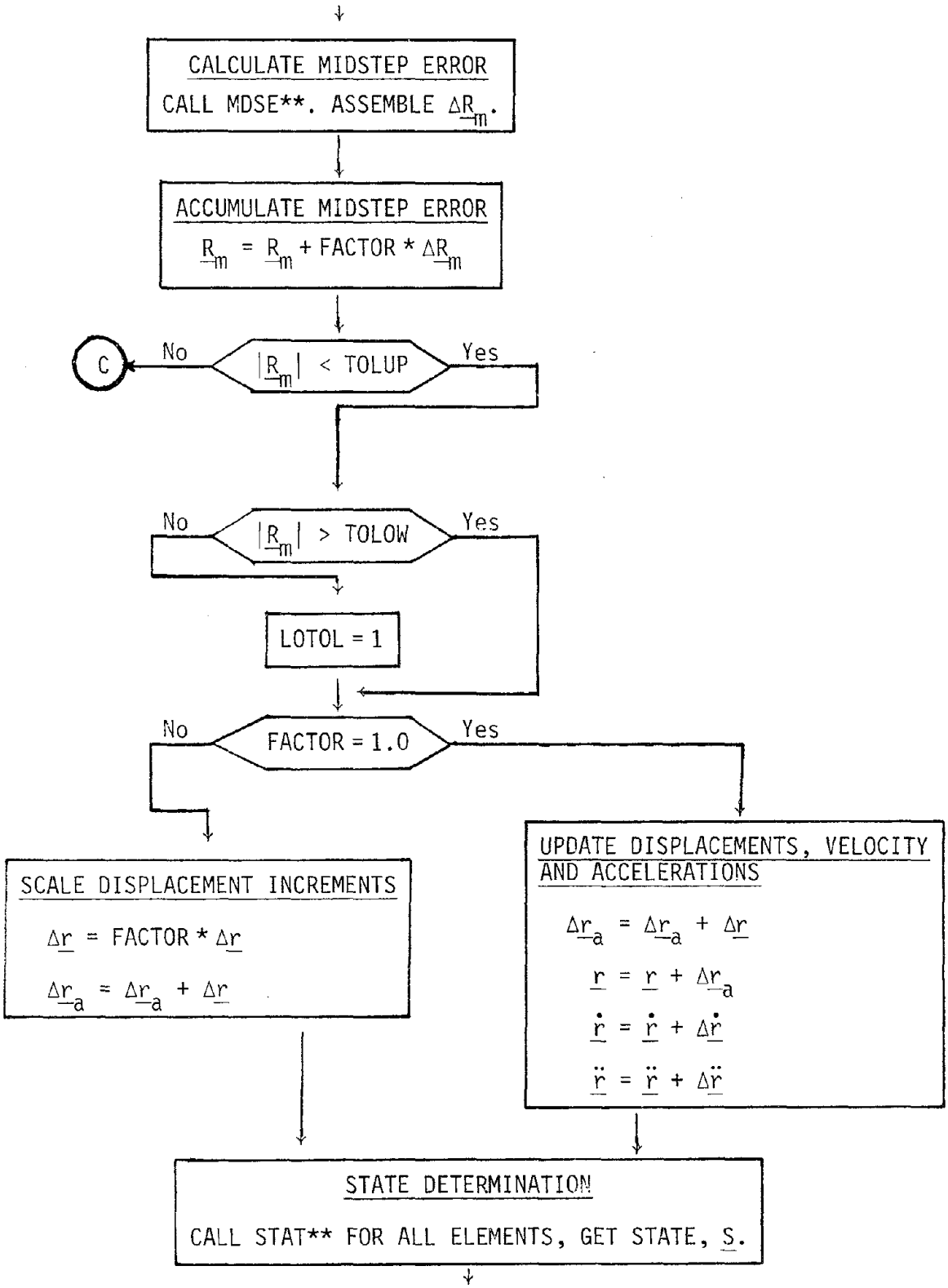


Table 3.2 (cont'd)



(continued)

Table 3.2 (cont'd)

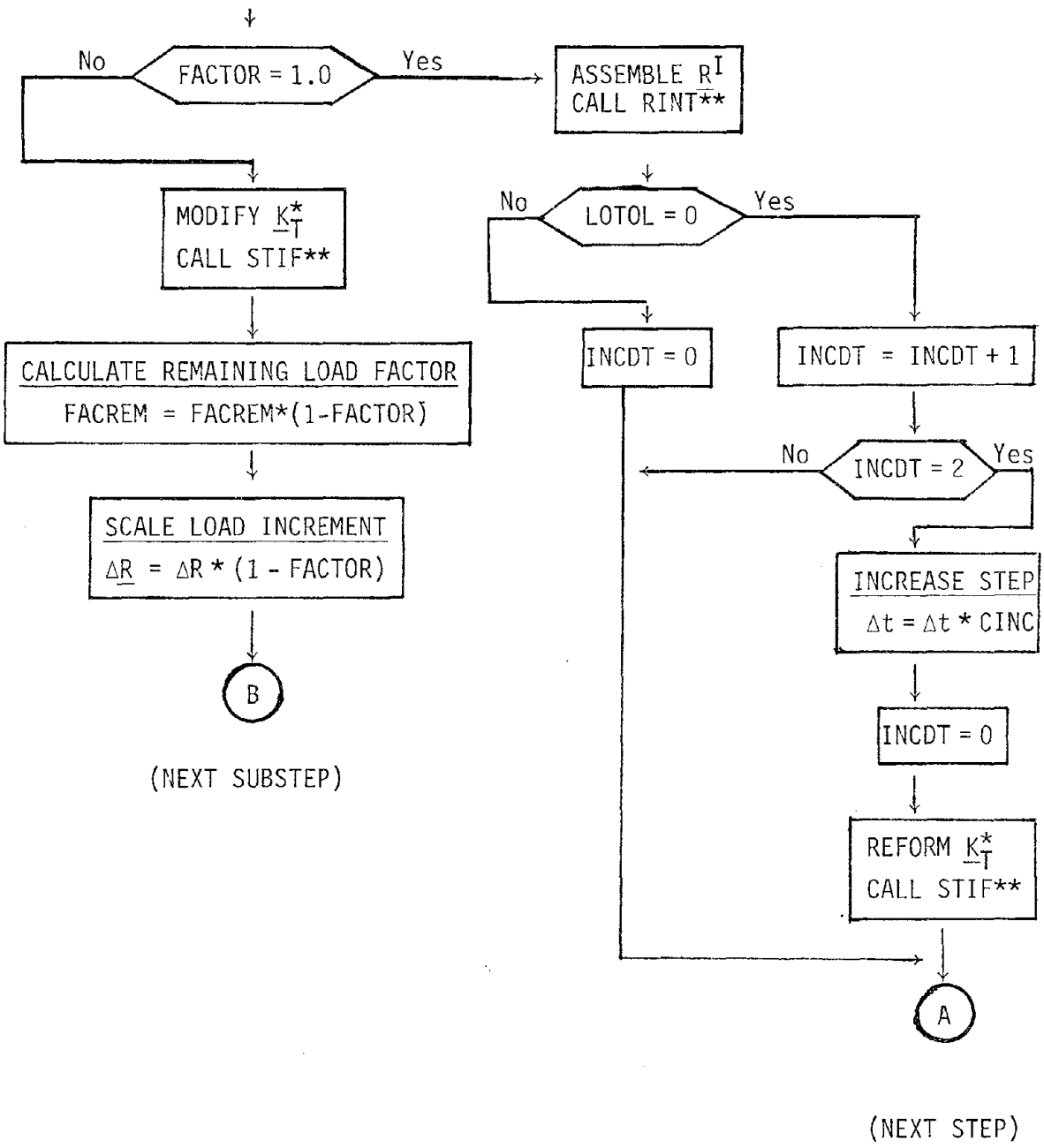
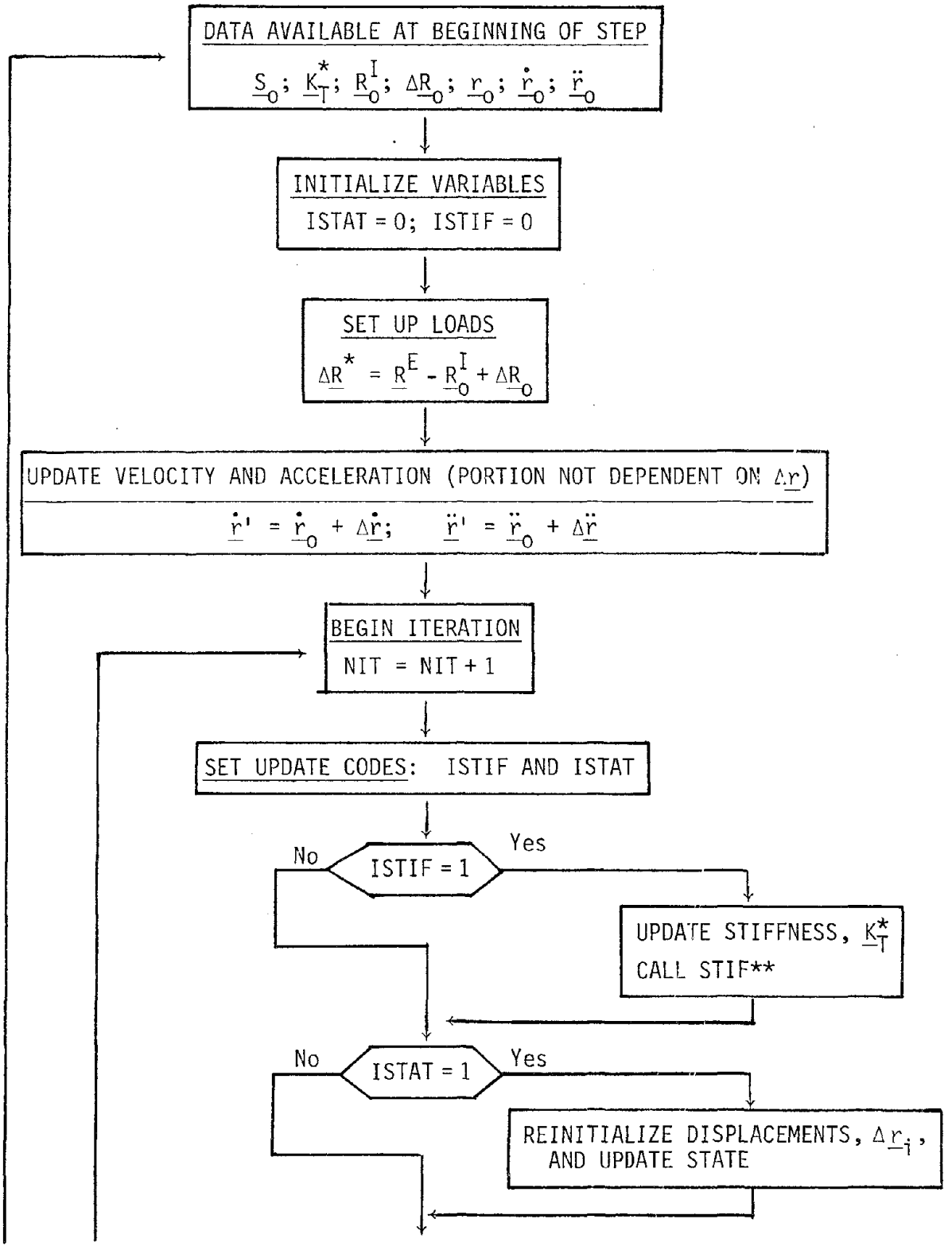


TABLE 3.3

LOGIC FOR DYNAMIC ANALYSIS WITH ITERATION (DYNMIT OPTION)



(continued)

Table 3.3 (cont'd)

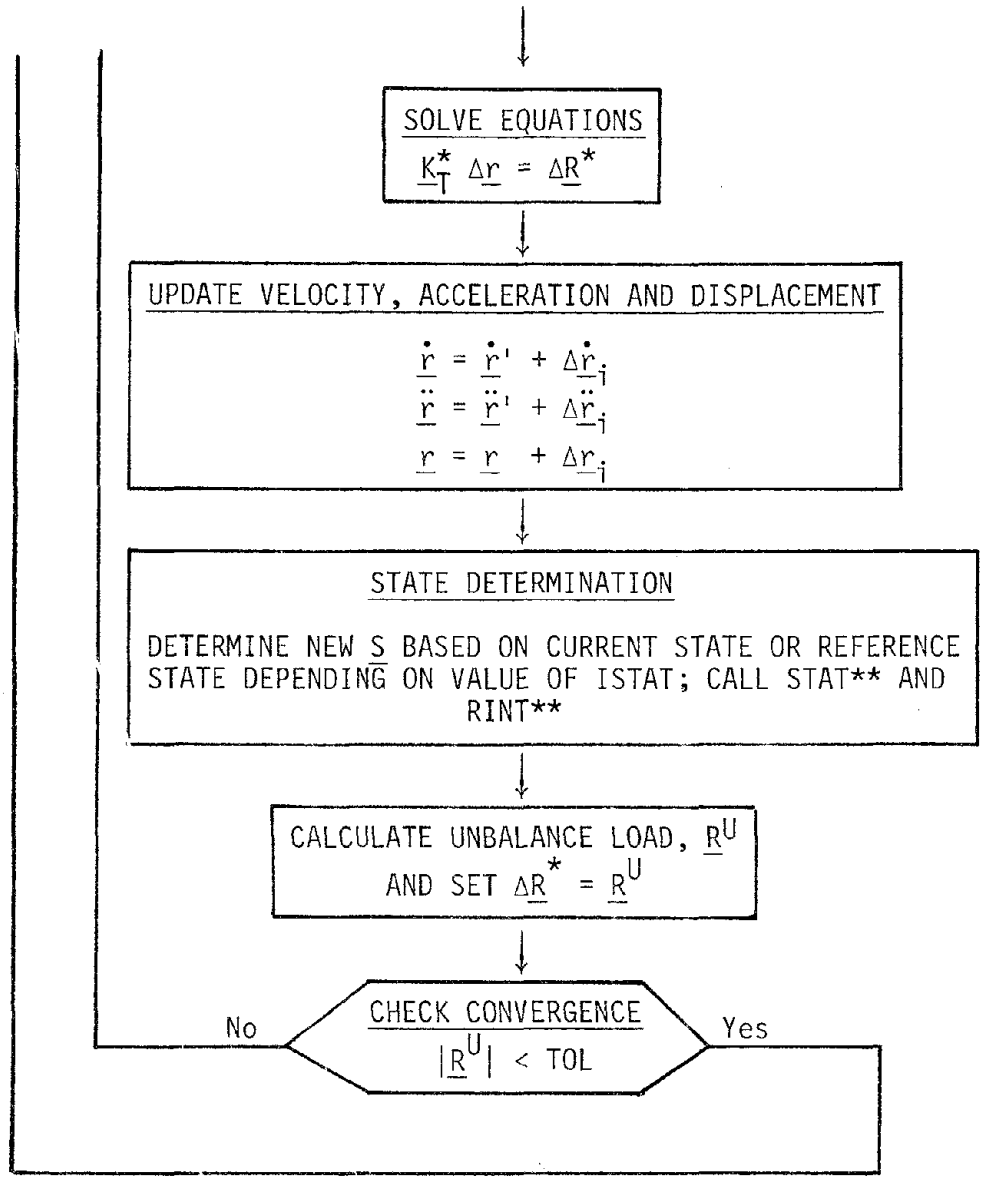


TABLE 6.1

INPUT DATA, CASE B1

```
star      3-story single bay uplift steel 2-d frame
nofl
core  symm
  19    15     2     0     0     3     2     9     2     0
   1           0.          208.          0.
   4         144.          208.          0.
   5           0.          144.          0.
   8         144.          144.          0.
   9           0.           80.          0.
  10          48.           80.          0.
  11          72.           80.          0.
  12          96.           80.          0.
  13         144.           80.          0.
  14           0.           40.          0.
  15         144.           40.          0.
  16           0.            0.          0.
  17         144.            0.          0.
  18           0.           10.          0.
  19         144.           10.          0.
   1     4     2     1
   5     8     2     1
   1     001110    15
  16     101110    17
  18     111111    19
1                1     2     3     4
1                5     6     7     8
   1.000544041.000544041          0.          0.          0.          0.     4     3
   2.005880829.005880829          0.          0.          0.          0.     3     1
   5.000621762.000621762          0.          0.          0.          0.     8     3
   6.005880829.005880829          0.          0.          0.          0.     7     1
   9.000582902.000582902          0.          0.          0.          0.    13     4
  10.005880829.005880829          0.          0.          0.          0.    12     2
  11.000103627.000103627          0.          0.          0.          0.
  14.000194301.000194301          0.          0.          0.          0.    15     1
  16.000077720.000077720          0.          0.          0.          0.    17     1
   5     2     1                                      u-bar restraint elements
   1     0
   1     3-.00000001         0.1
     400.        400.        400.
  40000.    4000000.
   1    16    18     1     0     1     0        .05     .00005
   2    17    19     1     0     1     0        .05     .00005
   6    20     1   .002196                      beam-column elements
   3     0     0
   1
  629300.    629200.    629100.    629000.       1000.       1100.       1200.
  629300.    629200.    629100.    629000.       1000.       1100.       1200.
  629300.    629200.    629100.    629000.       1000.       1100.       1200.
  102660.    102560.    102460.    102360.       1000.       1100.       1200.
   2
  617700.    617600.    617500.    617400.       1000.       1100.       1200.
  617700.    617600.    617500.    617400.       1000.       1100.       1200.
  617700.    617600.    617500.    617400.       1000.       1100.       1200.
  136300.    136200.    136100.    136000.       1000.       1100.       1200.
   3
   35960.     35860.     35760.     35660.       1000.       1100.       1200.
```

TABLE 6.1 (CONTINUED)

```
 35960.     35860.     35760.     35660.      1000.      1100.      1200.
 35960.     35860.     35760.     35660.      1000.      1100.      1200.
 41760.     41660.     41560.     41460.      1000.      1100.      1200.
  1    1     2    1     0    1     1    0     0    1     0    0     0

  4    5     6    1     0    1     1    0     0    1     0    0     0

  7    9    10    1     0    1     1    0     0    1     0    0     0

 11    1     5    3     0    2     2    0     0    1     0    0     0

 13    5     9    1     0    2     2    0     0    1     0    0     1

 14    8    13    1     0    2     2    0     0    1     0    0     0

 15    9    14    1     0    2     2    0     0    1     0    0     0

 16   13    15    1     0    2     2    0     0    1     0    0     0

 17   14    16    1     0    2     2    0     0    1     0    0     0

 18   15    17    1     0    2     2    0     0    1     0    0     0

 19   11    14    1     0    3     3    0     0    1     0    0     0

 20   11    15    1     0    3     3    0     0    1     0    0     0

  1
  9       gravity load
  1         0.       -.21          0.          0.          0.          0.     4     3
  2         0.      -2.27          0.          0.          0.          0.     3     1
  5         0.       -.24          0.          0.          0.          0.     8     3
  6         0.      -2.27          0.          0.          0.          0.     7     1
  9         0.       -.225         0.          0.          0.          0.    13     4
 10         0.      -2.27          0.          0.          0.          0.    12     2
 11         0.       -.04          0.          0.          0.          0.
 14         0.       -.075         0.          0.          0.          0.    15     1
 16         0.       -.03          0.          0.          0.          0.    17     1
  1    0    0    0    0  150
 el-centro 1940 n-s earthquake record                                    (12f6.0)
 150    1         0.       772.
0.000 .0000 0.042 .0010 0.097 .0159 0.161-.0001 0.221 .0189 0.263 .0001
0.291 .0059 0.332-.0012 0.374 .0200 0.429-.0237 0.471 .0076 0.581 .0425
0.623 .0094 0.665 .0138 0.720-.0088 0.720-.0256 0.789-.0387 0.789-.0568
0.872-.0232 0.872-.0343 0.941-.0402 0.941-.0603 0.997-.0789 1.066-.0666
1.066-.0381 1.094-.0429 1.168 .0897 1.315-.1696 1.384-.0828 1.412-.0828
1.440-.0945 1.481-.0885 1.509-.1080 1.537-.1280 1.628 .1144 1.703 .2355
1.800 .1423 1.855 .1777 1.924-.2610 2.007-.3194 2.215 .2952 2.270 .2634
2.320-.2984 2.395 .0054 2.450 .2865 2.519-.0469 2.575 .1516 2.652 .2077
2.708 .1087 2.769-.0325 2.893 .1033 2.976-.0803 3.068 .0520 3.129-.1547
3.212 .0065 3.253-.2060 3.386 .1927 3.419-.0937 3.530 .1708 3.599-.0359
3.668 .0365 3.738-.0736 3.835 .0311 3.904-.1833 4.014 .0227 4.056-.0435
4.106 .0216 4.222-.1972 4.314-.1762 4.416 .1460 4.471-.0047 4.618 .2572
4.665-.2045 4.756 .0608 4.831-.2733 4.970 .1779 5.039 .0301 5.108 .2183
5.199 .0267 5.233 .1252 5.302 .1290 5.330 .1089 5.343-.0239 5.454 .1723
5.510-.1021 5.606 .0141 5.690-.1949 5.773-.0242 5.800-.0050 5.809-.0275
```

TABLE 6.1 (CONTINUED)

```
5.869-.0573 5.883-.0327 5.925 .0216 5.980 .0108 6.013 .0235 6.085-.0665
6.132 .0014 6.174 .0493 6.188 .0149 6.188-.0200 6.229-.0381 6.279 .0207
6.326-.0058 6.368-.0603 6.382-.0162 6.409 .0200 6.459-.0176 6.478-.0033
6.520 .0043 6.534-.0040 6.562-.0099 6.575-.0017 6.603-.0170 6.645 .0373
6.686 .0457 6.714 .0385 6.728 .0009 6.769-.0288 6.769 .0016 6.811 .0113
6.852 .0022 6.908 .0092 6.991-.0996 7.074 .0360 7.121 .0078 7.143-.0277
7.149 .0026 7.171 .0272 7.226 .0576 7.295-.0492 7.370 .0297 7.406 .0109
7.425 .0186 7.461-.0253 7.525-.0347 7.572 .0036 7.600-.0628 7.641-.0280
7.669-.0196 7.691 .0068 7.752-.0054 7.794-.0603 7.835-.0357 7.877-.0716
7.960-.0140 7.987-.0056 8.001 .0222 8.070 .0468 8.126 .0260 8.126-.0335
   1    2
   4
  16   17
stat       static analysis with gravity load
   1
   1    0        0.    0    1        1.    1    0    1    0    5    1
   1    1    0    1    3    0
   0    0       .01       .01
       1.
dyn1       dynamic analysis using event-to-event strategy with variable time step
   1    0      .005       0.       0.       0.   -1    1
    .0004       .04       .5       2.       .48      .048
1400         7.         .1
   1         0.    1         0.
   1         0.         1.
none
stop
```

TABLE 6.2

NUMBER OF TIME STEPS

Case	No. of Steps ANSR-III	No. of Steps DRAIN-2D2
A1	1400	1400
A2	1400	-
B1	1141	1208
B2	845	990
B3	533	571
C1	1763	-

APPENDIX A

ANSR-III PROGRAM - USER'S GUIDE

APPENDIX A

ANSR-III PROGRAM - USER'S GUIDE

A. PROBLEM TITLE

One card.

COLUMNS	NOTE	NAME	DATA
2 - 5(A)			Execution indicator: (a) STAR : Execute. (b) CHEC: Data check only. (c) STOP : End of data.
9 - 80(A)			Problem title.

B. RESTART SPECIFICATION

One card.

COLUMNS	NOTE	NAME	DATA
2 - 5(A)	B.1	IDAT	Data file code: (a) OLDF: Use existing file. (b) NEWF: Start new file. (c) NOFL or blank: No file. Leave rest of card blank.
6 - 10(1)	B.1	NRSEQ	Data sequence number for restart (OLDF only). Zero or blank = unstressed state.
15(I)			Print code for restart data (OLDF only): (a) 0: No print. (b) 1: Geometry data only. (c) 2: Element states only. (d) 3: Geometry data and element states.
16 - 80(A)	B.2		Data file identifier (OLDF or NEWF only).

C. STORAGE ALLOCATION AND PROBLEM SIZE

C1. STORAGE ALLOCATION

One card.

COLUMNS	NOTE	NAME	DATA
2 - 5(A)	C.1		Storage scheme indicator: (a) CORE: Entirely in-core (b) DISK: Element data out-of-core. Stiffness may be in-core or blocked. Duplicate stiffness out-of-core.
7 - 10(A)	C.1		Stiffness matrix type: (a) SYMM: Symmetric (b) USYM: Unsymmetric
11 - 20(I)	C.1		Stiffness storage allocation (DISK option only). No. of words of core to be used for used for stiffness matrix storage. Default = all available.

C2. PROBLEM SIZE

One card. Omit for OLDF option.

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NODES	No. of nodes.
6 - 10(I)		NCNOD	No. of control nodes (coordinates specified directly). (Section D1)
11 - 15(I)		NODGC	No. of coordinate generation cards (Section D2).
16 - 20(I)		NCSYS	No. of rotated coordinate systems (Section D3).
21 - 25(I)		NDSYS	No. of coordinate assignment cards (Section D4).
26 - 30(I)		NDCON	No. of cards specifying deleted d.o.f. (Section D5).
31 - 35(I)		NIDDOF	No. of cards specifying equal d.o.f. (Section D6).
36 - 40(I)		NMSGC	No. of cards specifying nodal masses (Section D7).
41 - 45(I)	C.2	NELGR	No. of element groups.
46 - 50(I)		NCREEP	No. of sets of creep properties.

D. NODE DATA

Omit entire Section D if IDAT = OLDF.

D1. CONTROL NODE COORDINATES

NCNOD cards.

COLUMNS	NOTE	NAME	DATA
1 - 5(I)	D.1		Node number, in any sequence.
6 - 15(F)			X coordinate.
16 - 25(F)			Y coordinate.
26 - 35(F)			Z coordinate.

D2. COORDINATE GENERATION

NODGC cards.

COLUMNS	NOTE	NAME	DATA
1 - 5(I)	D.2	NB	Node at beginning of line.
6 - 10(I)		NE	Node at end of line.
11 - 15(I)		NOD	No. of nodes to be generated.
16 - 20(I)		ND	Node number difference. Leave blank if nodes are listed. May be negative.
21 - 30(F)	D.2	SPAC	Spacing between nodes: (a) Zero or blank: Uniform spacing. (b) < 1.0: Spacing = (length of line) x SPAC. (c) > 1.0: Spacing = SPAC. (d) < 0.0: Error.
31 - 80(I)			List of nodes. Up to 10 nodes, in 5-column fields. Omit if ND is nonzero.

D3. ROTATED COORDINATE SYSTEMS

NCSYS cards.

COLUMNS	NOTE	NAME	DATA
1 - 5(I)	D.3		Primary direction: (a) 1: rotated X. (b) 2: rotated Y. (c) 3: rotated Z.
6 - 16(F)			Direction cosine w.r.t. global X.
16 - 25(F)			Direction cosine w.r.t. global Y.
26 - 35(F)			Direction cosine w.r.t. global Z.
36 - 40(I)	D.3		Secondary direction: (a) 1: rotated X. (b) 2: rotated Y. (c) 3: rotated Z.
41 - 50(F)			Direction cosine w.r.t. global X.
51 - 60(F)			Direction cosine w.r.t. global Y.
61 - 70(F)			Direction cosine w.r.t. global Z.

D4. COORDINATE SYSTEM ASSIGNMENT

NDSYS cards.

COLUMNS	NOTE	NAME	DATA
1 - 5(I)	D.3	N	Node, or first node in a series.
6 - 10(I)			Rotated coordinate system number.
11 - 15(I)		NE	Last node in series. Blank if a single node or if nodes are listed.
16 - 20(I)		ND	Node number difference. Default = 1.
21 - 80(I)		NLIST	List of nodes. Up to 12 nodes, in successive 5-column fields. Omit if ND is nonzero.

D5. DELETED DEGREES OF FREEDOM

NDCON cards.

COLUMNS	NOTE	NAME	DATA
1 - 5(I)	D.4	N	Node, or first node in a series.
10(I)			Code for X translation: (a) 1: Deleted. (b) 0 or blank: Not deleted.
11(I)			Code for Y translation.
12(I)			Code for Z translation.
13(I)			Code for X rotation.
14(I)			Code for Y rotation.
15(I)			Code for Z rotation.
16 - 20(I)	D.5	NE	Last node in series. Blank if a single node or if nodes are listed.
21 - 25(I)		ND	Node number difference. Default = 1.
26 - 80(I)		NLIST	List of nodes. Up to 11 nodes, in successive 5-column fields. Omit if ND is nonzero.

D6. NODES WITH EQUAL DISPLACEMENTS

NIDDOF cards.

COLUMNS	NOTE	NAME	DATA
1(I)	D.6		Code for X translation: (a) 0: X displacements identical for all nodes in list (same d.o.f.). (b) 0 or blank: Not identical (different d.o.f.).
2(I)			Code for Y translation.
3(I)			Code for Z translation.
4(I)			Code for X rotation.
5(I)			Code for Y rotation.
6(I)			Code for Z rotation.
11 - 80(I)		NLIST	List of nodes. Up to 14 nodes, in successive 5-column fields.

D7. NODAL MASSES

NMSGC cards.

COLUMNS	NOTE	NAME	DATA
1 - 5(I)	D.7	NF	Node, or first node in a series.
6 - 15(F)			X mass.
16 - 25(F)			Y mass.
26 - 35(F)			Z mass.
36 - 45(F)			X rotational inertia.
46 - 55(F)			Y rotational inertia.
56 - 65(F)			Z rotational inertia.
66 - 70(I)		NL	Last node in series. Blank for a single node.
71 - 75(I)		ND	Node number difference. Default = 1.

D8. STORAGE ALLOCATION FOR CREEP PROPERTIES

NCREEP CARD.

COLUMNS	NOTE	NAME	DATA
1 - 5(I)	D.8	NCSS	No. of rows (stress values) in secondary dashpot property table.
6 - 10(I)		NCST	No. of columns (temperature values) in primary dashpot property table.
11 - 15(I)		NCPS	No. of rows in primary dashpot property table.
16 - 20(I)		NCPT	No. of columns in primary dashpot property table.
21 - 25(I)		NCSS	No. of rows in primary spring property table.
26 - 30(I)		NCST	No. of columns in primary spring property table.
31 - 35(I)		MNTPR	Max. no. of time steps in creep analysis.

D9. CREEP PROPERTIES

NCREEP sets of cards.

D9(a). CONTROL CARD

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		KLOG	Interpolation Code: (a) 0: Interpolate linearly between natural values. (b) 1: Interpolate linearly between logarithmic values.
11 - 30(A)		IFORM	Input format.

D9(b). PROPERTIES FOR SECONDARY DASHPOT

As many cards as needed to input table CS (NCSS, NCPT), row by row with format IFORM.

D9(c). PROPERTIES FOR PRIMARY DASHPOT

As many cards as needed to input table CP (NCPS, NCPT), row by row with format IFORM.

D9(d). PROPERTIES FOR PRIMARY SPRING

As many cards as needed to input table KP (NKPS, NKPT), row by row with format IFORM.

E. ELEMENT DATA

Insert cards defining elements. The input requirements are described in separate reports.

Omit entire Section E if IDAT = OLDF.

F. STATIC LOAD PATTERNS

Omit entire Section F if IDAT = OLDF.

F1. CONTROL INFORMATION

One card.

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NPNF	No. of nodal force patterns (Section F2).
6 - 10(I)		NPFF	No. of follower force patterns (Section F3).
11 - 15(I)		NPTP	No. of nodal temperature patterns (Section F4). These patterns may also define internal pressures for pipe elements.
16 - 20(I)		NPP4	No. of 4-node surface pressure patterns (Section F5).
21 - 25(I)		NPND	No. of imposed nodal displacement patterns (Section F6).

F2. NODAL FORCE PATTERNS

NPNF sets of cards.

F2(a). CONTROL INFORMATION

One card.

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NCNF	No. of cards defining this pattern.
9 - 80(A)			Optional pattern title.

F2(b). NODAL FORCES

NCNF cards.

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NF	Node, or first node in a series.
6 - 15(F)			X force.
16 - 25(F)			Y force.
26 - 35(F)			Z force.
36 - 45(F)			X moment.
46 - 55(F)			Y moment.
56 - 65(F)			Z moment.
66 - 70(I)		NL	Last node in series. Blank for a single node.
71 - 75(I)		ND	Node no. difference. Default = 1.

F3. FOLLOWER FORCE PATTERNS

NPFF sets of cards.

F3(a). CONTROL INFORMATION

One card.

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NCFF	No. of cards defining this pattern.
9 - 80(A)			Optional pattern title.

F3(b). FOLLOWER FORCES

NCFF cards.

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NODA	Loaded node.
6 - 10(I)		NODD	Follower node.
11 - 20(F)			Force, positive towards follower node.

F4. NODAL TEMPERATURE PATTERNS

NPTP sets of cards.

F4(a). CONTROL INFORMATION

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NCTP	No. of cards defining this pattern.
9 - 80(A)			Optional pattern title.

F4(b). TEMPERATURES AND PRESSURES

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NF	First node in a series.
6 - 10(I)		NL	Last node in series. Blank for a single node.
11 - 15(I)		ND	Node number difference. Default = 1.
16 - 25(F)		TEMP	Temperature at nodes.
26 - 35(F)		PRES	Pipe internal pressure at nodes.

F5. 4-NODE SURFACE PRESSURE PATTERNS

NPP4 sets of cards.

F5(a). CONTROL INFORMATION

One card.

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NCEL	No. of surface element cards for this pattern.
10(I)		KPRS	Pressure code: (a) 1: Nodal pressures input directly. (b) 2: Pressure from fluid depth. (c) 3: Uniform pressure.
11 - 15(I)		NCPR	No. of pressure cards (KPRS = 1 only).
16 - 25(F)			(a) KPRS = 2: Fluid density. (b) KPRS = 3: Uniform pressure value.
26 - 55			For KPRS = 2 only, direction cosines or projections of vertically upwards direction.
26 - 35(F)			X direction cosine, or projection on global X axis.
36 - 45(F)			Y.
46 - 55(F)			Z.
56 - 75			For KPRS = 2 only, coordinates of a point in the free surface.
56 - 60(F)			X coordinate.
61 - 65(F)			Y coordinate.
66 - 75(F)			Z coordinate.

F5(b). SURFACE ELEMENTS

NCEL cards. Enter nodes clockwise when viewed along outward normal.

COLUMNS	NOTE	NAME	DATA
1 - 5(I)	F.1	NOD1	Node 1.
6 - 10(I)		NOD2	Node 2.
11 - 15(I)		NOD3	Node 3.
16 - 20(I)		NOD4	Node 4.
21 - 25(I)		NPS	No. of elements. Default = 1.
26 - 30(I)		NDF	Node no. difference. Default = 1.

F5(c). NODAL PRESSURES

NCPR cards (KPRS = 1 only).

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NF	First node.
6 - 10(I)		NL	Last node. Blank for a single node.
11 - 15(I)		ND	Node no. difference. Default = 1.
16 - 25(F)			Pressure.

F6. NODAL DISPLACEMENT PATTERNS

NPND sets of cards.

F6(a). CONTROL INFORMATION

One card.

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NCND	No. of cards defining this pattern.
9 - 80(A)			Optional pattern title.

F6(b). NODAL DISPLACEMENTS

NCND cards.

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NF	Node, or first node in a series.
6 - 15(F)			X displacement.
16 - 25(F)			Y displacement.
26 - 35(F)			Z displacement.
36 - 45(F)			X rotation.
46 - 55(F)			Y rotation.
56 - 65(F)			Z rotation.
66 - 70(I)		NL	Last node in series. Blank for a single node.
71 - 75(I)		ND	Node no. difference. Default = 1.

G. DYNAMIC LOAD RECORDS

Omit entire Section G if IDAT = OLDF.

G1. CONTROL INFORMATION

One card.

COLUMNS	NOTE	NAME	DATA
1 - 5(I)	G.1	NGAR	No. of ground acceleration records (Section G2).
6 - 10(I)	G.1	NFHR	No. of dynamic force records (Section G3).
11 - 15(I)	G.1	NSDR	No. of dynamic displacement records (Section G4).
16 - 20(I)	G.2	NFHN	Max. no. of nodes with dynamic forces (must not be less than total no. of nodes specified in Section K7 or L6).
21 - 25(I)	G.2	NSDN	Max. no. of nodes with dynamic support displacements (must not be less than total no. of nodes specified in Section K8 or L7).
26 - 30(I)	G.2	MAXIPT	Max. no. of input values for any dynamic record (see NIPT, Sections G2(b), G3(b), G4(b)).

G2. GROUND ACCELERATION RECORDS

NGAR sets of cards.

G2(a). RECORD TITLE

COLUMNS	NOTE	NAME	DATA
1 - 60(A)			Optional title.
61 - 80(A)	G.3	IFORM	Input format. No default.

G2(b). CONTROL CARD

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NIPT	No. of acceleration values. Max. = MAXIPT, Section G1.
10(I)			Print code: (a) 0: No print. (b) 1: Print as input and scaled.
11 - 20(F)	G.3		Input time interval: (a) Zero or blank: Input time - acceleration pairs. (b) Nonzero: Input accelerations only.
26 - 30(F)	G.4		Acceleration scale factor. Default = 1.0.

G2(c). ACCELERATION VALUES

As many cards as needed to define NIPT acceleration values, with format IFORM. If time-acceleration pairs are input, time must immediately precede corresponding acceleration.

G3. DYNAMIC FORCE RECORDS

NFHR sets of cards.

G3(a). RECORD TITLE

COLUMNS	NOTE	NAME	DATA
1 - 60(A)			Optional title.
61 - 80(A)	G.3	IFORM	Input format. No default.

G3(b). CONTROL CARD

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NIPT	No. of force values. Max. = MAXIPT, Section G1.
10(I)			Print code: (a) 0: No print. (b) 1: Print as input and scaled.
11 - 20(F)	G.3		Input time interval: (a) Zero or blank: Input time - force pairs. (b) Nonzero: Input forces only.
21 - 30(F)			Force scale factor. Default = 1.0.

G3(c). FORCE VALUES

As many cards as needed to define NIPT force values, with format IFORM. If time-force pairs are input, time must immediately precede corresponding force.

G4. DYNAMIC DISPLACEMENT RECORDS

NSDR sets of cards.

G4(a). RECORD TITLE

COLUMNS	NOTE	NAME	DATA
1 - 60(A)			Optional title.
61 - 80(A)	G.3	IFORM	Input format. No default.

G4(b). CONTROL CARD

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NIPT	No. of displacement values.
10(I)			Print code: (a) 0: No print. (b) 1: Print as input and scaled.
11 - 20(F)	G.3		Input time interval: (a) Zero or blank: Input displacement pairs. (b) Nonzero: Input displacements only.
21 - 30(F)			Displacement scale factor. Default = 1.0.

G4(c). MOTION VALUES

As many cards as needed to define NIPT displacement values, with format IFORM. If time-displacement pairs are input, time must immediately precede corresponding displacement.

H. OUTPUT SPECIFICATION

H1. CONTROL CARD

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NODSX	No. of nodes for X-displacement printout. Input -1 for all nodes. For dynamic analysis, includes velocities and accelerations.
6 - 10(I)		NODSY	No. of nodes for Y-displacement printout. Input -1 for all nodes.
11 - 15(I)		NODSZ	No. of nodes for Z-displacement printout. Input -1 for all nodes.

H2. NODE LISTS

Three sets of cards.

(1) As many cards as needed. List NODSX nodes in successive 5-column fields (integer format), sixteen nodes to a card. Omit for NODSX = 0 or -1.

(2) As many cards as needed. List NODSY nodes in successive 5-column fields (integer format), sixteen nodes to a card. Omit for NODSY = 0 or -1.

(3) As many cards as needed. List NODSZ nodes in successive 5-column fields (integer format), sixteen nodes to a card. Omit for NODSZ = 0 or -1.

I. ANALYSIS TYPE

COLUMNS	NOTE	NAME	DATA
1 - 4(A)	I.1	IANTYP	Analysis type code: (a) STAT : Static analysis. (b) DYN1 : Dynamic analysis using event-to-event scheme. (c) DYN2 : Dynamic analysis using iteration scheme. (d) VISC : Visco-static analysis. (e) CREP: Creep analysis. (f) NONE: End execution. Default = NONE
9 - 80(A)			Optional analysis heading.

J. STATIC ANALYSIS

Omit entire Section J if IANTYP ≠ STAT.

J1. TITLE

COLUMNS	NOTE	NAME	DATA
1 - 5(I)	J.1	NSAS	No. of analysis segments
9 - 80(A)			Optional analysis title.

J2. SEGMENT SPECIFICATION

NSAS sets of cards, one set for each analysis segment.

J2(a). SOLUTION CONTROL

COLUMNS	NOTE	NAME	DATA
5(I)	J.2	KTYS	Step type: (a) 1: Load stepping. (b) 2: Displacement stepping.
10(I)	J.3	KPAS	Stepping parameter (KTYS = 2 only): (a) 1: Euclidean norm. (b) 2: Maximum value. (c) 3: Specified stepping vector(s)
11 - 20(F)		DISI	Segment value (KPAS = 1,2 only).
25(I)		NVES	No. of stepping vectors (KPAS = 3 only). Max = 4. Default = 1.
30(I)		KDIR	Step direction code: (a) 1: Increase load magnitude each step. (b) 2: Increase scalar displacement using (critical) stepping vector (KPAS = 3 only). (c) 3: Increase scalar displacement using direction vector. (d) 4: Determine sign of load increment using Bergan's current stiffness parameter.
31 - 40(F)		STEPI	Initial step factor for segment. Must be between 0.0 and 1.0.
41 - 45(I)		MAXST	Maximum no. of steps. Analysis stops if this no. is exceeded. Default = no limit.
46 - 50(I)		IEVENT	Element event code: (a) 0: No event check. (b) 1: Event check, no limit on number of events. (c) N(>1): Check events, up to N events in any step.
55(I)		KTYI	Iteration type: (a) 1: Constant load. (b) 2: Constant scalar displacement.

60(I)	KPAI	Iteration parameter (KTYI = 2 only). (a) 1: Constant work. (b) 2: Constant "arc length". (c) 3: Critical stepping vector (KPAS must = 3). (d) 4: Iteration vector.
65(I)	MAXIT	Max. no. of iterations per step.
70(I)	IQUIT	Termination code: (a) 0: If MAXIT is exceeded, restep. Include card J2(e). (b) 1: If MAXIT is exceeded, stop. (c) 2: If MAXIT is exceeded, continue.

J2(b). SPECIFIED VECTORS

NVES sets of cards defining the stepping vectors if KPAS = 3. Plus one set of cards defining the direction vector if KDIR = 3. Plus one set of cards defining the iteration vector if KPAI = 4.

Each set consists of a vector title card plus one or more cards defining the terms of the vector. The number of terms for any vector must not exceed 20.

J2(b) (i) VECTOR TITLE

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NDGC	No. of terms in this vector (max. = 20).
6 - 15(F)		FACI	Segment value (stepping vectors only).
16 - 80(A)			Optional title.

J2(b) (ii) VECTOR DEFINITION

As many cards as needed to specify NDGC terms, 4 terms per card.

COLUMNS	NOTE	NAME	DATA
1 - 5(I)			Node no.
10(I)			D.O.F. number (1-6).
11 - 20(F)			Weighting factor. Default = 1.0.
21 - 40			Similar data for second term (2I5,F10.0).
41 - 60			Similar data for third term (2I5,F10.0).
61 - 80			Similar data for fourth term (2I5,F10.0).

J2(c). STIFFNESS, STATE AND OUTPUT CONTROL

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		KUFS	Step control for stiffness update. Stiffness is updated every KUFS steps. Default = 1.
6 - 10(I)		KUFI	Iteration control for stiffness update. Stiffness is updated every KUFI iterations. Default = 1.
15(I)		KBFGS	BFGS code: (a) 0: No BFGS. (b) 1: BFGS between stiffness updates.
16 - 20(I)		KUFSD	Iteration control for state update. State is updated every KUFSD iterations. State is automatically updated at the end of every step.
25(I)		KPRIN	Results print code: (a) 0: No printed output. (b) 1: Print at end of analysis segment only. (c) 2: Print every step. (d) 3: Print every iteration.
30(I)		KSAVE	Results save code: (a) 0: Do not save on restart file. (b) 1: Save on restart file (at the end of the analysis segment).

J2(d). CONVERGENCE CRITERIA

COLUMNS	NOTE	NAME	DATA
5(I)		INRM	Unbalance type: (a) 0: Max. value. (b) 1: Euclidean norm.
10(I)		MOCD	Moment code for unbalance: (a) 0: Ignore moments. (b) 1: Include moments.
11 - 20(F)		TOL	Unbalance tolerance, for all except last step of segment.
21 - 30(F)		TOLF	Unbalance tolerance for last step. Default = TOL.
31 - 40(F)		UNBLS	Unbalance for step size scaling (must be greater than TOL). (a) 0.0: No scaling. (b) >0.0: Linear scaling of step size until unbalance is less than UNBLS.
41 - 45(I)		MAXLS	Maximum no. of line searches for any iteration. Default = none. (Leave rest of card blank.)
46 - 55(F)		UPLS	Upper line search limit (multiple of calculated displacement). Must be > 1.0.
56 - 65(F)		BMLS	Lower line search limit (multiple of calculated displacement). Must be between 0.0 and 1.0.
66 - 75(F)		TOLS	Convergence tolerance for line search (multiple of initial unbalance for the iteration). Must be between 0.0 and 1.0.

J2(e). RESTEPPING CRITERIA (omit if IQUIT not 0)

COLUMNS	NOTE	NAME	DATA
1 - 10(F)		DNFAC	Multiplier for step factor reduction. If MAXIT is exceeded step factor is multiplied by DNFAC. Default = 0.25.
11 - 20(F)		STPMIN	Minimum allowable step factor, as a multiple of the initial step factor. Default = 1/64 = .0156.
21 - 25(I)		LQUIT	Termination code: (a) 0: If STPMIN is reached, continue with step factor STPMIN. (b) N: If more than N attempts are made to reduce the step factor below STPMIN, stop the analysis.
26 - 30(I)		MINIT	Lower limit on the number of iterations for step factor increase. If convergence is obtained in MINIT or fewer iterations, multiply step factor by UPFAC.
31 - 40(F)		UPFAC	Multiplier for step factor increase. Default = 4.
41 - 50(F)		STPMAX	Maximum allowable step factor as a multiple of the initial step factor. Default = 1.
55(I)		ITPR	Convergence prediction code: (a) 0: No prediction. (b) N: Start prediction at iteration N.
56 - 65(F)		GRACE	Grace ratio. If predicted no. of additional iterations for convergence is less than (GRACE)*(current iteration no.) then iteration continues even if MAXIT is exceeded. Default = 0.5.
66 - 75(F)		WTFAC	Weighting factor for iteration prediction. The convergence ratio of the current iteration is weighted 1.0, at the previous iteration 1.0/WTFAC, etc. Default = 1.0.

J2(f). STATIC LOAD PATTERN APPLICATION

The total number of static load patterns is

NPAT = NPNF + NPFF + NPTP + NPP4 + NPND

from Card F1. Static load patterns are numbered in the sequence that they are input.

Use as many cards as needed to specify NPAT scale factors (one per pattern), 8 per card in 8F10.0 format. To omit any pattern, specify a zero or blank scale factor.

K. DYNAMIC ANALYSIS USING DYN1 OPTION.

Omit entire Section K if IANTYP ≠ DYN1.

K1. ANALYSIS CONTROL

COLUMNS	NOTE	NAME	DATA
5(I)	K.1	KTYP	Solution type: (a) 1: Event-to-event with variable time step. (b) 2: Event-to-event with constant time step. (c) 3: Variable time step with no event calculation.
6 - 10(I)	K.2	ICOR	Equilibrium correction code: (a) 0: Adjust accelerator at end of time step to eliminate equilibrium unbalance. (b) 1: Apply correcting forces in next time step to eliminate unbalance.
11 - 20(F)		DT	Time step (initial step for KTYP = 1 or 3).
21 - 30(F)	K.3	ALPHA	Parameter α for integration scheme (< 0; default = 0.).
31 - 40(F)	K.3	BETA	Parameter β for integration scheme. Default = $0.25\ (1 - \alpha)^2$
41 - 50(F)	K.4	DAMPM	Mass proportional damping factor, α_m.
44 - 45(I)	K.5	NICGC	Initial conditions code: (a) -1: Zero initial velocities and accelerations. (b) 0: If first analysis in this computer run, conditions of restart data sequence NRSEQ, Section B. If second or subsequent analysis, conditions at end of preceding analysis. (c) >0: Initial velocities and accelerations specified (Section K5).
60(I)	K.6	IGM	Ground acceleration code: (a) 0: Ground accelerations not applied. (b) 1: Ground accelerations applied (Section K6).

61 - 65(I)	K.6	NDLGC	No. of dynamic force commands: (a) 0: Forces not applied. (b) >0: Forces applied (Section K7).
66 - 70(I)	K.6	NSDGC	No. of dynamic support displacement commands: (a) 0: Support displacements not applied. (b) >0: Support displacements applied (Section K8).

K2. TIME STEP SELECTION

Omit for KTYP = 2.

COLUMNS	NOTE	NAME	DATA
1 - 10(F)	K.7	DTMIN	Minimum allowable step size: (a) Positive : Continue analysis using DTMIN. (b) Negative: Stop analysis if time step is reduced below DTMIN.
11 - 20(F)	K.7	DTMAX	Maximum allowable step size.
21 - 30(F)	K.7	CRED	Step size reduction factor (<1, default = 0.5).
31 - 40(F)	K.7	CINC	Step size increase factor (>1, default = 2.).
41 - 50(F)	K.7		Upper tolerance on midstep error.
51 - 60(F)	K.7		Lower tolerance on midstep error.
65(I)	K.7	IMOM	Moment code for midstep error: (a) 0: Ignore moments. (b) 1: Include moments.

K3. TERMINATION CONTROL

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NSTEP	Max. no. of time steps for this analysis.
6 - 15(F)		TINC	Max. time increment for this analysis. The analysis stops when the number of time steps reaches NSTEP or the sum of the steps reaches TINC, whichever occurs first.
16 - 25(F)		TOL	Max. allowable unbalanced force. If exceeded, analysis stops. Default = no limit.
26 - 35(F)		DISMAX	Max. allowable displacement. If exceeded, analysis stops. Default = no limit.
36 - 45(F)		DISLIM	Max. displacement increment for event control (for KTYP = 1 or 2 only). Default = no limit.
46 - 50(I)		MAXSUB	Max. allowable number of substeps in any time step (for KTYP = 1 or 2 only). Default = 10: (a) Positive : Continue analysis if MAXSUB exceeded. (b) Negative : Stop analysis if MAXSUB exceeded.

K4. OUTPUT CONTROL

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NITHJ	Time step interval for nodal response printout. Default = very large (does not control).
6 - 15(F)		TITHJ	Time interval for nodal response printout. The response is printed every NITHJ steps or TITHJ seconds, whichever occurs first. Default = very large.
16 - 20(I)		NITHE	Time step interval for element response printout. Default = very large.
21 - 30(F)		TITHE	Time interval for element response printout. Default = very large.
31 - 35(I)		NIENV	Time step interval for envelope printout. Default = NSTEP (print at end of analysis only).
36 - 45(F)		TIENV	Time interval for envelope printout. The envelopes are printed every NIENV steps or TIENV seconds whichever occurs first. Default = TINC (print at end of analysis only).
46 - 50(I)		NISAVE	Time step interval for saving results on restart file. Default = very large (does not control).
51 - 60(F)		TISAVE	Time interval for saving results on restart file. The results are saved every NISAVE steps or TISAVE seconds, whichever occurs first. Default = very large.

K5. INITIAL CONDITION SPECIFICATION

NIDGC cards. Omit if NICGC = 0 or -1.

COLUMNS	NOTE	NAME	DATA
5(I)		KDOF	D.O.F. code: (a) 1, 2, or 3 = X, Y, or Z translation, respectively. (b) 4, 5, or 6 = X, Y, or Z rotation, respectively.
6 - 15(F)			Velocity.
16 - 25(F)			Acceleration.
26 - 80(I)			List nodes with the same initial velocity and acceleration, in successive 5-column fields.

K6. GROUND ACCELERATION APPLICATION

Omit if IGM = 0.

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NR	Ground acceleration record no. for X direction. Default = none.
6 - 15(F)	K.8	TR	Time in record at time step zero.
16 - 25(F)		SFAC	Acceleration scale factor. No default.
26 - 50			Repeat for Y direction (I5,2F10.0).
51 - 75			Repeat for Z direction (I5,2F10.0).

K7. DYNAMIC FORCE APPLICATION

NDLGC cards.

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NR	Dynamic force record no.
10(I)		KDIR	Direction code: (a) 1, 2, or 3 = X, Y, or Z translation, respectively. (b) 4, 5, or 6 = X, Y, or Z rotation, respectively.
11 - 20(F)	K.3	TR	Time in record at time step zero. May be negative (time delay).
21 - 30(F)		SFAC	Force scale factor. No default.
31 - 80(I)		NODL	List up to 10 nodes in successive 5-column fields. Total nodes must not exceed NFHN, Section G.1.

K8. SUPPORT DISPLACEMENT APPLICATION

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NR	Support displacement record no.
10(I)		KDIR	Direction code: (a) 1, 2, or 3 = X, Y, or Z translation, respectively. (b) 4, 5, or 6 = X, Y, or Z rotation, respectively.
11 - 20(F)	K.8	TR	Time in record at time step zero. May be negative (time delay).
21 - 30		SFAC	Displacement scale factor. No default.
31 - 80(I)		NODL	List up to 10 nodes in successive 5-column fields. Total nodes must not exceed NSDN, Section G.1.

L. DYNAMIC ANALYSIS USING DYN2 OPTION

Omit entire Section L if IANTYP ≠ DYN2.

L1. CONTROL INFORMATION

COLUMNS	NOTE	NAME	DATA
1 - 10(F)		DT	Time step.
11 - 20(F)	K.3	ALPHA	Parameter α for integration scheme (< 0, default = 0).
21 - 30(F)	K.3	BETA	Parameter β for integration scheme. Default = $0.25(1 - \alpha)^2$.
31 - 40(F)	K.4	DAMPM	Mass proportional damping factor, α_m.

L2. ANALYSIS CONTROL

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NSTEP	No. of time steps for this analysis.
6 - 10(I)	K.5	NICGC	Initial conditions code: (a) -1: Zero initial velocities and accelerations. (b) 0: If first analysis in this computer run, conditions of restart data sequence NRSEQ, Section B. If second or subsequent analysis, conditions at end of preceding analysis. (c) >0: Initial velocities and accelerations specified (Section K4).
15(I)	K.6	IGM	Ground acceleration code: (a) 0: Ground accelerations not applied. (b) 1: Ground accelerations applied (Section L5).
16 - 20(I)	K.6	NDLGC	No. of dynamic force commands: (a) 0: Forces not applied. (b) >0: Forces applied (Section L6).
21 - 25(I)	K.6	NSDGC	No. of dynamic support displacement commands: (a) 0: Support displacements not applied. (b) >0: Support displacements applied (Section L7).
26 - 30(I)		NITHJ	Time step interval for nodal response printout. Default = no printout.
31 - 35(I)		NITHE	Time step interval for element response printout. Default = no printout.
36 - 40(I)		NIENV	Time step interval for envelope printout. Default = no printout.
41 - 45(I)		KSAVE	Time step interval for saving results on restart file. Default = no saving.

L3. ITERATION PROCEDURE

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		MAXIT	Maximum no. of iterations in any time step. Default = 1.
6 - 10(I)		KSUF	Time step control for stiffness update. Stiffness is updated every KSUF time steps. Default = 1.
11 - 15(I)		KIUF	Iteration control for stiffness update. Stiffness is updated every KIUF iterations. Default = 1.
16 - 20(I)		KSDUF	Iteration control for state update. State is updated every KSDUF iterations. Default = 1. State is updated at the end of each time step, regardless of the value of KSDUF.
25(I)		IQUIT	Solution termination code. (a) 0: Continue with next time step if no convergence. (b) 1: Stop analysis if no convergence.
26 - 35(F)		TOL	Unbalance tolerance.
40(I)			Unbalance type: (a) 0: Maximum value. (b) 1: Euclidean norm.
45(I)			Moment code for unbalance: (a) 0: Ignore moments. (b) 1: Include moments.

L4. INITIAL CONDITION SPECIFICATION

NIDGC cards. Omit if NICGC = 0 or -1.

COLUMNS	NOTE	NAME	DATA
5(I)		KDOF	D.O.F. code: (a) 1, 2, or 3 = X, Y, or Z translation, respectively. (b) 4, 5, or 6 = X, Y, or Z rotation, respectively.
6 - 15(F)			Velocity.
16 - 25(F)			Acceleration.
26 - 80(I)			List nodes with the same initial velocity and acceleration, in successive 5-column fields.

L5. GROUND ACCELERATION APPLICATION

Omit if IGM = 0.

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NR	Ground acceleration record no. for X direction. Default = none.
6 - 15(F)	K.8	TR	Time in record at time step zero.
16 - 25(F)		SFAC	Acceleration scale factor. No default.
26 - 50			Repeat for Y direction (I5,2F10.0).
51 - 75			Repeat for Z direction (I5,2F10.0).

L6. DYNAMIC FORCE APPLICATION

NDLGC cards.

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NR	Dynamic force record no.
10(I)		KDIR	Direction code: (a) 1, 2, or 3 = X, Y, or Z translation, respectively. (b) 4, 5, or 6 = X, Y, or Z rotation, respectively.
11 - 20(F)	K.8	TR	Time in record at time step zero. May be negative (time delay).
21 - 30(F)		SFAC	Force scale factor. No default.
31 - 80(I)		NODL	List up to 10 nodes in successive 5-column fields. Total nodes must not exceed NFHN, Section G.1.

L7. SUPPORT DISPLACEMENT APPLICATION

NSDGC cards.

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NR	Support displacement record no.
10(I)		KDIR	Direction code: (a) 1, 2, or 3 = X, Y, or Z translation, respectively. (b) 4, 5, or 6 = X, Y, or Z rotation, respectively.
11 - 20(F)	K.8	TR	Time in record at time step zero. May be negative (time delay).
21 - 30(F)		SFAC	Displacement scale factor. No default.
31 - 80(I)		NODL	List up to 10 nodes in successive 5-column fields. Total nodes must not exceed NSDN, Section G.1.

M. VISCO-STATIC ANALYSIS

Omit entire Section M if IANTYP ≠ VISC

M1. TITLE

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NSLI	No. of load increments.
9 - 80(A)			Optional analysis title.

M2. ANALYSIS CONTROL

COLUMNS	NOTE	NAME	DATA
4 - 5(I)		NICGC	Initial conditions code: (a) -1: Zero initial velocities and accelerations. (b) 0: If first analysis in this computer run, conditions of restart data sequence NRSEQ, Section B. If second or subsequent analysis, conditions at end of preceding analysis. (c) >0: Initial velocities and accelerations specified (Section M3).
6 - 15(F)		DT	Time step.
16 - 25(F)		THET	Parameter in integration method: (a) 1.0: Backwards difference rule. (b) 0.5: Trapezoidal rule. Default = 1.0

M3. INITIAL CONDITIONS

NICGC cards. Omit if NICGC = 0 or -1.

COLUMNS	NOTE	NAME	DATA
5(I)		KDOF	DOF code: (a) 1, 2, or 3 = X, Y, or Z translation, respectively. (b) 4, 5, or 6 = X, Y, or Z rotation, respectively.
6 - 15(F)			Velocity.
16 - 25(F)			Acceleration.
26 - 80(I)			List nodes with the same initial velocity and acceleration, in successive 5-column fields.

M4. LOAD INCREMENT

NSLI sets of cards.

M4(a). CONTROL CARD

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NSTEP	No. of equal load steps: (a) Positive : Results envelopes not printed at end of increment. (b) Negative: Envelopes printed.
10(I)		KSAVE	Results saving code: (a) 0: Results not saved. (b) 1: Results at end of load increment saved on restart file.
11 - 15(I)		KLSUF	Load step control for stiffness update. Stiffness is updated every KLSUF load steps. Default = 1.
16 - 20(I)		KTSUF	Time step control for stiffness update. Stiffness is updated every KTSUF time steps. Default = 1.
21 - 25(I)		KIUF	Iteration control for stiffness update. Stiffness is updated every KIUF iterations. Defaut = 1.
26 - 30(I)		KSDUF	Iteration control for state update. State is updated every KSDUF iterations. Default = 1.
31 - 35(I)		KTSDUF	Time step control for state update. State is updated every KTSDUF time steps. Default = 1.
40(I)		IQUIT	Solution termination code for load step: (a) 0: Continue with next load step if no convergence. (b) 1: Stop analysis if no convergence.
45(I)		IQUITV	Solution termination code for time step: (a) 0: Continue with next time step if no convergence. (b) 1: Stop analysis if no convergence.
46 - 50(I)		MAXSTP	Max. no. of time steps per load step.
51 - 55(I)		MAXIT	Max. no. of iterations per time step.

56 - 60(I)	NILSJ	Load step interval for nodal response printout. Default = very large.
61 - 65(I)	NILSE	Load step interval for element response printout. Default = very large.
70(I)	IPRNT	Print code within load steps: (a) 0: No printout. (b) 1: Print at every iteration.

M4(b). CONVERGENCE TOLERANCES

COLUMNS	NOTE	NAME	DATA
1 - 10(F)		TOLF	Total unbalance tolerance in last step of load increment.
11 - 20(F)		TOL	Total unbalance tolerance in all except last step.
21 - 30(F)		TOLFV	Non-viscous unbalance tolerance in last step of load increment.
31 - 40(F)		TOLV	Non-viscous unbalance tolerance in all except last step.
41 - 50(F)		TOLS	Unbalance tolerance to switch to static analysis.
55(I)			Unbalance type: (a) 0: Euclidean norm. (b) 1: Maximum value.
60(I)			Moment code for unbalance: (a) 0: Ignore moments. (b) 1: Include moments

M4(c). STATIC FORCE PATTERN APPLICATION

Omit for NSPAT = 0 (Section F.1). Otherwise, as many cards as needed to specify NSPAT scale factors. For each pattern in turn, specify a scale factor. 10-column (F10.0) fields, eight to a card.

M4(d). STATIC DISPLACEMENT PATTERN APPLICATION

Omit for NPDP = 0 (Section F.1). Otherwise, as many cards as needed to specify NPDP scale factors. For each pattern in turn, specify a scale factor. 10-column (F10.0) fields, eight to a card.

USER GUIDE NOTES

NOTE B.1

When NEWF is specified, a new restart file is created. The structure geometry data and the load data are stored, and zero initial conditions (corresponding to the unstressed structure) are automatically set up. During the computer run, analysis results may be saved at certain stages in the analysis (at the end of any static load segment, and at specified time step intervals in dynamic analysis). The saved analysis results sets are numbered sequentially. Set number zero is the zero initial conditions. The computer printout includes a log of the results sets saved on the restart file.

For subsequent computer runs, the OLDF option may be used to recall the geometry and load data, and any results set number (NRSEQ). These results are used as initial conditions for continued analysis. Additional results may be saved during the analysis. These results are added to those previously saved, and are numbered in continuing sequence.

NOTE B.2

For the NEWF option, the user must provide a restart file identifier. This identifier is stored as the first record on the restart file. Subsequent uses of the restart file (OLDF) must specify the identical identifier, otherwise execution will be terminated. The two identifiers must match character for character, including blanks.

NOTE C.1

The program includes in-core and out-of-core, symmetric and unsymmetric equation solvers. The structure tangent stiffness matrix is stored columnwise in compacted form omitting most zero elements ("active column" form). A substantial amount of element data must also be stored.

As the structure becomes nonlinear, the tangent stiffness matrix progressively changes. Two stiffness matrices are stored, one in the "triangularized" form following Crout decomposition, and a "duplicate," which is not triangularized. As the structure becomes nonlinear, a complete new stiffness is not assembled. Instead, modifications are made to the duplicate stiffness, and these changes are then incorporated into the triangularized form. For structures with local nonlinearities, this can be substantially more economical than forming and triangularizing a complete new stiffness.

For small structures, or for computers with large real or virtual core capacity, it is recommended that the triangularized stiffness, duplicate stiffness and all element data be held in core (CORE option). For large structures, or for computers with small core capacity, it may be necessary to store the data out of core (DISK option). In this case, the element data is stored on a sequential scratch file, one record per element, and the stiffness matrices may be held entirely in core, or may be stored on sequential scratch files in blocks. If sufficient core is available, there will be only one stiffness matrix block, and the matrices will be stored in core. If not, the matrices will be blocked and stored out of core. For the DISK option, either all available core (i.e., that remaining after other data is stored) may be allocated for storage of the stiffness matrices, or else the amount of core to be allocated may be specified. In this second case, the total core requirement can be held below the full machine capacity. Note that if the OLDP option is used, the storage allocation may be changed for each new computer run.

Execution times can be expected to be substantially larger with the DISK option than with the CORE option, because of much larger I/O costs.

The number of stiffness matrix blocks is currently limited to 20. If the number of blocks exceeds the value, execution ceases.

NOTE C.2

The finite elements must be divided into "groups." All elements in any group must be of the same type (e.g., truss, 3-D solid), and, typically, all elements of a single type will be included in a single group. However, elements of the same type may be divided into separate groups if desired. There is no limit on the number of element groups.

The first element within any group may be given any number, not necessarily 1. The remaining elements in the group must then be numbered sequentially.

NOTE D.1

The "control node" coordinates must be defined with respect to an orthogonal, right-handed coordinate system X, Y, and Z. The coordinates of the remaining nodes may be generated. Only a straight line generation option is currently available (Section D2).

NOTE D.2

Each generation command can generate the coordinates of one or more nodes. The coordinates of the nodes at the beginning and end of the generation lines must have been previously defined, either by direct specification of by a previous straight line generation.

The generated nodes are spaced along the generation line as follows:

(a) SPAC = 0.0: The generated nodes are uniformly spaced between the two end nodes.

(b) SPAC < 1.0: The spacing between generated nodes is this proportion of the length of the generation line. Note that this can allow generated nodes to lie beyond node NE if desired.

(c) SPAC > 1.0: The spacing between generated nodes is this distance. Again, generated nodes may lie beyond node NE.

The program also incorporates a default option for nodes which are not otherwise defined. It is not necessary to provide generation commands for nodes which are (a) sequentially numbered between the beginning and end nodes of any straight line, and (b) equally spaced along that line. After all generation commands have been executed, the coordinates for each group of unspecified nodes are automatically generated, assuming sequential numbering and equal spacing along the line joining the specified nodes immediately preceding and following the group. That is, any generation command with SPAC = 0.0 and a node number difference of one is superfluous.

NOTE D.3

The displacements at any node may be specified to be in a rotated coordinate system rather than the global system. This allows, for example, inclined boundary conditions. The orientations of the rotated coordinate systems are specified by cards D3. The nodes are associated with particular coordinate systems by cards D4. The default is the global X,Y,Z system.

Rotated coordinate systems are defined by specifying the direction cosine of two of the rotated axes with respect to the global X,Y,Z axes. The two rotated axes defined in this way must be perpendicular to each other. The third rotated axis is determined automatically.

NOTE D.4

Each node of the structure may have up to six displacement degrees of freedom, namely, X, Y, and Z displacement, and X, Y, and Z rotation. However, some displacements may be constrained to be zero, and may be deleted (constraint code = 1). It is important to note that all degrees of freedom at which **nonzero** static or dynamic displacements are to be imposed must be "free" (code = 0).

If constraint codes are specified more than once for any node, the last specified value is used. Hence, if most but not all nodes have similar constraints, the first command may cover all nodes, and specify "basic" constraint codes. Later commands may then modify the "basic" codes for particular nodes.

For plane analysis, the out-of-plane displacement and in-plane rotations must all be deleted.

NOTE D.5

If nodes are not listed, the affected nodes are nodes N, N + ND, N + 2*ND, etc., up to node NE. If nodes are listed, NE and ND are ignored, and only the listed nodes are affected.

NOTE D.6

If may often be reasonable to assume that certain nodes displace equally in certain directions. The displacements (or rotations) can then be assigned the same degree of freedom number. Each equal displacement command covers at most 14 nodes. If a group of nodes has more than 14 nodes, specify the remaining nodes with additional commands. The smallest numbered node in any group must be the first node in the list for all commands defining that group.

Greater computational efficiency will often be achieved by constraining nodes to have equal displacements. However, the effect may be to increase the effective band width of the structure stiffness matrix, and may increase the required stiffness matrix storage and/or the computational effort. Equal displacements should therefore be specified with caution.

NOTE D.7

Any node may, if desired, appear in more than one specification command. In such cases, the mass will be the sum of the values specified in the separate commands. If certain nodes are constrained to have equal displacements, the

mass associated with the affected degree of freedom will be the sum of the masses specified for the individual nodes. If a mass is specified for any degree of freedom that is deleted, the mass is ignored.

NOTE D.8

Each table defines the values of secondary dashpot properties (array CS(NCSS, NCST)), primary dashpot properties (array CP(NCPS, NCPT)) and primary spring properties (array KP(NKPS, NKPT)) as functions of stress and temperature. Numerical values for any given stress and temperature are obtained by either linear or logarithmic interpolation. Each row of each array contains the properties for one stress value, and each column the properties for one temperature value. For stresses or temperatures outside the range of the tables, the values for the nearest stress and/or temperature are used.

NOTE G.1

Any number of ground acceleration record, dynamic force records, and/or support motion records may be specified. Dynamic force records are applied as nodal loads, and support motion records as nodal displacements. Records can be applied singly or in combination. The combination factors are specified in Sections G2, G3, and G4. The nodes affected by the records are specified in Sections K7 and K8 or L6 and L7.

Ground accelerations are identified with the X, Y, and Z directions in Section K6. Although any number of ground acceleration records may be defined, a maximum of three may be applied at any time, one each in the X, Y, and/or Z directions.

NOTE G.2

The values of NFHN, NSDN and MAXIPT are required by the program for storage allocation. If unnecessarily large values are specified, some core storage will be wasted.

NOTE G.3

The input format for reading time-value pairs for any dynamic record (ground acceleration, dynamic force, or support motion) must be specified. No default format is assumed by the program. The format must be enclosed in parentheses, e.g., (8F10.0), without the word "FORMAT." If any record contains record values (acceleration, forces, or support motions) at constant time intervals, only the record values need to be input. The times of this case are automatically generated by the program as 0.0, TINT, 2*TINT, . . ., etc.

NOTE G.4

The scale factor may be used to convert from multiples of gravity to acceleration units, or to scale the record. A further scale factor may be specified in Section K6.

NOTE I.1

Static or viscostatic analysis includes prescribed static loads and/or displacements. Dynamic analysis includes ground accelerations, dynamic forces and/or support motions, applied singly or in combination. Any sequence of static or dynamic analyses may be performed in the same computer run. Execution is terminated with the NONE option (or a blank card).

NOTE J.1

Static loads and imposed displacements can be applied in a number of static load **segments.** Each segment is obtained by combining static force and/or imposed displacement **patterns.** Each segment can be applied in a number of **steps** that are either load-controlled or displacement-controlled. For each step the solution is typically found in a number of **iterations,** the iteration path depending on the solution strategy selected. The results at the end of any segment may be saved on the restart file.

A different solution strategy may be specified for each new load segment. The scale factors for the load and displacement patterns may also be different from one segment to the next.

NOTE K.1

The dynamic analysis using DYN1 option permits three different solution schemes, depending on the value of KTYP. For KTYP = 1, the event-to-event strategy is used with automatic time step selection; for KTYP = 2, the event-to-event strategy is used with a constant time step; and for KTYP = 3, event calculations are omitted but the time step may vary during the analysis. No iteration is performed within a time step. Any unbalanced force at the end of a step is eliminated by applying correcting forces in the next time steps or by adjusting the accelerations at the end of the step (see following Note).

NOTE K.2

If ICOR = 1, any unbalanced forces are applied as correcting forces during the next time step. If ICOR = 0, unbalanced forces are eliminated by adjusting the accelerations at the end of each time step. That is, if ICOR = 0 and the unbalanced force for a given nodal displacement, r_i, is R_{ui}, the acceleration r_i is increased by $\Delta r_i = R_{ui}/M_i$, where M_i = mass (translational or rotational) associated with r_i. This correction is possible only if M_i is nonzero. For displacements with $M_i = 0$, the accelerations are not modified, and the unbalanced forces are applied as corrections in the next time step.

NOTE K.3

The Newmark "$\beta = 1/4$" scheme (the trapezoidal or "constant average acceleration" scheme) is the default option (i.e., $\alpha = 0$; $\beta = 0.25$). For nonzero values of α ($-0.33 < \alpha < 0$) Hibbitt's form of the Hilber-Hughes-Taylor scheme is used, to introduce numerical damping. Modifications of these schemes can be obtained by specifying β to be other than $0.25(1 - \alpha)^2$. However, this is not recommended.

NOTE K.4

See Note K.6 for the damping assumption. The stiffness proportional damping factors, β_0 and β_T, are specified separately for each element group.

NOTE K.5

For any dynamic analysis, initial velocities and accelerations are required. The default option (NICGC = 0) is as follows:

(a) First analysis in this computer run (no preceding static or dynamic analyses): Conditions of restart data sequence NRSEQ; or, for no restart, zero initial velocity and acceleration.

(b) Second or subsequent analysis in this computer run: Final conditions of immediately preceding analysis. If this was a static analysis, the initial velocities and accelerations will all be zero.

The default option may be overridden by specifying NIGC = -1 or > 0, as indicated. This override applied to velocities and accelerations only. The nodal displacements and element states of stress are those from the preceding analysis (for restart, data sequence NRSEQ).

NOTE K.6

Ground motions may be specified using either ground acceleration specification or dynamic support motion specification.

In the ground acceleration case, a "conventional" seismic time-history analysis is performed, by integrating the equation

$$M.\ddot{dx}_r + C.\dot{dx}_r + K_T.dx_r = -M.\ddot{dx}_g$$

in which x_r = nodal displacements relative to ground, and x_g = ground

displacements. For analyses of this type, all support points must move in phase, and the calculated displacements, velocities, and accelerations are values relative to the ground.

In the dynamic support motion case, the analysis is performed by integrating the equation

$$M.d\ddot{x} + C.d\dot{x} + K_T.dx = K_T.dx_g$$

in which x = total nodal displacements and x_g = ground displacements (at support points). The support points need not move in phase, and the calculated displacements, velocities and accelerations are total values.

The program allows both ground accelerations and dynamic support displacements to be applied simultaneously. However, the calculated results will be inconsistent because relative and total values will be combined. Because total values are calculated for dynamic force application, it will also be inconsistent to combine ground accelerations with dynamic forces. A similar inconsistency can arise if a ground acceleration analysis is followed by a dynamic force or dynamic support motion analysis, because the final conditions from the first analysis may not be consistent with the required initial conditions for the second.

If only static plus "conventional" seismic analyses are being performed, either the ground acceleration option or the dynamic support motion option may be used. If both dynamic forces and seismic motions are to be considered, only the dynamic support motion option should be used. If out-of-phase seismic motions are to be considered, use of the dynamic support motion option is essential.

The amount of viscous damping may also depend on the type of analysis being performed. The damping matrix, C, is given by:

$$C = \alpha_m M + \beta_0 K_0 + \beta_T K_T$$

Because rigid body motion of a structure will be unrestrained, it follows that:

$$K.dx_r = K.dx$$

in which $K = K_0$ or K_r. Hence, if $\alpha_m = 0$, the damping forces will be the same for both the ground acceleration and support motion options. If $\alpha_m \neq 0$, however, the damping forces will differ. Physically, "$\alpha_m M$" damping implies the presence of mass-proportional damping links as follows:

(a) Ground acceleration option: links connecting nodes to points which move with the ground.

(b) Support motion option: links connecting nodes to points which are fixed in space.

NOTE K.7

If KTYP = 1 or 3, the time step, DT, will be varied automatically during the analyses. If the midstep error in any time step exceeds the upper tolerance, DT is multiplied by CRED and the time step is repeated. The value of DT is not allowed to be smaller than DTMIN. If the midstep error is smaller than the lower tolerance for two consecutive time steps, DT is multiplied by CINC for the next time step. The value of DT is not allowed to exceed DTMAX.

The midstep error is the maximum term in the midstep error vector. This vector includes both translational forces and moments, and hence its terms are not all in the same units. If IMOM = 0 (the recommended option) only translational forces are considered in selecting the maximum term. If IMOM = 1, both translational forces and moments are considered.

NOTE K.8

Any dynamic record (e.g., ground acceleration record, force record, or support displacement record) may be applied with or without a time delay. If a record is applied with no delay (TR $\geq$ 0.), time TR of the record corresponds to time zero of the analysis. If a record is applied with a delay

(TR < 0.), the record is not applied until the time of the analysis reaches TR; up to this time the value of the record is assumed to be zero.

A typical application will be seismic excitation of structures which are long in plan. Assume that the ground motion is a train of waves traveling from left to right (note that this is not correct for a true earthquake, but may be a reasonable assumption). The delay for the first (leftmost) support will be zero; the delay for the second support will be the wave travel time from the first support to the second; the delay for the third support will be the wave travel time from the first support to the third; etc.

Time delays will usually be needed if an analysis is stopped and restarted in the middle of a record, using the option TR > 0.

APPENDIX B

U-BAR RESTRAINT ELEMENT

APPENDIX B

U-BAR RESTRAINT ELEMENT

B1. CONTROL INFORMATION

Two cards.

B1(a). FIRST CARD

COLUMNS	NOTE	NAME	DATA
4 - 5(I)		NGR	Element type indicator. (= 5 to indicate that group consists of U-bar restraint elements.)
6 - 10(I)		NELS	Number of elements in group.
11 - 15(I)		MFST	Element number of first element in group. Default = 1.
16 - 25(F)		DKO	Initial stiffness damping factor, β_0.
26 - 35(F)		DKT	Tangent stiffness damping factor, β_T.
41 - 80(A)			Optional group heading.

B1(b). SECOND CARD

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NMAT	Number of different property types.
10(I)		JFIX	Connection code: (a) Zero or blank: Single node option. Node J assumed fixed. (b) 1: Two node option. Node J assumed not fixed.

B2. PROPERTY TYPES

NMAT sets of cards. Three cards per set.

B2(a). CONTROL INFORMATION

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		N	Type number, in sequence beginning with 1.
6 - 10(I)		NSEGG	Number of segments in force-extension relationship. Max. 6; Min. 2.
11 - 20(F)		DBB	Initial clearance.
21 - 30(F)		TOLL	Stiffness reformulation tolerance (radians). Suggested value = 0.1.

B2(b). STIFFNESSES

COLUMNS	NOTE	NAME	DATA
1 - 60(F)		SPSTT	Up to 6 fields, each 10 columns. Specify NSEGG stiffnesses.

B2(c). STRENGTHS

COLUMNS	NOTE	NAME	DATA
1 - 50(F)		FFF	Up to 5 fields, each 10 columns. Specify NSEGG-1 yield forces.

B3. ELEMENT DATA

NELS cards. Generation is not permitted.

COLUMNS	NOTE	NAME	DATA
1 - 5(I)		NEL	Element number.
6 - 10(I)		NODI	Node I.
11 - 15(I)		NODJ	Node J.
16 - 20(I)		IMBT	Property type number.
25(I)		KGEOM	Large displacement code: (a) Zero or blank: small displacements. (b) 1: large displacements.
30(I)		KOUT	Time history code: (a) Zero or blank: No output. (b) 1: Output required.
31 - 35(I)		NODK	Node K. Optional, to define normal direction.
36 - 45(F)		TOLY	Overshoot tolerance for yield or gap opening events.
46 - 55(F)		TOLU	Overshoot tolerance for unloading or gap closure events.

TABLE B.1

U-BAR RESTRAINT ELEMENT

```
      subroutine inel05 (njt,ndkod,x,y,z,kexec)
      implicit double precision (a-h,o-z)
c
c     ****************************************************************
c     subroutine to read, generate and print element data.
c     element type = 5. u bar restraint element, type b.
c     ****************************************************************
c
      common /infel / imem,kst,lm(6),nodi,nodj,nodk,kgeom,jfix,kout,iup,
     *                ilow,line,iup7,xi,yi,zi,xj,yj,zj,spst(7),uval(6),
     *                tol,flp,vtot,ftot,vlow,vup7,sst,vup,dcos(3),
     *                dcosp(3),dcosn(3),dism,tdis,fom,tfo,toly,tolu,xkp,
     *                skp(21)
      common /tapes / niu,nou,nt1,nt2,nt3,nt4,nt5,nt6,nt7,nt8,nt9,
     *                nt10,nt11,nt12,nt13,nt14,nt15,nt16
      common /infgr / ngr,nels,mfst,igrhed(10),ninfc,lstat,
     *                lstf,lstc,ndof,dko,dkt,eprop(4,250)
      common /precis/ zero,one
      common /work/ spstt(7,20),uvall(6,20),fff(5),toll(20),nseg(20),
     *              dbb(20)
c
      dimension x(njt),y(njt),z(njt),icom(146),ndkod(njt,6)
      dimension iyesno(2)
      equivalence(imem,icom(1))
c
      data iyesno/4h  no,4h yes/
c
c     ----------------------------------------------------------------
c
c     meanings of variables in common block /infel/
c
c     imem.........- element number
c     kst..........- stiffness update code
c     lm(6)........- location matrix
c     nodi,nodj,nodk- node number at i,j,k
c     xi,yi,zi.....- coordinates of node i
c     xj,yj,zj.....- coordinates of node j
c     spst(7)......- stiffnesses
c     uval(6)......- extension at limits of stiffnesses
c     kgeom........- geometric stiffness code
c     jfix.........- connection code
c     kout.........- response output code
c     tol..........- tolerance
c     flp..........- current member length
c     vtot.........- current extension
c     ftot.........- current force
c     vup..........- extension at upper limit of current segment
c     vlow.........- extension at lower limit of current segment
c     iup..........- next line if extension increases
c     ilow.........- next line if extension decreases
c     line.........- current line
c     iup7.........- next line = 7
c     vup7.........- extension at upper limit if line = 7
c     xkp..........- previous stiffness
c     sst..........- elastic stiffness for beta-0 damping (always zero)
c     skp(21)......- current stiffness matrix
```

TABLE B.1 (CONTINUED)

```
c      dcos(3).......- direction cosine
c      dcosp(3)......- previous direction cosine
c      dcosn(3)......- direction cosine of normal direction
c      dism..........- maximum extension
c      tdis..........- time at which maximum extension occurs
c      fom...........- maximum force
c      tfo...........- time at which maximum force occurs
c      toly..........- overshoot tolerance for yield event
c      tolu..........- overshoot tolerance for unloading event
c
c      ---------------------------------------------------------------
c
c      control variables
c
       ninfc = 146
       lstat = 102
       lstc  =   2
       lstf  =  42
       ndof=3
       kst=1
       if (mfst.le.0) mfst = 1
c
       do 100 i=3,ninfc
100    icom(i)=0
c
c      read and print element group information
c
       read(niu,1000)nmbt,jfix
1000   format(2i5)
       if (jfix.ne.0) ndof=6
       if(nmbt.le.0)nmbt=1
       write(nou,2000)ngr,nels,mfst,ndof,ninfc,dko,dkt,nmbt,jfix
2000   format(34h u-bar restraint elements (type b)////
     *        5x,26helement group indicator  =i3/
     *        5x,26hno. of elements          =i3/
     *        5x,26hno. of first element     =i3/
     *        5x,26hno. of degrees of freedom=i3/
     *        5x,26hlength of element infor. =i3/
     *        5x,26hdamping coeff. beta-o    =f9.5/
     *        5x,26hdamping coeff. beta-t    =f9.5/
     *        5x,26hno. of stiffness types   =i3/
     *        5x,26hconnection code          =i3)
c
c      stiffness properties
c
       write(nou,2010)
2010   format(////16h stiffness types//
     *        3x,4htype,2x,4h no.,2x,9hclearance,3x,5hevent/
     *        3x,4h no.,2x,4hsegm,2x,9h    db    ,3x,5h tol./)
c
       do 200 nt=1,nmbt
c
       read(niu,1010)n,nsegg,dbb(nt),toll(nt)
1010   format(2i5,2f10.0)
       write(nou,2020)nt,nsegg,dbb(nt),toll(nt)
2020   format(2i6,f11.3,f9.5)
```

TABLE B.1 (CONTINUED)

```
      toll(nt)=cos(toll(nt))
      if(nsegg.ge.2.and.nsegg.le.6) go to 150
      write(nou,2030)
2030  format(/25h error  in no. of segments/)
      kexec=1
      nsegg=6
c
150   nseg(nt)=nsegg
      read(niu,1020)(spstt(n,nt),n=1,6)
1020  format(6f10.0)
      write(nou,2040)(spstt(n,nt),n=1,nsegg)
2040  format(40x,11hstiffnesses,4x,6e12.4)
c
      spstt(7,nt)=0.
      do 160 n=1,nsegg
      spstt(7,nt)=amax1(spstt(7,nt),spstt(n,nt))
      if(spstt(n,nt).ne.zero)go to 160
      kexec=1
      write(nou,2050)
2050  format(/23h error - zero stiffness/)
160   continue
c
      nsegg=nsegg-1
      read(niu,1020)(fff(n),n=1,nsegg)
      write(nou,2060)(fff(n),n=1,nsegg)
2060  format(40x,12hyield forces,3x,5e12.4)
      ffp=0.
      uup=0.
      do 200 n=1,6
      if(n.gt.nsegg) go to 170
      uvall(n,nt)=uup+(fff(n)-ffp)/spstt(n,nt)
      uup=uvall(n,nt)
      ffp=fff(n)
      go to 200
170   uvall(n,nt)=1.e25
200   continue
c
c     element data
c
      write(nou,2070)
2070  format(////22h element specification//
     *      3x,4helem,3x,4hnode,2x,4hnode,3x,4hstif,3x,4hgeom,3x,4htime,
     *      3x,4hnode,4x,9hovershoot,4x,9hovershoot/
     *      3x,4h no.,3x,4h  i ,2x,4h  j ,3x,4htype,3x,4hcode,3x,4hhist,
     *      3x,4h  k ,4x,9hyield tol,4x,9hunldg tol/)
c
      nlast=mfst+nels-1
      do 300 imem=mfst,nlast
c
      read(niu,1030)n,nodi,nodj,imbt,kgeom,kout,nodk,toly,tolu
      if (kout.ne.0) kout=1
      if (kgeom.ne.0) kgeom=1
      write(nou,2080)imem,nodi,nodj,imbt,iyesno(kgeom+1),iyesno(kout+1),
     *               nodk,toly,tolu
1030  format(7i5,2f10.0)
2080  format(i6,i7,i6,i7,4x,a4,3x,a4,i6,2e15.4)
```

TABLE B.1 (CONTINUED)

```
c
c     location matrix
c
      do 210 i=1,3
      call ncodlm(lm(i),njt,nodi,i)
210   call ncodlm(lm(i+3),njt,nodj,i)
c
      call band(lm,ndof,ndkod)
c
c     element properties
c
      nsegc  = nseg(imbt)
      do 220 n=1,7
220   spst(n) = spstt(n,imbt)
      do 230 n=1,6
230   uval(n) = uvall(n,imbt)
      line = 8
      vtot = -dbb(imbt)
      ftot = 0.
      if (vtot.le.zero) go to 232
      line = 1
      vlow = vtot
      vup  = uval(1)
      ftot = vtot * spst(line)
      ilow = 7
      iup  = 2
      go to 233
232   vlow = -1.e25
      vup = 0.
      ilow = 8
      iup = 1
233   tol = toll(imbt)
c
c     direction cosines of axis
c
      xi = x(nodi)
      yi = y(nodi)
      zi = z(nodi)
      xj = x(nodj)
      yj = y(nodj)
      zj = z(nodj)
      dx = xi-xj
      dy = yi-yj
      dz = zi-zj
      flp = sqrt(dx**2+dy**2+dz**2)
      dcos(1) = dx/flp
      dcos(2) = dy/flp
      dcos(3) = dz/flp
c
      do 240 i=1,3
240   dcosp(i) = dcos(i)
c
c     direction cosines of normal
c
      if(nodk.eq.0)go to 280
      dx = x(nodk)-xi
```

TABLE B.1 (CONTINUED)

```
      dy = y(nodk)-yi
      dz = z(nodk)-zi
      ff = sqrt(dx**2+dy**2+dz**2)
      dcosn(1) = dx/ff
      dcosn(2) = dy/ff
      dcosn(3) = dz/ff
      ff = 0.
      do 250 n=1,3
250   ff=ff+dcosn(n)*dcos(n)
      ffp = 0.
      do 260 n=1,3
      dcosn(n) = dcosn(n)-ff*dcos(n)
260   ffp=ffp+dcosn(n)**2
      ffp = sqrt(ffp)
      do 270 n=1,3
270   dcosn(n) = dcosn(n)/ffp
c
c     compact element data
c
280   call compac
c
300   continue
c
      return
      end
```

TABLE B.1 (CONTINUED)

```
      subroutine stif05 (istep,ndf,cdko,cdkt,fk,indfk,istfc)
      implicit double precision (a-h,o-z)
c
c     ****************************************************************
c     subroutine to compute element tangent stiffness matrix.
c     element type = 5. u bar restraint element, type b.
c     ****************************************************************
c
      common /infel / imem,kst,lm(6),nodi,nodj,nodk,kgeom,jfix,kout,iup,
     *                ilow,line,iup7,xi,yi,zi,xj,yj,zj,spst(7),uval(6),
     *                tol,flp,vtot,ftot,vlow,vup7,sst,vup,dcos(3),
     *                dcosp(3),dcosn(3),dism,tdis,fom,tfo,toly,tolu,xkp,
     *                skp(21)
      common /infgr / ngr,nels,mfst,igrhed(10),ninfc,lstat,
     *                lstf,lstc,ndof,dko,dkt,eprop(4,250)
c
      dimension fk(ndf,ndf),s(6,6)
c
      indfk = 0
c
c     stiffness update
c
      xk = 0.
      if(line.ne.8)xk=spst(line)
      xkp = xk
c
c     stiffness for beta=0
c
      sst = 0.
      if(line.ne.8)sst=spst(1)
c
c     effective stiffness
c
      if(istep.gt.0)xk=sst*cdko+xk*(1.+cdkt)
c
c     form 6x6 element stiffness matrix
c
      do 100 i=1,3
      cc = dcos(i)*xk
      do 100 j=i,3
100   s(i,j) = cc*dcos(j)
      if(jfix.eq.0)go to 120
      do 110 i=1,3
      do 110 j=i,3
      s(i,j+3)   = -s(i,j)
      s(j,i+3)   =  s(i,j+3)
110   s(i+3,j+3) =  s(i,j)
120   continue
c
c     geometric stiffness
c
      if (kgeom.eq.0) go to 150
c
      pl = ftot/flp
      ptnine = 0.999999
      if (abs(dcos(2)).gt.ptnine) go to 140
```

TABLE B.1 (CONTINUED)

```
c
      b12 = dcos(1) * dcos(2)
      b23 = dcos(2) * dcos(3)
      denom = dcos(1)**2 + dcos(3)**2
c
      s(1,1) = s(1,1) + p1 * ( b12**2 + dcos(3)**2 )/denom
      s(1,2) = s(1,2) - p1 * b12
      s(1,3) = s(1,3) + p1 * ( b12 * b23 - dcos(1) * dcos(3) )/denom
      s(2,2) = s(2,2) + p1 * denom
      s(2,3) = s(2,3) - p1 * b23
      s(3,3) = s(3,3) + p1 * ( b23**2 + dcos(1)**2 )/denom
c
      if (jfix.eq.0) go to 150
      do 130 i=1,3
      do 130 j=i,3
      s(i,j+3)   = -s(i,j)
      s(j,i+3)   =  s(i,j+3)
130   s(i+3,j+3) =  s(i,j)
      go to 150
c
140   s(1,1) =  s(1,1) + p1
      s(3,3) =  s(3,3) + p1
      if (jfix.eq.0) go to 150
      s(4,4) =  s(1,1)
      s(6,6) =  s(3,3)
      s(1,4) = -s(1,1)
      s(3,6) = -s(3,3)
150   continue
c
c     compute change in stiffness and retain current stiffness
c
      if (istfc.eq.0) go to 170
c
      ij = 0
      do 160 i=1,ndof
      do 160 j=i,ndof
      ij       = ij + 1
      stif     =  s(i,j)
      fk (i,j) = stif
      fk (j,i) = stif
160   skp(ij)  = stif
      go to 190
c
170   ij = 0
      do 180 i=1,ndof
      do 180 j=i,ndof
      ij       = ij + 1
      stif     =  s(i,j)
      stifd    = stif - skp(ij)
      fk (i,j) = stifd
      fk (j,i) = stifd
180   skp(ij)  = stif
c
c     update element information
c
190   kst  = 0
```

TABLE B.1 (CONTINUED)

```
      do 200 i=1,3
200   dcosp(i)=dcos(i)
c
      return
      end
```

TABLE B.1 (CONTINUED)

```
      subroutine stat05 (ndf,q,time)
      implicit double precision(a-h,o-z)
c
c     ****************************************************************
c     subroutine for state determination calculations.
c     element type = 5. u bar restraint element, type b.
c     ****************************************************************
c
      common /infel / imem,kst,lm(6),nodi,nodj,nodk,kgeom,jfix,kout,iup,
     *                ilow,line,iup7,xi,yi,zi,xj,yj,zj,spst(7),uval(6),
     *                tol,flp,vtot,ftot,vlow,vup7,sst,vup,dcos(3),
     *                dcosp(3),dcosn(3),dism,tdis,fom,tfo,toly,tolu,xkp,
     *                skp(21)
      common /infgr / ngr,nels,mfst,igrhed(10),ninfc,lstat,
     *                lstf,lstc,ndof,dko,dkt,eprop(4,250)
      common /precis/ zero,one
c
      dimension q(ndf)
      dimension disi(3),disj(3)
c
c     set current stiffness
c
      xk = 0.
      if (line.ne.8) xk = spst(line)
c
c     displacements affecting extension
c
      do 100 i=1,3
100   disi(i)=q(i)
      if(jfix.eq.0)go to 120
      do 110 i=1,3
110   disj(i)=q(i+3)
      go to 140
120   do 130 i=1,3
130   disj(i)=0.
c
c     sweep out normal displacements
c
140   if(nodk.eq.0.or.line.ne.8)go to 190
      ff=0.
      do 150 i=1,3
150   ff=ff+dcosn(i)*disi(i)
      do 160 i=1,3
160   disi(i)=disi(i)-ff*dcosn(i)
      if(jfix.eq.0)go to 190
      ff=0.
      do 170 i=1,3
170   ff=ff+dcosn(i)*disj(i)
      do 180 i=1,3
180   disj(i)=disj(i)-ff*dcosn(i)
c
c     large displacements
c
190   if (kgeom.eq.0) go to 210
      xi=xi+disi(1)
      yi=yi+disi(2)
```

TABLE B.1 (CONTINUED)

```
      zi=zi+disi(3)
      if (jfix.eq.0) go to 200
      xj=xj+disj(1)
      yj=yj+disj(2)
      zj=zj+disj(3)
      dx=xi-xj
      dy=yi-yj
      dz=zi-zj
200   fl=sqrt(dx**2+dy**2+dz**2)
c
c     update direction cosine
c
      dcos(1)=dx/fl
      dcos(2)=dy/fl
      dcos(3)=dz/fl
      dv=fl-flp
      flp=fl
      go to 230
c
c     small displacements
c
210   dv=0.
      do 220 i=1,3
220   dv=dv+dcos(i)*(disi(i)-disj(i))
c
c
230   if (dv) 400,500,300
c
c     bar extends
c
300   fac=(vup-vtot)/dv
      if(fac.ge.one) go to 500
c
      dv=vtot+dv-vup
      ftot=ftot+(vup-vtot)*xk
      vtot=vup
      line=iup
      xk=spst(iup)
      vlow=vtot
      if(iup.eq.7) go to 330
      vup=uval(iup)
      ilow=7
      iup=iup+1
      go to 300
330   vup=vup7
      ilow=8
      iup=iup7
      go to 300
c
c     bar shortens
c
400   fac=(vlow-vtot)/dv
      if(fac.ge.one) go to 500
c
      dv=vtot+dv-vlow
      ftot=ftot+(vlow-vtot)*xk
```

TABLE B.1 (CONTINUED)

```
      vtot=vlow
      if(ilow.eq.8) go to 430
      xk=spst(7)
      vlow=vtot-ftot/xk
      vup=vtot
      vup7=vtot
      iup=line
      iup7=line
      line=7
      ilow=8
      go to 400
430   xk=0.
      vlow=-1.e25
      vup=vtot
      iup=7
      ilow=8
      line=8
      go to 400
c
c     complete cycle
c
500   vtot=vtot+dv
      if(line.lt.7) vlow=vtot
      ftot=ftot+xk*dv
c
c     check for stiffness change
c
      kst=0
      smalln = 1.e-6
      if(abs(xk-xkp).gt.smalln)kst=1
c
c     check rotation of axis - skip if line = 8
c
      if(line.eq.8)go to 600
      dv=0.
      do 550 i=1,3
550   dv=dv+dcos(i)*dcosp(i)
      if(dv.lt.tol)kst=1
c
c     accumulate envelopes
c
600   if (ftot.le.fom) go to 700
      fom=ftot
      tfo=time
700   if (vtot.le.dism) return
      dism=vtot
      tdis=time
c
      return
      end
```

TABLE B.1 (CONTINUED)

```
      subroutine evnt05 (ndf,q,vel,acc,efac,iev)
      implicit double precision(a-h,o-z)
c
c     ****************************************************************
c     subroutine to calculate event factor.
c     element type = 5. u bar restraint element, type b.
c     ****************************************************************
c
      dimension q(ndf),vel(ndf),acc(ndf)
c
      common /infel / imem,kst,lm(6),nodi,nodj,nodk,kgeom,jfix,kout,iup,
     *                ilow,line,iup7,xi,yi,zi,xj,yj,zj,spst(7),uval(6),
     *                tol,flp,vtot,ftot,vlow,vup7,sst,vup,dcos(3),
     *                dcosp(3),dcosn(3),dism,tdis,fom,tfo,toly,tolu,xkp,
     *                skp(21)
      common /infgr / ngr,nels,mfst,igrhed(10),ninfc,lstat,
     *                lstf,lstc,ndof,dko,dkt,eprop(4,250)
      common /precis/ zero,one
      common /work  / disi(3),disj(3),d(3)
c
c     iev event type indicator
c         1 = ubar gap closure
c         2 = ubar tensile yield
c         3 = ubar unloading
c         4 = ubar gap opening
c
c     displacements affecting extension
c
      ptnine = .9999
      do 100 i=1,3
100   disi(i)=q(i)
      if(jfix.eq.0)go to 120
      do 110 i=1,3
110   disj(i)=q(i+3)
      go to 140
120   do 130 i=1,3
130   disj(i)=0.
c
c     sweep out normal displacements
c
140   if(nodk.eq.0.or.line.ne.8)go to 190
      ff=0.
      do 150 i=1,3
150   ff=ff+dcosn(i)*disi(i)
      do 160 i=1,3
160   disi(i)=disi(i)-ff*dcosn(i)
      if(jfix.eq.0)go to 190
      ff=0.
      do 170 i=1,3
170   ff=ff+dcosn(i)*disj(i)
      do 180 i=1,3
180   disj(i)=disj(i)-ff*dcosn(i)
c
c     large displacements
c
190   xq = 0.
```

TABLE B.1 (CONTINUED)

```
      if (kgeom.eq.0) go to 210
      xii=xi+disi(1)
      yii=yi+disi(2)
      zii=zi+disi(3)
      xjj=xj+disj(1)
      yjj=yj+disj(2)
      zjj=zj+disj(3)
200   fl=sqrt((xii-xjj)**2+(yii-yjj)**2+(zii-zjj)**2)
c
      dv=fl-flp
      do 205 i=1,3
205   d(i) = disi(i) - disj(i)
      xl = 0.
      do 206 i=1,3
206   xl = xl + dcos(i) * d(i)
      xl = abs(xl)
      xq = d(1)**2 + d(2)**2 + d(3)**2
      if (xq.ne.zero) xl = flp * xl/xq
      go to 230
c
c     small displacements
c
210   dv=0.
      do 220 i=1,3
220   dv=dv+dcos(i)*(disi(i)-disj(i))
c
230   efac = 1.0
      iev = 0
c
      if (dv) 400,500,300
c
c     bar extends
c
300   fac = (vup-vtot)/dv
      if (fac.gt.ptnine) return
c
      if (line.ne.8) go to 350
      iev = 1
      efac = fac + tolu/((dv+vtot-vup)*spst(iup))
      go to 500
c
350   iev = 2
      efac = fac + toly/(dv*spst(line))
      go to 500
c
400   fac = (vlow-vtot)/dv
      if (fac.gt.ptnine) return
c
      if (line.eq.7) go to 450
      iev = 3
      efac = - tolu/(dv*spst(7))
      go to 500
c
450   iev = 4
      efac = fac - toly/(dv*spst(7))
c
```

TABLE B.1 (CONTINUED)

```
500   if (xq.eq.zero) return
      dv = efac * abs(dv)
      efac = -xl + sqrt( xl**2 + ( dv**2 + 2. * dv * flp ) / xq )
c
      return
      end
```

TABLE B.1 (CONTINUED)

```
      subroutine mdse05 (dvele,ndf)
      implicit double precision(a-h,o-z)
c
c     ****************************************************************************
c     subroutine to calculate the midstep error
c     element type = 5. u bar restraint element, type b.
c     ****************************************************************************
c
      dimension dvele(ndf)
c
      common /infel / imem,kst,lm(6),nodi,nodj,nodk,kgeom,jfix,kout,iup,
     *                ilow,line,iup7,xi,yi,zi,xj,yj,zj,spst(7),uval(6),
     *                tol,flp,vtot,ftot,vlow,vup7,sst,vup,dcos(3),
     *                dcosp(3),dcosn(3),dism,tdis,fom,tfo,toly,tolu,xkp,
     *                skp(21)
      common /infgr / ngr,nels,mfst,igrhed(10),ninfc,lstat,
     *                lstf,lstc,ndof,dko,dkt,eprop(4,250)
      common /work  / b(6),s(6,6)
c
c     form 6x6 element stiffness matrix
c
      xk = 0.
      if (line.ne.8) xk = spst(line)
c
      do 100 i=1,3
      cc = dcos(i)*xk
      do 100 j=1,3
100   s(i,j) = cc*dcos(j)
      if(jfix.eq.0)go to 300
      do 200 i=1,3
      do 200 j=1,3
      s(i,j+3) = -s(i,j)
      s(i+3,j) = -s(i,j)
200   s(i+3,j+3) = s(i,j)
c
c     compute midstep error
c
300   do 500 i=1,ndof
      sum = 0.0
      do 400 j=1,ndof
400   sum = sum + s(i,j)*dvele(j)
500   b(i) = sum
      do 600 i=1,ndof
600   dvele(i) = b(i)
c
      return
      end
```

TABLE B.1 (CONTINUED)

```
      subroutine rint05 (ndf,q,vel,fe,fd,time)
      implicit double precision(a-h,o-z)
c
c     ****************************************************************
c     subroutine for internal force calculations.
c     element type = 5. u bar restraint element, type b.
c     ****************************************************************
c
      common /infel / imem,kst,lm(6),nodi,nodj,nodk,kgeom,jfix,kout,iup,
     *                ilow,line,iup7,xi,yi,zi,xj,yj,zj,spst(7),uval(6),
     *                tol,flp,vtot,ftot,vlow,vup7,sst,vup,dcos(3),
     *                dcosp(3),dcosn(3),dism,tdis,fom,tfo,toly,tolu,xkp,
     *                skp(21)
      common /infgr / ngr,nels,mfst,igrhed(10),ninfc,lstat,
     *                lstf,lstc,ndof,dko,dkt,eprop(4,250)
      common /precis/ zero,one
c
      dimension q(ndf),fe(ndf),fd(ndf),vel(ndf)
c
c     compute equivalent nodal loads
c
      do 100 i=1,3
      fe(i+3)=0.
100   fe(i)=ftot*dcos(i)
      if(jfix.eq.0)go to 300
      do 200 i=1,3
200   fe(i+3)=-fe(i)
c
300   if(time.eq.zero)return
c
      xk = 0.
      if (line.ne.8) xk = spst(line)
c
      dvd=0.
      do 400 i=1,3
400   dvd=dvd-dcos(i)*vel(i)
      if(jfix.eq.0)go to 600
      do 500 i=1,3
500   dvd=dvd+dcos(i)*vel(i+3)
c
600   sd=(dko*sst+dkt*xk)*dvd
c
      do 700 i=1,3
700   fd(i)=sd*dcos(i)
      if(jfix.eq.0)return
      do 800 i=1,3
800   fd(i+3)=-fd(i)
c
      return
      end
```

TABLE B.1 (CONTINUED)

```
      subroutine rdyn05 (ndf,q,vel,acc,fz,c7,c8)
      implicit double precision(a-h,o-z)
c
c     ****************************************************************
c     subroutine for dynamic step load calculations.
c     element type = 5. u bar restraint element, type b.
c     ****************************************************************
c
      common /infel / imem,kst,lm(6),nodi,nodj,nodk,kgeom,jfix,kout,iup,
     *                ilow,line,iup7,xi,yi,zi,xj,yj,zj,spst(7),uval(6),
     *                tol,flp,vtot,ftot,vlow,vup7,sst,vup,dcos(3),
     *                dcosp(3),dcosn(3),dism,tdis,fom,tfo,toly,tolu,xkp,
     *                skp(21)
      common /infgr / ngr,nels,mfst,igrhed(10),ninfc,lstat,
     *                lstf,lstc,ndof,dko,dkt,eprop(4,250)
c
      dimension q(ndf),vel(ndf),acc(ndf),fz(ndf)
c
c     compute equivalent nodal loads
c
      xk = 0.
      if (line.ne.8) xk = spst(line)
c
      dvd=0.
      cc7=1.+c7
      do 100 i=1,3
100   dvd=dvd+dcos(i)*(cc7*vel(i)+c8*acc(i))
      if(jfix.eq.0)go to 300
      do 200 i=1,3
200   dvd=dvd-dcos(i)*(cc7*vel(i+3)+c8*acc(i+3))
c
300   sd=(dko*sst+dkt*xk)*dvd
c
      do 400 i=1,3
400   fz(i)=sd*dcos(i)
      if(jfix.eq.0)return
      do 500 i=1,3
500   fz(i+3)=-fz(i)
c
      return
      end
```

TABLE B.1 (CONTINUED)

```
      subroutine outs05 (kpr,time)
      implicit double precision (a-h,o-z)
c
c     **********************************************************************
c     subroutine to print time history of current state,
c     element type = 5. u bar restraint element, type b.
c     ********************************** carlo v. oughourlian ****
c
      common /tapes / niu,nou,nt1,nt2,nt3,nt4,nt5,nt6,nt7,nt8,nt9,
     *                nt10,nt11,nt12,nt13,nt14,nt15,nt16
      common /infgr / ngr,nels,mfst,igrhed(10),ninfc,lstat,
     *                lstf,lstc,ndof,dko,dkt,eprop(4,250)
      common /infel / imem,kst,lm(6),nodi,nodj,nodk,kgeom,jfix,kout,iup,
     *                ilow,line,iup7,xi,yi,zi,xj,yj,zj,spst(7),uval(6),
     *                tol,flp,vtot,ftot,vlow,vup7,sst,vup,dcos(3),
     *                dcosp(3),dcosn(3),dism,tdis,fom,tfo,toly,tolu,xkp,
     *                skp(21)
c
      if(imem.eq.mfst)khed=0
c
      if (kpr.eq.0.or.kout.eq.0) go to 200
      if (khed.ne.0) go to 100
      kkpr=iabs(kpr)
      write(nou,2000)kkpr,time,igrhed
2000  format(///16h results for group,i3,
     *       34h, u-bar restraints, type b, time =e11.4//5x,10a4//
     *       5x,5h elem,3x,4hnode,3x,4hnode,3x,4hline,5x,9hrestraint,
     *       5x,9hrestraint/
     *       5x,5h  no.,3x,4h  i ,3x,4h  j ,3x,4h no.,5x,9h  force  ,
     *       5x,9hextension/)
      khed=1
100   vtott=vtot
      if(line.eq.0) vtott=vup
      write(nou,2010)imem,nodi,nodj,line,ftot,vtott
2010  format(i9,3i7,e15.4,e14.4)
c     write(nt15,3000) time,ftot
3000  format(2f12.6)
200   continue
      return
      end
```

TABLE B.1 (CONTINUED)

```
      subroutine eout05
      implicit double precision(a-h,o-z)
c
c     ****************************************************************
c     subroutine to print peak values of response.
c     element type = 5. u bar restraint element, type b.
c     ****************************************************************
c
      common /infel / imem,kst,lm(6),nodi,nodj,nodk,kgeom,jfix,kout,iup,
     *                ilow,line,iup7,xi,yi,zi,xj,yj,zj,spst(7),uval(6),
     *                tol,flp,vtot,ftot,vlow,vup7,sst,vup,dcos(3),
     *                dcosp(3),dcosn(3),dism,tdis,fom,tfo,toly,tolu,xkp,
     *                skp(21)
      common /tapes / niu,nou,nt1,nt2,nt3,nt4,nt5,nt6,nt7,nt8,nt9,
     *                nt10,nt11,nt12,nt13,nt14,nt15,nt16
      common /infgr / ngr,nels,mfst,igrhed(10),ninfc,lstat,
     *                lstf,lstc,ndof,dko,dkt,eprop(4,250)
c
c     envelope output - u bar restraints,type b
c
      if (imem.eq.1)write(nou,2000)igrhed
 2000 format(25h u-bar restraints, type b//5x,10a4//
     *       5x,4helem,2x,4hnode,2x,4hnode,6x,7hmaximum,7x,4h    ,
     *       8x,9h maximum ,7x,4h    /
     *       5x,4h no.,2x,4h  i ,2x,4h  j ,6x,7h force ,7x,4htime,
     *       8x,9hextension,7x,4htime/)
c
      write(nou,2100) imem,nodi,nodj,fom,tfo,dism,tdis
 2100 format(i8,i6,i6,e15.4,e13.4,2e14.4)
c
      return
      end
```

EARTHQUAKE ENGINEERING RESEARCH CENTER REPORTS

NOTE: Numbers in parentheses are Accession Numbers assigned by the National Technical Information Service; these are followed by a price code. Copies of the reports may be ordered from the National Technical Information Service, 5285 Port Royal Road, Springfield, Virginia, 22161. Accession Numbers should be quoted on orders for reports (PB --- ---) and remittance must accompany each order. Reports without this information were not available at time of printing. The complete list of EERC reports (from EERC 67-1) is available upon request from the Earthquake Engineering Research Center, University of California, Berkeley, 47th Street and Hoffman Boulevard, Richmond, California 94804.

UCB/EERC-77/01 "PLUSH - A Computer Program for Probabilistic Finite Element Analysis of Seismic Soil-Structure Interaction," by M.P. Romo Organista, J. Lysmer and H.B. Seed - 1977 (PB81 177 651)A05

UCB/EERC-77/02 "Soil-Structure Interaction Effects at the Humboldt Bay Power Plant in the Ferndale Earthquake of June 7, 1975," by J.E. Valera, H.B. Seed, C.F. Tsai and J. Lysmer - 1977 (PB 265 795)A04

UCB/EERC-77/03 "Influence of Sample Disturbance on Sand Response to Cyclic Loading," by K. Mori, H.B. Seed and C.K. Chan - 1977 (PB 267 352)A04

UCB/EERC-77/04 "Seismological Studies of Strong Motion Records," by J. Shoja-Taheri - 1977 (PB 269 655)A10

UCB/EERC-77/05 Unassigned

UCB/EERC-77/06 "Developing Methodologies for Evaluating the Earthquake Safety of Existing Buildings," by No. 1 - B. Bresler; No. 2 - B. Bresler, T. Okada and D. Zisling; No. 3 - T. Okada and B. Bresler; No. 4 - V.V. Bertero and B. Bresler - 1977 (PB 267 354)A08

UCB/EERC-77/07 "A Literature Survey - Transverse Strength of Masonry Walls," by Y. Omote, R.L. Mayes, S.W. Chen and R.W. Clough - 1977 (PB 277 933)A07

UCB/EERC-77/08 "DRAIN-TABS: A Computer Program for Inelastic Earthquake Response of Three Dimensional Buildings," by R. Guendelman-Israel and G.H. Powell - 1977 (PB 270 693)A07

UCB/EERC-77/09 "SUBWALL: A Special Purpose Finite Element Computer Program for Practical Elastic Analysis and Design of Structural Walls with Substructure Option," by D.Q. Le, H. Peterson and E.P. Popov - 1977 (PB 270 567)A05

UCB/EERC-77/10 "Experimental Evaluation of Seismic Design Methods for Broad Cylindrical Tanks," by D.P. Clough (PB 272 280)A13

UCB/EERC-77/11 "Earthquake Engineering Research at Berkeley - 1976," - 1977 (PB 273 507)A09

UCB/EERC-77/12 "Automated Design of Earthquake Resistant Multistory Steel Building Frames," by N.D. Walker, Jr. - 1977 (PB 276 526)A09

UCB/EERC-77/13 "Concrete Confined by Rectangular Hoops Subjected to Axial Loads," by J. Vallenas, V.V. Bertero and E.P. Popov - 1977 (PB 275 165)A06

UCB/EERC-77/14 "Seismic Strain Induced in the Ground During Earthquakes," by Y. Sugimura - 1977 (PB 284 201)A04

UCB/EERC-77/15 Unassigned

UCB/EERC-77/16 "Computer Aided Optimum Design of Ductile Reinforced Concrete Moment Resisting Frames," by S.W. Zagajeski and V.V. Bertero - 1977 (PB 280 137)A07

UCB/EERC-77/17 "Earthquake Simulation Testing of a Stepping Frame with Energy-Absorbing Devices," by J.M. Kelly and D.F. Tsztoo - 1977 (PB 273 506)A04

UCB/EERC-77/18 "Inelastic Behavior of Eccentrically Braced Steel Frames under Cyclic Loadings," by C.W. Roeder and E.P. Popov - 1977 (PB 275 526)A15

UCB/EERC-77/19 "A Simplified Procedure for Estimating Earthquake-Induced Deformations in Dams and Embankments," by F.I. Makdisi and H.B. Seed - 1977 (PB 276 820)A04

UCB/EERC-77/20 "The Performance of Earth Dams during Earthquakes," by H.B. Seed, F.I. Makdisi and P. de Alba - 1977 (PB 276 821)A04

UCB/EERC-77/21 "Dynamic Plastic Analysis Using Stress Resultant Finite Element Formulation," by P. Lukkunapvasit and J.M. Kelly - 1977 (PB 275 453)A04

UCB/EERC-77/22 "Preliminary Experimental Study of Seismic Uplift of a Steel Frame," by R.W. Clough and A.A. Huckelbridge 1977 (PB 278 769)A08

UCB/EERC-77/23 "Earthquake Simulator Tests of a Nine-Story Steel Frame with Columns Allowed to Uplift," by A.A. Huckelbridge - 1977 (PB 277 944)A09

UCB/EERC-77/24 "Nonlinear Soil-Structure Interaction of Skew Highway Bridges," by M.-C. Chen and J. Penzien - 1977 (PB 276 176)A07

UCB/EERC-77/25 "Seismic Analysis of an Offshore Structure Supported on Pile Foundations," by D.D.-N. Liou and J. Penzien 1977 (PB 283 180)A06

UCB/EERC-77/26 "Dynamic Stiffness Matrices for Homogeneous Viscoelastic Half-Planes," by G. Dasgupta and A.K. Chopra - 1977 (PB 279 654)A06

UCB/EERC-77/27 "A Practical Soft Story Earthquake Isolation System," by J.M. Kelly, J.M. Eidinger and C.J. Derham - 1977 (PB 276 814)A07

UCB/EERC-77/28 "Seismic Safety of Existing Buildings and Incentives for Hazard Mitigation in San Francisco: An Exploratory Study," by A.J. Meltsner - 1977 (PB 281 970)A05

UCB/EERC-77/29 "Dynamic Analysis of Electrohydraulic Shaking Tables," by D. Rea, S. Abedi-Hayati and Y. Takahashi 1977 (PB 282 569)A04

UCB/EERC-77/30 "An Approach for Improving Seismic - Resistant Behavior of Reinforced Concrete Interior Joints," by B. Galunic, V.V. Bertero and E.P. Popov - 1977 (PB 290 870)A06

UCB/EERC-78/01 "The Development of Energy-Absorbing Devices for Aseismic Base Isolation Systems," by J.M. Kelly and D.F. Tsztoo - 1978 (PB 284 978)A04

UCB/EERC-78/02 "Effect of Tensile Prestrain on the Cyclic Response of Structural Steel Connections, by J.G. Bouwkamp and A. Mukhopadhyay - 1978

UCB/EERC-78/03 "Experimental Results of an Earthquake Isolation System using Natural Rubber Bearings," by J.M. Eidinger and J.M. Kelly - 1978 (PB 281 686)A04

UCB/EERC-78/04 "Seismic Behavior of Tall Liquid Storage Tanks," by A. Niwa - 1978 (PB 284 017)A14

UCB/EERC-78/05 "Hysteretic Behavior of Reinforced Concrete Columns Subjected to High Axial and Cyclic Shear Forces," by S.W. Zagajeski, V.V. Bertero and J.G. Bouwkamp - 1978 (PB 283 858)A13

UCB/EERC-78/06 "Three Dimensional Inelastic Frame Elements for the ANSR-I Program," by A. Riahi, D.G. Row and G.H. Powell - 1978 (PB 295 755)A04

UCB/EERC-78/07 "Studies of Structural Response to Earthquake Ground Motion," by O.A. Lopez and A.K. Chopra - 1978 (PB 282 790)A05

UCB/EERC-78/08 "A Laboratory Study of the Fluid-Structure Interaction of Submerged Tanks and Caissons in Earthquakes," by R.C. Byrd - 1978 (PB 284 957)A08

UCB/EERC-78/09 Unassigned

UCB/EERC-78/10 "Seismic Performance of Nonstructural and Secondary Structural Elements," by I. Sakamoto - 1978 (PB81 154 593)A05

UCB/EERC-78/11 "Mathematical Modelling of Hysteresis Loops for Reinforced Concrete Columns," by S. Nakata, T. Sproul and J. Penzien - 1978 (PB 298 274)A05

UCB/EERC-78/12 "Damageability in Existing Buildings," by T. Blejwas and B. Bresler - 1978 (PB 80 166 978)A05

UCB/EERC-78/13 "Dynamic Behavior of a Pedestal Base Multistory Building," by R.M. Stephen, E.L. Wilson, J.G. Bouwkamp and M. Button - 1978 (PB 286 650)A08

UCB/EERC-78/14 "Seismic Response of Bridges - Case Studies," by R.A. Imbsen, V. Nutt and J. Penzien - 1978 (PB 286 503)A10

UCB/EERC-78/15 "A Substructure Technique for Nonlinear Static and Dynamic Analysis," by D.G. Row and G.H. Powell - 1978 (PB 288 077)A10

UCB/EERC-78/16 "Seismic Risk Studies for San Francisco and for the Greater San Francisco Bay Area," by C.S. Oliveira - 1978 (PB 81 120 115)A07

UCB/EERC-78/17 "Strength of Timber Roof Connections Subjected to Cyclic Loads," by P. Gülkan, R.L. Mayes and R.W. Clough - 1978 (HUD-000 1491)A07

UCB/EERC-78/18 "Response of K-Braced Steel Frame Models to Lateral Loads," by J.G. Bouwkamp, R.M. Stephen and E.P. Popov - 1978

UCB/EERC-78/19 "Rational Design Methods for Light Equipment in Structures Subjected to Ground Motion," by J.L. Sackman and J.M. Kelly - 1978 (PB 292 357)A04

UCB/EERC-78/20 "Testing of a Wind Restraint for Aseismic Base Isolation," by J.M. Kelly and D.E. Chitty - 1978 (PB 292 833)A03

UCB/EERC-78/21 "APOLLO - A Computer Program for the Analysis of Pore Pressure Generation and Dissipation in Horizontal Sand Layers During Cyclic or Earthquake Loading," by P.P. Martin and H.B. Seed - 1978 (PB 292 835)A04

UCB/EERC-78/22 "Optimal Design of an Earthquake Isolation System," by M.A. Bhatti, K.S. Pister and E. Polak - 1978 (PB 294 735)A06

UCB/EERC-78/23 "MASH - A Computer Program for the Non-Linear Analysis of Vertically Propagating Shear Waves in Horizontally Layered Deposits," by P.P. Martin and H.B. Seed - 1978 (PB 293 101)A05

UCB/EERC-78/24 "Investigation of the Elastic Characteristics of a Three Story Steel Frame Using System Identification," by I. Kaya and H.D. McNiven - 1978 (PB 296 225)A06

UCB/EERC-78/25 "Investigation of the Nonlinear Characteristics of a Three-Story Steel Frame Using System Identification," by I. Kaya and H.D. McNiven - 1978 (PB 301 363)A05

UCB/EERC-78/26 "Studies of Strong Ground Motion in Taiwan," by Y.M. Hsiung, B.A. Bolt and J. Penzien - 1978 (PB 298 436)A06

UCB/EERC-78/27 "Cyclic Loading Tests of Masonry Single Piers: Volume 1 - Height to Width Ratio of 2," by P.A. Hidalgo, R.L. Mayes, H.D. McNiven and R.W. Clough - 1978 (PB 296 211)A07

UCB/EERC-78/28 "Cyclic Loading Tests of Masonry Single Piers: Volume 2 - Height to Width Ratio of 1," by S.-W.J. Chen, P.A. Hidalgo, R.L. Mayes, R.W. Clough and H.D. McNiven - 1978 (PB 296 212)A09

UCB/EERC-78/29 "Analytical Procedures in Soil Dynamics," by J. Lysmer - 1978 (PB 298 445)A06

UCB/EERC-79/01 "Hysteretic Behavior of Lightweight Reinforced Concrete Beam-Column Subassemblages," by B. Forzani, E.P. Popov and V.V. Bertero - April 1979(PB 298 267)A06

UCB/EERC-79/02 "The Development of a Mathematical Model to Predict the Flexural Response of Reinforced Concrete Beams to Cyclic Loads, Using System Identification," by J. Stanton & H. McNiven - Jan. 1979(PB 295 875)A10

UCB/EERC-79/03 "Linear and Nonlinear Earthquake Response of Simple Torsionally Coupled Systems," by C.L. Kan and A.K. Chopra - Feb. 1979(PB 298 262)A06

UCB/EERC-79/04 "A Mathematical Model of Masonry for Predicting its Linear Seismic Response Characteristics," by Y. Mengi and H.D. McNiven - Feb. 1979(PB 298 266)A06

UCB/EERC-79/05 "Mechanical Behavior of Lightweight Concrete Confined by Different Types of Lateral Reinforcement," by M.A. Manrique, V.V. Bertero and E.P. Popov - May 1979(PB 301 114)A06

UCB/EERC-79/06 "Static Tilt Tests of a Tall Cylindrical Liquid Storage Tank," by R.W. Clough and A. Niwa - Feb. 1979 (PB 301 167)A06

UCB/EERC-79/07 "The Design of Steel Energy Absorbing Restrainers and Their Incorporation into Nuclear Power Plants for Enhanced Safety: Volume 1 - Summary Report," by P.N. Spencer, V.F. Zackay, and E.R. Parker - Feb. 1979(UCB/EERC-79/07)A09

UCB/EERC-79/08 "The Design of Steel Energy Absorbing Restrainers and Their Incorporation into Nuclear Power Plants for Enhanced Safety: Volume 2 - The Development of Analyses for Reactor System Piping,""Simple Systems" by M.C. Lee, J. Penzien, A.K. Chopra and K, Suzuki "Complex Systems" by G.H. Powell, E.L. Wilson, R.W. Clough and D.G. Row - Feb. 1979(UCB/EERC-79/08)A10

UCB/EERC-79/09 "The Design of Steel Energy Absorbing Restrainers and Their Incorporation into Nuclear Power Plants for Enhanced Safety: Volume 3 - Evaluation of Commercial Steels," by W.S. Owen, R.M.N. Pelloux, R.O. Ritchie, M. Faral, T. Ohhashi, J. Toplosky, S.J. Hartman, V.F. Zackay and E.R. Parker - Feb. 1979(UCB/EERC-79/09)A04

UCB/EERC-79/10 "The Design of Steel Energy Absorbing Restrainers and Their Incorporation into Nuclear Power Plants for Enhanced Safety: Volume 4 - A Review of Energy-Absorbing Devices," by J.M. Kelly and M.S. Skinner - Feb. 1979(UCB/EERC-79/10)A04

UCB/EERC-79/11 "Conservatism In Summation Rules for Closely Spaced Modes," by J.M. Kelly and J.L. Sackman - May 1979(PB 301 328)A03

UCB/EERC-79/12 "Cyclic Loading Tests of Masonry Single Piers; Volume 3 - Height to Width Ratio of 0.5," by P.A. Hidalgo, R.L. Mayes, H.D. McNiven and R.W. Clough - May 1979(PB 301 321)A08

UCB/EERC-79/13 "Cyclic Behavior of Dense Course-Grained Materials in Relation to the Seismic Stability of Dams," by N.G. Banerjee, H.B. Seed and C.K. Chan - June 1979(PB 301 373)A13

UCB/EERC-79/14 "Seismic Behavior of Reinforced Concrete Interior Beam-Column Subassemblages," by S. Viwathanatepa, E.P. Popov and V.V. Bertero - June 1979(PB 301 326)A10

UCB/EERC-79/15 "Optimal Design of Localized Nonlinear Systems with Dual Performance Criteria Under Earthquake Excitations," by M.A. Bhatti - July 1979(PB 80 167 109)A06

UCB/EERC-79/16 "OPTDYN - A General Purpose Optimization Program for Problems with or without Dynamic Constraints," by M.A. Bhatti, E. Polak and K.S. Pister - July 1979(PB 80 167 091)A05

UCB/EERC-79/17 "ANSR-II, Analysis of Nonlinear Structural Response, Users Manual," by D.P. Mondkar and G.H. Powell July 1979(PB 80 113 301)A05

UCB/EERC-79/18 "Soil Structure Interaction in Different Seismic Environments," A. Gomez-Masso, J. Lysmer, J.-C. Chen and H.B. Seed - August 1979(PB 80 101 520)A04

UCB/EERC-79/19 "ARMA Models for Earthquake Ground Motions," by M.K. Chang, J.W. Kwiatkowski, R.F. Nau, R.M. Oliver and K.S. Pister - July 1979(PB 301 166)A05

UCB/EERC-79/20 "Hysteretic Behavior of Reinforced Concrete Structural Walls," by J.M. Vallenas, V.V. Bertero and E.P. Popov - August 1979(PB 80 165 905)A12

UCB/EERC-79/21 "Studies on High-Frequency Vibrations of Buildings - 1: The Column Effect," by J. Lubliner - August 1979 (PB 80 158 553)A03

UCB/EERC-79/22 "Effects of Generalized Loadings on Bond Reinforcing Bars Embedded in Confined Concrete Blocks," by S. Viwathanatepa, E.P. Popov and V.V. Bertero - August 1979(PB 81 124 018)A14

UCB/EERC-79/23 "Shaking Table Study of Single-Story Masonry Houses, Volume 1: Test Structures 1 and 2," by P. Gülkan, R.L. Mayes and R.W. Clough - Sept. 1979 (HUD-000 1763)A12

UCB/EERC-79/24 "Shaking Table Study of Single-Story Masonry Houses, Volume 2: Test Structures 3 and 4," by P. Gülkan, R.L. Mayes and R.W. Clough - Sept. 1979 (HUD-000 1836)A12

UCB/EERC-79/25 "Shaking Table Study of Single-Story Masonry Houses, Volume 3: Summary, Conclusions and Recommendations," by R.W. Clough, R.L. Mayes and P. Gülkan - Sept. 1979 (HUD-000 1837)A06

UCB/EERC-79/26 "Recommendations for a U.S.-Japan Cooperative Research Program Utilizing Large-Scale Testing Facilities," by U.S.-Japan Planning Group - Sept. 1979(PB 301 407)A06

UCB/EERC-79/27 "Earthquake-Induced Liquefaction Near Lake Amatitlan, Guatemala," by H.B. Seed, I. Arango, C.K. Chan, A. Gomez-Masso and R. Grant de Ascoli.- Sept. 1979(NUREG-CR1341)A03

UCB/EERC-79/28 "Infill Panels: Their Influence on Seismic Response of Buildings," by J.W. Axley and V.V. Bertero Sept. 1979(PB 80 163 371)A10

UCB/EERC-79/29 "3D Truss Bar Element (Type 1) for the ANSR-II Program," by D.P. Mondkar and G.H. Powell - Nov. 1979 (PB 80 169 709)A02

UCB/EERC-79/30 "2D Beam-Column Element (Type 5 - Parallel Element Theory) for the ANSR-II Program," by D.G. Row, G.H. Powell and D.P. Mondkar - Dec. 1979(PB 80 167 224)A03

UCB/EERC-79/31 "3D Beam-Column Element (Type 2 - Parallel Element Theory) for the ANSR-II Program," by A. Riahi, G.H. Powell and D.P. Mondkar - Dec. 1979(PB 80 167 216)A03

UCB/EERC-79/32 "On Response of Structures to Stationary Excitation," by A. Der Kiureghian - Dec. 1979(PB 80166 929)A03

UCB/EERC-79/33 "Undisturbed Sampling and Cyclic Load Testing of Sands," by S. Singh, H.B. Seed and C.K. Chan Dec. 1979(ADA 087 298)A07

UCB/EERC-79/34 "Interaction Effects of Simultaneous Torsional and Compressional Cyclic Loading of Sand," by P.M. Griffin and W.N. Houston - Dec. 1979(ADA 092 352)A15

UCB/EERC-80/01 "Earthquake Response of Concrete Gravity Dams Including Hydrodynamic and Foundation Interaction Effects," by A.K. Chopra, P. Chakrabarti and S. Gupta - Jan. 1980(AD-A087297)A10

UCB/EERC-80/02 "Rocking Response of Rigid Blocks to Earthquakes," by C.S. Yim, A.K. Chopra and J. Penzien - Jan. 1980 (PB80 166 002)A04

UCB/EERC-80/03 "Optimum Inelastic Design of Seismic-Resistant Reinforced Concrete Frame Structures," by S.W. Zagajeski and V.V. Bertero - Jan. 1980(PB80 164 635)A06

UCB/EERC-80/04 "Effects of Amount and Arrangement of Wall-Panel Reinforcement on Hysteretic Behavior of Reinforced Concrete Walls," by R. Iliya and V.V. Bertero - Feb. 1980(PB81 122 525)A09

UCB/EERC-80/05 "Shaking Table Research on Concrete Dam Models," by A. Niwa and R.W. Clough - Sept. 1980(PB81 122 368)A06

UCB/EERC-80/06 "The Design of Steel Energy-Absorbing Restrainers and their Incorporation into Nuclear Power Plants for Enhanced Safety (Vol 1A): Piping with Energy Absorbing Restrainers: Parameter Study on Small Systems," by G.H. Powell, C. Oughourlian and J. Simons - June 1980

UCB/EERC-80/07 "Inelastic Torsional Response of Structures Subjected to Earthquake Ground Motions," by Y. Yamazaki April 1980(PB81 122 327)A08

UCB/EERC-80/08 "Study of X-Braced Steel Frame Structures Under Earthquake Simulation," by Y. Ghanaat - April 1980 (PB81 122 335)A11

UCB/EERC-80/09 "Hybrid Modelling of Soil-Structure Interaction," by S. Gupta, T.W. Lin, J. Penzien and C.S. Yeh May 1980(PB81 122 319)A07

UCB/EERC-80/10 "General Applicability of a Nonlinear Model of a One Story Steel Frame," by B.I. Sveinsson and H.D. McNiven - May 1980(PB81 124 877)A06

UCB/EERC-80/11 "A Green-Function Method for Wave Interaction with a Submerged Body," by W. Kioka - April 1980 (PB81 122 269)A07

UCB/EERC-80/12 "Hydrodynamic Pressure and Added Mass for Axisymmetric Bodies," by F. Nilrat - May 1980(PB81 122 343)A08

UCB/EERC-80/13 "Treatment of Non-Linear Drag Forces Acting on Offshore Platforms," by B.V. Dao and J. Penzien May 1980(PB81 153 413)A07

UCB/EERC-80/14 "2D Plane/Axisymmetric Solid Element (Type 3 - Elastic or Elastic-Perfectly Plastic) for the ANSR-II Program," by D.P. Mondkar and G.H. Powell - July 1980(PB81 122 350)A03

UCB/EERC-80/15 "A Response Spectrum Method for Random Vibrations," by A. Der Kiureghian - June 1980(PB81 122 301)A03

UCB/EERC-80/16 "Cyclic Inelastic Buckling of Tubular Steel Braces," by V.A. Zayas, E.P. Popov and S.A. Mahin June 1980(PB81 124 885)A10

UCB/EERC-80/17 "Dynamic Response of Simple Arch Dams Including Hydrodynamic Interaction," by C.S. Porter and A.K. Chopra - July 1980(PB81 124 000)A13

UCB/EERC-80/18 "Experimental Testing of a Friction Damped Aseismic Base Isolation System with Fail-Safe Characteristics," by J.M. Kelly, K.E. Beucke and M.S. Skinner - July 1980(PB81 148 595)A04

UCB/EERC-80/19 "The Design of Steel Energy-Absorbing Restrainers and their Incorporation into Nuclear Power Plants for Enhanced Safety (Vol 1B): Stochastic Seismic Analyses of Nuclear Power Plant Structures and Piping Systems Subjected to Multiple Support Excitations," by M.C. Lee and J. Penzien - June 1980

UCB/EERC-80/20 "The Design of Steel Energy-Absorbing Restrainers and their Incorporation into Nuclear Power Plants for Enhanced Safety (Vol 1C): Numerical Method for Dynamic Substructure Analysis," by J.M. Dickens and E.L. Wilson - June 1980

UCB/EERC-80/21 "The Design of Steel Energy-Absorbing Restrainers and their Incorporation into Nuclear Power Plants for Enhanced Safety (Vol 2): Development and Testing of Restraints for Nuclear Piping Systems," by J.M. Kelly and M.S. Skinner - June 1980

UCB/EERC-80/22 "3D Solid Element (Type 4-Elastic or Elastic-Perfectly-Plastic) for the ANSR-II Program," by D.P. Mondkar and G.H. Powell - July 1980(PB81 123 242)A03

UCB/EERC-80/23 "Gap-Friction Element (Type 5) for the ANSR-II Program," by D.P. Mondkar and G.H. Powell - July 1980 (PB81 122 285)A03

UCB/EERC-80/24 "U-Bar Restraint Element (Type 11) for the ANSR-II Program," by C. Oughourlian and G.H. Powell July 1980(PB81 122 293)A03

UCB/EERC-80/25 "Testing of a Natural Rubber Base Isolation System by an Explosively Simulated Earthquake," by J.M. Kelly - August 1980(PB81 201 360)A04

UCB/EERC-80/26 "Input Identification from Structural Vibrational Response," by Y. Hu - August 1980(PB81 152 308)A05

UCB/EERC-80/27 "Cyclic Inelastic Behavior of Steel Offshore Structures," by V.A. Zayas, S.A. Mahin and E.P. Popov August 1980(PB81 196 180)A15

UCB/EERC-80/28 "Shaking Table Testing of a Reinforced Concrete Frame with Biaxial Response," by M.G. Oliva October 1980(PB81 154 304)A10

UCB/EERC-80/29 "Dynamic Properties of a Twelve-Story Prefabricated Panel Building," by J.G. Bouwkamp, J.P. Kollegger and R.M. Stephen - October 1980(PB82 117 128)A06

UCB/EERC-80/30 "Dynamic Properties of an Eight-Story Prefabricated Panel Building," by J.G. Bouwkamp, J.P. Kollegger and R.M. Stephen - October 1980(PB81 200 313)A05

UCB/EERC-80/31 "Predictive Dynamic Response of Panel Type Structures Under Earthquakes," by J.P. Kollegger and J.G. Bouwkamp - October 1980(PB81 152 316)A04

UCB/EERC-80/32 "The Design of Steel Energy-Absorbing Restrainers and their Incorporation into Nuclear Power Plants for Enhanced Safety (Vol 3): Testing of Commercial Steels in Low-Cycle Torsional Fatigue," by P. Spencer, E.R. Parker, E. Jongewaard and M. Drory

UCB/EERC-80/33 "The Design of Steel Energy-Absorbing Restrainers and their Incorporation into Nuclear Power Plants for Enhanced Safety (Vol 4): Shaking Table Tests of Piping Systems with Energy-Absorbing Restrainers," by S.F. Stiemer and W.G. Godden - Sept. 1980

UCB/EERC-80/34 "The Design of Steel Energy-Absorbing Restrainers and their Incorporation into Nuclear Power Plants for Enhanced Safety (Vol 5): Summary Report," by P. Spencer

UCB/EERC-80/35 "Experimental Testing of an Energy-Absorbing Base Isolation System," by J.M. Kelly, M.S. Skinner and K.E. Beucke - October 1980(PB81 154 072)A04

UCB/EERC-80/36 "Simulating and Analyzing Artificial Non-Stationary Earthquake Ground Motions," by R.F. Nau, R.M. Oliver and K.S. Pister - October 1980(PB81 153 397)A04

UCB/EERC-80/37 "Earthquake Engineering at Berkeley - 1980," - Sept. 1980(PB81 205 674)A09

UCB/EERC-80/38 "Inelastic Seismic Analysis of Large Panel Buildings," by V. Schricker and G.H. Powell - Sept. 1980 (PB81 154 338)A13

UCB/EERC-80/39 "Dynamic Response of Embankment, Concrete-Gravity and Arch Dams Including Hydrodynamic Interaction," by J.F. Hall and A.K. Chopra - October 1980(PB81 152 324)A11

UCB/EERC-80/40 "Inelastic Buckling of Steel Struts Under Cyclic Load Reversal," by R.G. Black, W.A. Wenger and E.P. Popov - October 1980(PB81 154 312)A08

UCB/EERC-80/41 "Influence of Site Characteristics on Building Damage During the October 3, 1974 Lima Earthquake," by P. Repetto, I. Arango and H.B. Seed - Sept. 1980(PB81 161 739)A05

UCB/EERC-80/42 "Evaluation of a Shaking Table Test Program on Response Behavior of a Two Story Reinforced Concrete Frame," by J.M. Blondet, R.W. Clough and S.A. Mahin

UCB/EERC-80/43 "Modelling of Soil-Structure Interaction by Finite and Infinite Elements," by F. Medina - December 1980(PB81 229 270)A04

UCB/EERC-81/01 "Control of Seismic Response of Piping Systems and Other Structures by Base Isolation," edited by J.M. Kelly - January 1981 (PB81 200 735)A05

UCB/EERC-81/02 "OPTNSR - An Interactive Software System for Optimal Design of Statically and Dynamically Loaded Structures with Nonlinear Response," by M.A. Bhatti, V. Ciampi and K.S. Pister - January 1981 (PB81 218 851)A09

UCB/EERC-81/03 "Analysis of Local Variations in Free Field Seismic Ground Motions," by J.-C. Chen, J. Lysmer and H.B. Seed - January 1981 (AD-A099508)A13

UCB/EERC-81/04 "Inelastic Structural Modeling of Braced Offshore Platforms for Seismic Loading," by V.A. Zayas, P.-S.B. Shing, S.A. Mahin and E.P. Popov - January 1981(PB82 138 777)A07

UCB/EERC-81/05 "Dynamic Response of Light Equipment in Structures," by A. Der Kiureghian, J.L. Sackman and B. Nour-Omid - April 1981 (PB81 218 497)A04

UCB/EERC-81/06 "Preliminary Experimental Investigation of a Broad Base Liquid Storage Tank," by J.G. Bouwkamp, J.P. Kollegger and R.M. Stephen - May 1981(PB82 140 385)A03

UCB/EERC-81/07 "The Seismic Resistant Design of Reinforced Concrete Coupled Structural Walls," by A.E. Aktan and V.V. Bertero - June 1981(PB82 113 358)A11

UCB/EERC-81/08 "The Undrained Shearing Resistance of Cohesive Soils at Large Deformations," by M.R. Pyles and H.B. Seed - August 1981

UCB/EERC-81/09 "Experimental Behavior of a Spatial Piping System with Steel Energy Absorbers Subjected to a Simulated Differential Seismic Input," by S.F. Stiemer, W.G. Godden and J.M. Kelly - July 1981

UCB/EERC-81/10 "Evaluation of Seismic Design Provisions for Masonry in the United States," by B.I. Sveinsson, R.L. Mayes and H.D. McNiven - August 1981

UCB/EERC-81/11 "Two-Dimensional Hybrid Modelling of Soil-Structure Interaction," by T.-J. Tzong, S. Gupta and J. Penzien - August 1981(PB82 142 118)A04

UCB/EERC-81/12 "Studies on Effects of Infills in Seismic Resistant R/C Construction," by S. Brokken and V.V. Bertero - September 1981

UCB/EERC-81/13 "Linear Models to Predict the Nonlinear Seismic Behavior of a One-Story Steel Frame," by H. Valdimarsson, A.H. Shah and H.D. McNiven - September 1981(PB82 138 793)A07

UCB/EERC-81/14 "TLUSH: A Computer Program for the Three-Dimensional Dynamic Analysis of Earth Dams," by T. Kagawa, L.H. Mejia, H.B. Seed and J. Lysmer - September 1981(PB82 139 940)A06

UCB/EERC-81/15 "Three Dimensional Dynamic Response Analysis of Earth Dams," by L.H. Mejia and H.B. Seed - September 1981 (PB82 137 274)A12

UCB/EERC-81/16 "Experimental Study of Lead and Elastomeric Dampers for Base Isolation Systems," by J.M. Kelly and S.B. Hodder - October 1981 (PB82 166 182)A05

UCB/EERC-81/17 "The Influence of Base Isolation on the Seismic Response of Light Secondary Equipment," by J.M. Kelly - April 1981 (PB82 255 266)A04

UCB/EERC-81/18 "Studies on Evaluation of Shaking Table Response Analysis Procedures," by J. Marcial Blondet - November 1981 (PB82 197 278)A10

UCB/EERC-81/19 "DELIGHT.STRUCT: A Computer-Aided Design Environment for Structural Engineering," by R.J. Balling, K.S. Pister and E. Polak - December 1981 (PB82 218 496)A07

UCB/EERC-81/20 "Optimal Design of Seismic-Resistant Planar Steel Frames," by R.J. Balling, V. Ciampi, K.S. Pister and E. Polak - December 1981 (PB82 220 179)A07

UCB/EERC-82/01 "Dynamic Behavior of Ground for Seismic Analysis of Lifeline Systems," by T. Sato and A. Der Kiureghian - January 1982 (PB82 218 926)A05

UCB/EERC-82/02 "Shaking Table Tests of a Tubular Steel Frame Model," by Y. Ghanaat and R. W. Clough - January 1982 (PB82 220 161)A07

UCB/EERC-82/03 "Experimental Behavior of a Spatial Piping System with Shock Arrestors and Energy Absorbers under Seismic Excitation," by S. Schneider, H.-M. Lee and G. W. Godden - May 1982

UCB/EERC-82/04 "New Approaches for the Dynamic Analysis of Large Structural Systems," by E. L. Wilson - June 1982 (PB83 148 080)A05

UCB/EERC-82/05 "Model Study of Effects of Damage on the Vibration Properties of Steel Offshore Platforms," by P. Shahrivar and J. G. Bouwkamp - June 1982

UCB/EERC-82/06 "States of the Art and Practice in the Optimum Seismic Design and Analytical Response Prediction of R/C Frame-Wall Structures," by A. E. Aktan and V. V. Bertero - July 1982 (PB83 147 736)A05

UCB/EERC-82/07 "Further Study of the Earthquake Response of a Broad Cylindrical Liquid-Storage Tank Model," by G. C. Manos and R. W. Clough - July 1982 (PB83 147 744)A11

UCB/EERC-82/08 "An Evaluation of the Design and Analytical Seismic Response of a Seven Story Reinforced Concrete Frame - Wall Structure," by F. A. Charney and V. V. Bertero - July 1982

UCB/EERC-82/09 "Fluid-Structure Interactions: Added Mass Computations for Incompressible Fluid," by J. S.-H. Kuo - August 1982

UCB/EERC-82/10 "Joint-Opening Nonlinear Mechanism: Interface Smeared Crack Model," by J. S.-H. Kuo - August 1982 (PB83 149 195)A05

UCB/EERC-82/11 "Dynamic Response Analysis of Techi Dam," by R. W. Clough, R. M. Stephen and J. S.-H. Kuo - August 1982 (PB83 147 496)A06

UCB/EERC-82/12 "Prediction of the Seismic Responses of R/C Frame-Coupled Wall Structures," by A. E. Aktan, V. V. Bertero and M. Piazza - August 1982 (PB83 149 203)A09

UCB/EERC-82/13 "Preliminary Report on the SMART 1 Strong Motion Array in Taiwan," by B. A. Bolt, C. H. Loh, J. Penzien, Y. B. Tsai and Y. T. Yeh - August 1982

UCB/EERC-82/14 "Shaking-Table Studies of an Eccentrically X-Braced Steel Structure," by M. S. Yang - September 1982

UCB/EERC-82/15 "The Performance of Stairways in Earthquakes," by C. Roha, J. W. Axley and V. V. Bertero - September 1982

UCB/EERC-82/16 "The Behavior of Submerged Multiple Bodies in Earthquakes," by W.-G. Liao - September 1982

UCB/EERC-82/17 "Effects of Concrete Types and Loading Conditions on Local Bond-Slip Relationships," by A. D. Cowell, E. P. Popov and V. V. Bertero - September 1982

UCB/EERC-82/18 "Mechanical Behavior of Shear Wall Vertical Boundary Members: An Experimental Investigation," by M. T. Wagner and V. V. Bertero - October 1982

UCB/EERC-82/19 "Experimental Studies of Multi-support Seismic Loading on Piping Systems," by J. M. Kelly and A. D. Cowell - November 1982

UCB/EERC-82/20 "Generalized Plastic Hinge Concepts for 3D Beam-Column Elements," by P. F.-S. Chen and G. H. Powell - November 1982

UCB/EERC-82/21 "ANSR-III: General Purpose Computer Program for Nonlinear Structural Analysis," by C. V. Oughourlian and G. H. Powell - November 1982