

**U.S. DEPARTMENT OF COMMERCE
National Technical Information Service**

PB-270 693

**DRAIN-TABS: A Computer Program for
Inelastic Earthquake Response of
Three-Dimensional Buildings**

California Univ, Berkeley Earthquake Engineering Research Ctr.

Prepared for

National Science Foundation, Washington, D C

Mar 77

PB 270 693

REPORT NO.
UCB/EERC-77/08
MARCH 1977

EARTHQUAKE ENGINEERING RESEARCH CENTER

DRAIN-TABS

**A COMPUTER PROGRAM FOR
INELASTIC EARTHQUAKE RESPONSE
OF THREE-DIMENSIONAL BUILDINGS**

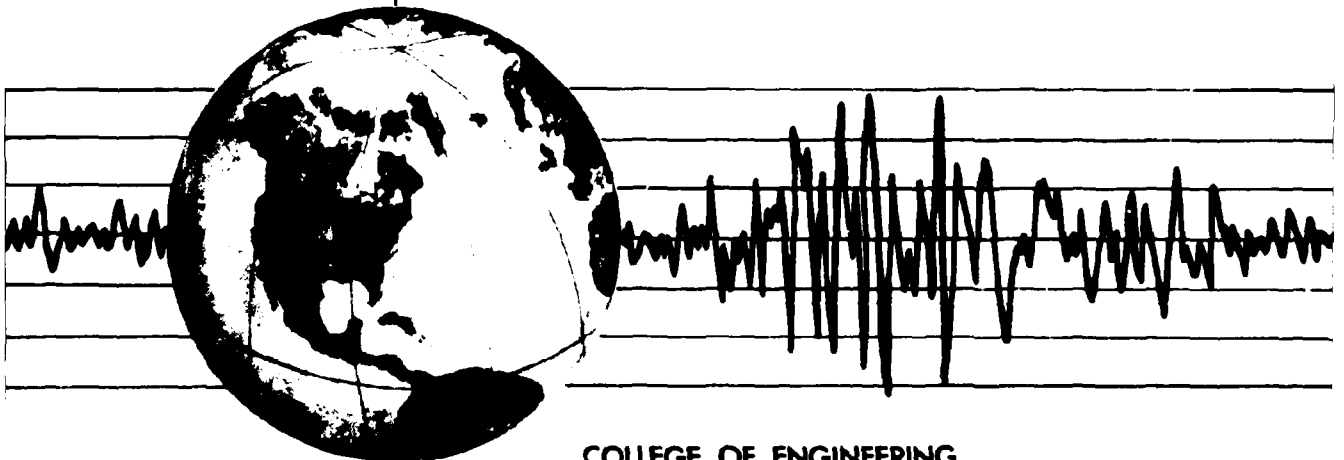
by

RAFAEL GUENDELMAN-ISRAEL

and

GRAHAM H. POWELL

Report to Sponsor:
National Science Foundation
Grant ENV76-04262



COLLEGE OF ENGINEERING

UNIVERSITY OF CALIFORNIA · Berkeley, California

REPRODUCED BY
**NATIONAL TECHNICAL
INFORMATION SERVICE**
U. S. DEPARTMENT OF COMMERCE
SPRINGFIELD, VA. 22161

BIBLIOGRAPHIC DATA SHEET		1. Report No. UCB/EERC-77/08	2.	3. Recipient's Accession No.	
4. Title and Subtitle DRAIN-TABS A computer Program for Inelastic Earthquake Response of Three-Dimensional Buildings			5. Report Date March 1977		
7. Author(s) Rafael Guendelman-Israel and Graham H. Powell			8. Performing Organization Rept. No. 77/08		
9. Performing Organization Name and Address Earthquake Engineering Research Center University of California, Berkeley 47th Street and Hoffman Boulevard Richmond, California 94804			10. Project/Task/Work Unit No.		
			11. Contract/Grant No. ENV76-04262		
12. Sponsoring Organization Name and Address National Science Foundation 1800 G Street, N.W. Washington, D.C. 20550			13. Type of Report & Period Covered		
			14.		
15. Supplementary Notes					
16. Abstracts A computational procedure and computer program for the inelastic dynamic response analysis of three-dimensional buildings of essentially arbitrary configurations is described. The building is idealized as a series of independent plane substructures interconnected by horizontal rigid diaphragms. Each substructure can be of arbitrary geometry and include structural elements of a variety of types. It is not necessary for all substructures to connect to all diaphragms, so that structures with independent diaphragms of some levels can be idealized (as, for example, two towers connected only at the top floors). The analysis makes use of substructuring techniques to improve computational efficiency. The major limitation is that the coupling of the substructures through common columns is not fully taken into account, so that the idealization is not suitable for tube-type buildings. The program consists of a "base" program which reads and prints data for the structure and its loading, allocates storage, carries out a variety of bookkeeping operations, assembles the substructure and building stiffnesses and loadings, solves the equilibrium equations, and determines the displacement response. This base program is then combined with a library of element subroutines to produce the complete program. Subroutines for new elements can be developed independently and added to the element library with relative ease. Subroutines for truss, beam column, shear panel, semi-rigid connection, and beam elements are currently included. The structure idealization is explained, and the computational procedure and computer program logic are described. Instructions to be followed when adding new elements to the program are presented. A computer program user's guide and an illustrative example are included.					
17c. COSATI Field Group					
18. Availability Statement Release unlimited			19. Security Class (This Report) UNCLASSIFIED		21. No. of Pages 147
			20. Security Class (This Page) UNCLASSIFIED		22. Price PCAO7 MF AD1

il

DRAIN-TABS: A COMPUTER PROGRAM FOR INELASTIC
EARTHQUAKE RESPONSE OF THREE-DIMENSIONAL BUILDINGS

by

Rafael Guendelman-Israel
University of Chile
Santiago, Chile

and

Graham H. Powell
Professor of Civil Engineering
University of California, Berkeley

Prepared under the sponsorship of
the National Science Foundation
and the Ford Foundation

Report No. UCB/EERC-77/08
Earthquake Engineering Research Center
College of Engineering
University of California
Berkeley, California

March 1977

ABSTRACT

A computational procedure and computer program for the inelastic dynamic response analysis of three-dimensional buildings of essentially arbitrary configurations is described.

The building is idealized as a series of independent plane substructures interconnected by horizontal rigid diaphragms. Each substructure can be of arbitrary geometry and include structural elements of a variety of types. It is not necessary for all substructures to connect to all diaphragms, so that structures with independent diaphragms of some levels can be idealized (as, for example, two towers connected only at the top floors). The analysis makes use of substructuring techniques to improve computational efficiency.

The major limitation is that the coupling of the substructures through common columns is not fully taken into account, so that the idealization is not suitable for tube-type buildings.

The program consists of a "base" program which reads and prints data for the structure and its loading, allocates storage, carries out a variety of bookkeeping operations, assembles the substructure and building stiffnesses and loadings, solves the equilibrium equations, and determines the displacement response. This base program is then combined with a library of element subroutines to produce the complete program. Subroutines for new elements can be developed independently and added to the element library with relative ease. Subroutines for truss, beam column, shear panel, semi-rigid connection, and beam elements are currently included.

The structure idealization is explained, and the computational procedure and computer program logic are described. Instructions to be followed when adding new elements to the program are presented. A computer program user's guide and an illustrative example are included.

ACKNOWLEDGEMENT

The research on which this report is based was supported in part by the National Science Foundation (grant ENV76-04262) and in part by the Ford Foundation (University of Chile - University of California Cooperative Program). This support is gratefully acknowledged.

Gratitude is expressed to Robin Cranford and Ellen Pirtz for typing the report, and to Lee Rambeau for drafting the illustrations.

TABLE OF CONTENTS

	<u>Page</u>
1. INTRODUCTION.	1
1.1 Objective.	1
1.2 Report Layout.	2
2. STRUCTURE IDEALIZATION.	3
3. COMPUTATIONAL PROCEDURE	8
3.1 General.	8
3.2 Frame Degrees of Freedom	8
3.3 Frame Stiffness.	10
3.4 Frame Load Vector.	11
3.5 Frame Mass Matrix.	11
3.6 Condensation of Frame Stiffness and Load Vector. . .	12
3.7 Building Stiffness	13
3.8 Building Load Vector	15
3.9 Building Mass Matrix	16
3.10 Solution for Static Loads.	16
3.11 Step-by-step Dynamic Analysis.	17
3.12 Geometric Stiffness.	17
3.13 Common Columns	19
3.14 Solution of Modified Equilibrium Equations	20
4. PROGRAM FEATURES	
4.1 Program Concepts	26
4.2 Modularity	26
4.3 Storage Allocation	26
4.4 Storage Options.	28
4.5 Overlay Structure.	31
4.6 Field Length	31
4.7 Random Access Requirement.	32
4.8 Reorganized Time-Histories	33
5. ADDITION OF ELEMENTS TO PROGRAM	38
5.1 General.	38
5.2 Control Information.	39

	<u>Page</u>
5.3 Element Information Block.	41
5.4 Time History Block	44
5.5 Work Block	46
5.6 Other Arrays	46
5.7 Common Column Procedure.	47
5.8 Subroutine <u>INELX</u>	49
5.9 Subroutine <u>STIFX</u>	53
5.10 Subroutine <u>RESPX</u>	55
5.11 Subroutine <u>OUTX</u>	59
5.12 Subroutine <u>THPRX</u>	60
6. EXAMPLE	62
7. CONCLUSION.	74
APPENDIX A: PROGRAM USER'S GUIDE	75
REFERENCES.	125

1. INTRODUCTION

1.1 OBJECTIVE

The seismic resistant design methods specified in building codes are intended primarily to safeguard public safety. They recognize that significant structural damage may be acceptable under certain circumstances, and implicitly rely on the inelastic deformations associated with structural damage to dissipate seismic energy. It is essential for safe performance, however, that the demands placed by an earthquake on the ductility of the structure should not exceed the available ductility, otherwise loss of strength may be severe and collapse may result. Methods of dynamic response analysis based on linear elastic assumptions can be carried out conveniently and economically, and can greatly assist in the production of safe designs. Such methods, however, cannot provide direct information on the inelastic behavior of the structure, or on the ductility demands on the structural members. To obtain such information, inelastic response analyses are necessary.

Buildings are three-dimensional, and often cannot be represented adequately by two-dimensional idealizations. In particular, if the center of rigidity of the structure does not coincide with its center of gravity, torsional deformations will occur and it becomes important to account for three-dimensional behavior in the analysis. A variety of computer programs have been developed for the efficient elastic analysis of three-dimensional buildings, most notably TABS [3]. Also, several computer programs have been developed for inelastic seismic analyses of two-dimensional structures, in particular DRAIN-2D [1,2]. The program described in this report combines the features of TABS and DRAIN-2D, to produce

an efficient program for the inelastic analysis of certain types of three-dimensional buildings.

Although tall buildings are three-dimensional structures, they often have special features that can be used to reduce the computational effort below that required for a full three-dimensional analysis. One procedure is to idealize the structure as an assemblage of plane frames linked by rigid floor diaphragms, with no enforcement of compatibility for vertical and rotational displacements at joints common to two or more frames. Although it is known that this type of idealization may not be accurate when applied to framing systems of certain types (particularly framed-tube structures), it is a sound idealization for many structures and is in widespread use for linear elastic analysis.

The purpose of the research described in this report has been to develop a similar procedure for the dynamic analysis of inelastic building structures. This report describes the theory and computational procedure, and presents detailed instructions for use and extension of the computer program developed to apply the procedure.

1.2 REPORT LAYOUT

A review of the structural idealization is presented in Chapter 2. The computational procedure is presented in Chapter 3 and the program concepts in Chapter 4. The procedures to be followed in adding new structural elements are described in Chapter 5. A sample problem is presented in Chapter 6, and conclusions in Chapter 7. Appendix A constitutes a detailed User's Guide for the program.

2. STRUCTURE IDEALIZATION

Several authors have proposed methods of analysis which take advantage of the special features of typical building frames. Clough, King and Wilson [4] and Wilson and Dovey [5] treated the structure as an assemblage of plane frames linked by rigid floor diaphragms. This technique greatly reduces the computational effort, but does not enforce compatibility of the vertical and rotational displacements at joints common to two or more frames. Weaver and Nelson [6] used a complete three-dimensional approach to the analysis of buildings, assuming rigid floor diaphragms and a rectangular layout of framing members. Wilson, Hollings and Dovey [7] and Nair [8] also used a three-dimensional idealization, but improved the computational efficiency over that of a full three-dimensional analysis by allowing only displacements in the plane of the frame at joints which are not common to two or more frames.

A basic assumption of these methods is that each floor acts as a rigid horizontal diaphragm in its own plane, so that the horizontal displacements of all points in the diaphragm plane are uniquely determined by two translations and one rotation of each floor. In accordance with this assumption, the beams are assumed to bend only normal to the floor slab, and to have no axial strain.

The computer program TABS [3] and various programs based on Reference [4] are in widespread use by the engineering profession. Although it is known that they may not be accurate when applied to framing systems of certain types (particularly framed-tube structures), the results for most tall building structures are sufficiently accurate for use in design.

The idealization selected herein for the analysis of inelastic frame buildings is essentially identical to the TABS idealization. However, each individual frame is idealized as in the DRAIN-2D computer program [1], so that the resulting idealization allows rather greater generality than TABS. In addition, greater freedom is allowed in the positioning of the floor diaphragms.

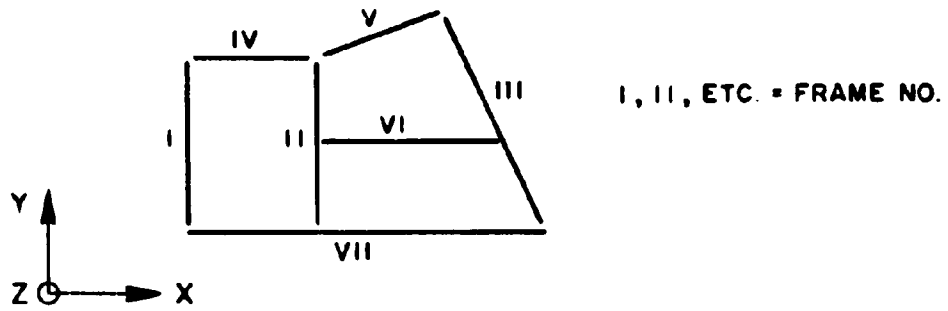
With reference to Fig. 2.1, the structural idealization can be summarized as follows.

- (a) The building must be separated into a series of discrete plane frames, connected together by rigid horizontal diaphragms, as indicated in Fig. 2.1(a). Each frame must be in the vertical plane, but may otherwise be essentially arbitrary. A typical frame is shown in Fig. 2.1(b). The frames may be arbitrarily oriented and located in plan.
- (b) Except for the common column approximation noted later, the only connection between frames is through the diaphragms. The diaphragms are assumed to be rigid and horizontal, but may otherwise be located arbitrarily. For example, there may be two or more independent diaphragms at any level, as shown in Fig. 2.1(b) for diaphragms 6 and 7. Some frames may be connected to a given diaphragm and others not connected, as in Fig. 2.1(b) where the frame shown is not connected to diaphragm 5.
- (c) The displacement degrees of freedom for any frame are organized into two groups namely, (1) internal degrees of freedom and (2) connected (or external) degrees of

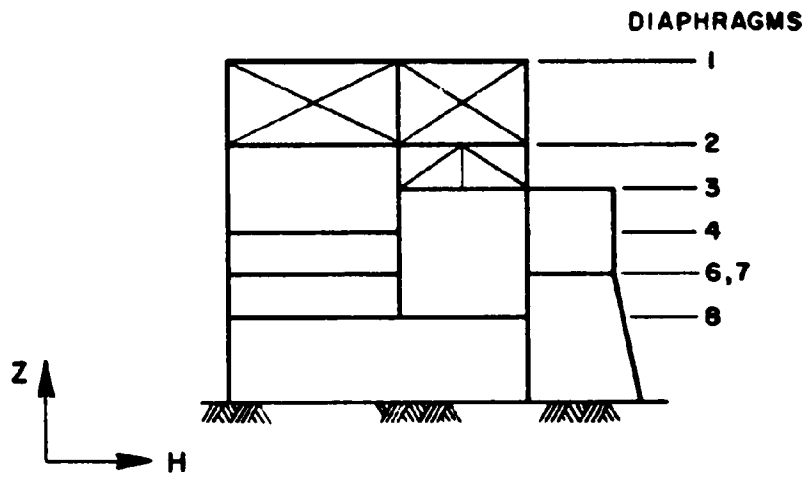
freedom. The horizontal displacements at those joints which connect to diaphragms are kinematically related to the diaphragm displacements, the relationship depending on the frame orientation and its location relative to the diaphragm center. These horizontal displacements are the connected degrees of freedom of the frame. All other displacements are internal degrees of freedom.

- (d) Compatibility of vertical and rotational displacements at joints common to two frames is not enforced. For joint rotations, if two frames intersect at right angles in plan view, then the rotations will actually be uncoupled provided the principal axes of the beams and columns lie in the planes of the frames. For other cases, however, the rotations will not be uncoupled, and hence the structural idealization is approximate. The assumption that common joints may have different vertical displacements, and hence that common columns may have different axial deformations in each frame, introduces a more serious approximation. To avoid the inconsistency of computing two or more different axial forces for a single column, an approximate correction for common-column effects is made, as described in the following paragraph. However, displacement compatibility is not enforced, and hence the idealization is not suitable for structures such as framed tubes, in which there is substantial coupling through common columns.

- (e) It is assumed that the axial forces in columns which are common to two different frames can be obtained by adding together the forces calculated for the two frames. This addition is carried out by the computer program and the combined force is used in assessing P-M interaction effects for common columns. For framing systems in which the computed behavior is likely to be affected little by axial deformations of the columns, this approach is believed to be a reasonable one. Because the dynamic analysis is performed step-by-step in a series of small load increments, consideration has been given to a successive corrections procedure in which vertical displacement compatibility is enforced by adding corrective nodal forces at each time step to eliminate the incompatibility while still satisfying equilibrium. At the time of writing, however, this procedure has not been incorporated into the computer program.
- (f) For dynamic analysis the mass of the structure may be lumped at the joints of the frames for vertical inertia effects, but should preferably be lumped entirely in the diaphragms for horizontal inertia effects. Lumping of the vertical inertias at the joints permits consideration of vertical as well as horizontal ground motions. The reason for lumping horizontal inertias into the diaphragm is explained subsequently.
- (g) The torsional stiffnesses of all members are ignored.



(a) PLAN OF BUILDING



(b) ELEVATION OF FRAME

FIG. 2.1 TYPE OF STRUCTURE TO WHICH PROGRAM APPLIES

3. COMPUTATIONAL PROCEDURE

3.1 GENERAL

Structures to which the proposed technique is applicable must be composed of structural components which are separated into a series of vertical frames tied together by rigid diaphragms. As indicated in Chapter 2, individual frames are idealized as in the DRAIN-2D computer program, except for the specification of nodes connected to the diaphragms. The solution procedure is, however, substantially more complicated than in DRAIN-2D, because of the use of substructure techniques to consider coupling of the frames through the diaphragms. The procedure is described in this chapter.

3.2 FRAME DEGREES OF FREEDOM

An elevation of a simple frame is shown in Fig. 3.1(a). The frame is idealized as a finite number of deformable elements (members), connecting a finite number of nodes (joints). The nodes are located in an H, Z coordinate system as shown, where H lies in the global X, Y plane and Z is vertical. Nodes may have finite dimensions, but are commonly considered only as points. The elements may be one- or two-dimensional. Loads may be applied to the frame directly at the nodes or through the elements. Element loads must be specified in terms of elastic fixed-end forces associated with each element. Lumped masses for consideration of vertical and rotational inertia effects may be placed at individual nodes. However, masses for consideration of horizontal inertia effects should preferably be lumped into translational and rotational inertias of the diaphragms.

For any frame, the degrees of freedom are typically the two translational and one rotational displacement at each node. These degrees of freedom permit extension, two modes of flexural deformation,

and three rigid body motions for each beam-type element. However, provision is made for degrees of freedom to be deleted, combined, or connected to the floor diaphragms, so that the total number of unknown displacements will usually be substantially less than three times the number of nodes.

The relationship between the potential displacements of the nodes (three at each node) and the actual degrees of freedom is defined by an Identifying (ID) Array. The procedure used to construct this array is similar to that for DRAIN-2D, but is extended to account for the connected degrees of freedom. The form of the ID array can be explained with reference to Fig. 3.1.

The degrees of freedom within any frame are basically numbered in the order H, Z and rotational displacement, respectively, at each node, starting with node 1 and continuing through the nodes in increasing numerical order. However, the numbering of the degrees of freedom will be affected by constraints of three types, namely zero displacement, identical displacement and connected displacement constraints, as follows. The constraints must be specified by the program user, but the numbering is carried out automatically by the computer program.

- (a) Zero displacements: The number zero is assigned to all degrees of freedom which are specified to have zero magnitudes. This will typically include displacements at "rigid" supports (nodes 9 through 12 in Fig. 3.1(a)).
- (b) Identical displacements: If displacements are constrained to be identical for a group of two or more nodes, the degree of freedom number is assigned for

the lowest numbered node of the group, and re-used for subsequent nodes. Horizontal displacements will typically be identical if extensions of a horizontal beam are to be ignored, and vertical displacements may be identical if column extensions are to be ignored. Rotational displacements might be specified to be identical in certain cases. In Fig. 3.1(a) the vertical columns are assumed to be inextensible, so that the vertical displacements at both ends of each such column have identical degree of freedom numbers.

- (c) Connected Displacements: The H displacements of all nodes which are connected to a given diaphragm will be identical, and must ultimately be related to the rigid body displacements of the diaphragm. These H displacements are numbered last within each frame. If there are k degrees of freedom for the frame, excluding connected degrees of freedom, then the H displacements for nodes connected to diaphragm n are assigned degree of freedom number $k + n$.

3.3 FRAME STIFFNESS

The frame stiffness matrix is assembled exactly as in the DRAIN-2D program, by establishing a Location Matrix, or "LM" array, for each element, relating the degrees of freedom of the element to those of the frame.

In expanded form, the assembled stiffness matrix for a typical frame will have the "arrow" form shown in Fig. 3.2, with the internal degrees of freedom corresponding to the shaft of the arrow and the connected degrees of freedom to the arrowhead. In

the computer program, only the top half of the stiffness is stored, columnwise in compacted form ignoring all terms up to the first nonzero term in any column.

3.4 FRAME LOAD VECTOR

For loads applied directly to the nodes of a frame, the frame load vector R , is assembled by use of the "ID" array. Loads originating within elements are assembled using the "iM" arrays. In either case, if a particular nodal displacement is fixed and has no degree of freedom, any corresponding loads are ignored. If two or more nodal displacements are assigned identical degrees of freedom, any corresponding loads are added together.

Loads may also be specified to act on the diaphragms, rather than on individual frames. The procedure for dealing with these loads is explained subsequently.

3.5 FRAME MASS MATRIX

The masses associated with a particular frame must be lumped at the nodal points, so that the frame mass matrix is diagonal. At the frame level, it is usually wise to specify only masses associated with vertical and rotational degrees of freedom, and to lump the masses for consideration of horizontal inertia effects into translational and rotational inertias of the diaphragms.

The reason for this is that the vertical and rotational displacements will always be internal degrees of freedom, whereas horizontal displacements may be connected to the diaphragms. During the analysis, horizontal inertia effects at connected nodes are transformed into equivalent horizontal and rotational inertias of the diaphragms. However, this transformation accounts only for inertia effects in the H direction for any frame, and not for inertia

normal to the plane of the frame. Hence, unless these normal inertias are accounted for as H inertias for the intersecting frames, some inertia effects may be lost. Hence, it is recommended that the program user lump all horizontal masses into translational and rotational inertias of the diaphragms, and specify them as diaphragm properties.

The frame mass matrix is assembled by use of the "ID" array. If a particular nodal displacement has no corresponding degree of freedom, any corresponding mass is ignored. If two or more nodal displacements are assigned the same degree of freedom, any corresponding masses are added together.

3.6 CONDENSATION OF FRAME STIFFNESS AND LOAD VECTOR

The equilibrium equations for a single frame can be written as:

$$\begin{bmatrix} \underline{K}_{II} & \underline{K}_{IE} \\ \underline{K}_{EI} & \underline{K}_{EE} \end{bmatrix} \begin{bmatrix} \underline{r}_I \\ \underline{r}_E \end{bmatrix} = \begin{bmatrix} \underline{R}_I \\ \underline{R}_E \end{bmatrix} \quad (3.1)$$

in which \underline{r}_I and \underline{r}_E are the internal and connected degrees of freedom of the frame, respectively; \underline{R}_I and \underline{R}_E are loads on the frame corresponding to \underline{r}_I and \underline{r}_E ; and \underline{K}_{II} , \underline{K}_{IE} , \underline{K}_{EI} and \underline{K}_{EE} are frame stiffness submatrices. The load vector and stiffness matrices are assembled as described in the preceding sections. Because the connected degrees of freedom are numbered last for each frame, the stiffness matrix storage within the computer conforms to the partitioned form of Eq. 3.1.

For each frame, a stiffness and load vector in terms of the connected degrees of freedom only are obtained by condensing out

the internal degrees of freedom, \underline{r}_I . In matrix form the condensation process is as follows:

$$\underline{r}_I = \underline{K}_{II}^{-1} \underline{R}_I - \underline{K}_{II}^{-1} \underline{K}_{IE} \underline{r}_E \quad (3.2)$$

Hence

$$\underline{\bar{K}}_{EE} \underline{r}_E = \underline{\bar{R}}_E \quad (3.3)$$

in which

$$\underline{\bar{K}}_{EE} = \underline{K}_{EE} - \underline{K}_{EI} \underline{K}_{II}^{-1} \underline{K}_{IE} \quad (3.4)$$

and

$$\underline{\bar{R}}_E = \underline{R}_E - \underline{K}_{EI} \underline{K}_{II}^{-1} \underline{R}_I \quad (3.5)$$

$\underline{\bar{K}}_{EE}$ and $\underline{\bar{R}}_E$ are the required condensed stiffness and load vector, respectively. Eq. 3.2 allows \underline{r}_I to be determined when \underline{r}_E is known.

In the computer program, the condensation is carried out by eliminating only the equations corresponding to \underline{r}_I in the equation solving process. The remaining portions of the stiffness matrix and load vector are then the required matrices $\underline{\bar{K}}_{EE}$ and $\underline{\bar{R}}_E$. The partially triangularized matrix also provides all the necessary coefficients to allow \underline{r}_I to be determined when \underline{r}_E is known.

3.7 BUILDING STIFFNESS

The condensed stiffness matrix for each frame is the frame stiffness in terms of the H displacements only. These condensed matrices can be combined to form a stiffness matrix for the complete building, in terms of the rigid body displacements of the diaphragms. To do this the condensed frame stiffnesses must be transformed to a common displacement system in terms of two translations and one

rotation of each diaphragm at the center of mass of the diaphragm. The centers of mass need not have the same X, Y coordinates for all diaphragms.

The transformation relating the connected degrees of freedom of a typical frame to the diaphragm displacement can be obtained with reference to Fig. 3.3. The transformation for diaphragm n and frame m is:

$$\underline{r}_{nm} = \langle \cos\alpha_m \quad \sin\alpha_m \quad -d_{nm} \rangle \begin{pmatrix} r_{xn} \\ r_{yn} \\ r_{\theta n} \end{pmatrix} \quad (3.6)$$

in which r_{nm} = H displacement for diaphragm n and frame m, and the remaining quantities are shown in Fig. 3.3. The transformation for a single frame and all diaphragms can hence be written as:

$$\underline{r}_m = [\cos\alpha_m [I] \quad \sin\alpha_m [I] \quad -[d_{nm}]] \begin{pmatrix} r_{x1} \\ r_{xn} \\ r_{y1} \\ r_{yn} \\ r_{\theta 1} \\ \vdots \\ r_{\theta n} \end{pmatrix} \quad (3.7)$$

$$\text{or } \underline{r}_m = \underline{A}_m \underline{q}$$

in which [I] is a unit matrix; $[d_{nm}]$ is a diagonal matrix containing the normal distances d_{nm} from the diaphragm centers to the frames; and \underline{q} is the vector of diaphragm displacements. The condensed frame stiffness transformed to the diaphragm system is therefore

$$\bar{\underline{K}}_m = \underline{A}_m^T \underline{K}_m \underline{A}_m \quad (3.8)$$

in which \underline{K}_m is the condensed stiffness matrix of frame m, in frame coordinates; and $\bar{\underline{K}}_m$ is the contribution of frame m to the building

stiffness. Eq. 3.8 can also be written as

$$\bar{K}_m = \begin{bmatrix} \cos^2 \alpha_m K_m & \cos \alpha_m \sin \alpha_m K_m & -\cos \alpha_m K_m [\delta_{nm}] \\ \hline & \sin^2 K_m & -\sin \alpha_m K_m [\delta_{nm}] \\ \hline \text{SYMMETRICAL} & & [\delta_{nm}] K_m [\delta_{nm}] \end{bmatrix} \quad (3.9)$$

The building stiffness matrix for all frames is assembled by simple addition of the individual frame stiffnesses. That is

$$K = \sum \bar{K}_m \quad (3.10)$$

It can be observed that if all frames are parallel to the X or Y axes, then the terms $\cos \alpha_m \sin \alpha_m K_m$ will be zero, so that it is not necessary to form this part of the matrix. Other off-diagonal submatrices of Eq. 3.10 may be zero because of symmetry in certain cases.

The building stiffness matrix is stored columnwise by the program as a compacted array, in order to minimize storage requirements.

3.8 BUILDING LOAD VECTOR

Loads may be applied to the individual frames or directly to the diaphragms. Loads applied to the frames are condensed during the static condensation process, as already described, and must then be transformed to the diaphragm system. The transformation is

$$Q_m = A_m^T R_m \quad (3.11)$$

in which Q_m is the contribution to the building load vector from the loads on frame m ; and R_m is the condensed load vector for frame m in frame coordinates.

The building load vector is formed by adding the frame contributions and any loads, \underline{F} , which are applied directly to the diaphragms. That is

$$\underline{Q} = \Sigma \underline{Q}_m + \underline{F} \quad (3.12)$$

3.9 BUILDING MASS MATRIX

Horizontal inertia effects at the building level are obtained from the specified translational and rotational inertias of the diaphragms. It is required that the diaphragm inertias be referred to the diaphragm centroids, so that the diaphragm mass matrix is diagonal.

3.10 SOLUTION FOR STATIC LOADS

The equilibrium equations for the building under static loading can be written as

$$\underline{Q} = \underline{K} \underline{q} \quad (3.13)$$

Eq. 3.13 is solved directly to give a vector of diaphragm displacements \underline{q} . For each frame the connected degrees of freedom, \underline{r}_m are computed from these diaphragm displacements, using the transformation of Eq. 3.7. To find the internal displacements of each frame, \underline{r}_1 , Eq. 3.2 is then used. In the computer program these internal displacements are computed by back substitution, beginning from the last internal degree of freedom in the partially reduced frame stiffness. The determination of member forces is then carried out exactly as in the DRAIN-2D program, using the state determination procedure described in Reference [1].

3.11 STEP-BY-STEP DYNAMIC ANALYSIS

The step-by-step procedure for dynamic response is essentially identical to that used in the DRAIN-2D program, except that the equation solving is modified to account for substructuring.

At the frame level, Eq. 3.1 is modified so that the stiffness matrix and load vector include contributions from the frame mass and damping matrices and the initial conditions at the beginning of the time step. The load vector also includes terms to correct for equilibrium unbalance at the end of the previous time step. The stiffness matrix is the tangent stiffness at the beginning of the time step.

The condensation of the frame stiffness matrix and load vector, and the transformation and addition of the condensed matrices to give the building stiffness and load vector, are carried out exactly as for static analysis. The building stiffness matrix and load vector are then augmented by terms depending on the diaphragm mass matrix and the diaphragm motions, and the equilibrium equations for the building are solved to give increments of diaphragm displacement. The diaphragm displacements are then transformed back to the individual frames, and the back substitution is completed. The state determination phase proceeds exactly as in DRAIN-2D.

3.12 GEOMETRIC STIFFNESS

In the elements developed to date for the DRAIN-2D program, it has been assumed that the geometric stiffness is zero for the static analysis, and remains constant throughout the dynamic analysis. The simple "truss bar" geometric stiffness has been used

for both truss elements and beam-column elements, using the axial force under static load. This assumption of constant geometric stiffness is reasonable for a typical plane building frame with vertical columns, because the total P-delta effect in any story for a single frame depends on the sum of the column forces in the story, and this sum remains constant if only horizontal ground motions and inertia effects are considered. The assumption of constant geometric stiffness is convenient computationally because it ensures that changes in element stiffness occur only when an element yields or unloads. If the geometric stiffness were to be changed continuously, then the structure stiffness would change every time step, and execution times would increase.

In a three-dimensional building, unfortunately, the P-delta effect in any story is no longer constant. This is because there are both translational and torsional P-delta effects in the story, and although the translational effects remain constant, the torsional effects do not. Nevertheless, in the current version of the computer program, the geometric stiffnesses are assumed to remain constant, as for a single plane frame. As a result, the torsional P-delta effect in any story will not be represented exactly at any instant of time. It is not clear at the time of writing whether this can introduce significant error.

It should be noted that the assumption of constant geometric stiffness is not essential, and that elements in which this stiffness varies from time step to time step could be developed if desired.

During the condensation of the frame stiffnesses and assembly of the building stiffness, the frame geometric stiffnesses are transformed into geometric stiffnesses for the building. Within

any frame, however, the geometric stiffness accounts for P-delta effects only in the H direction, and not in the direction normal to the plane of the frame. Hence, if the P-delta effect in the normal direction is not accounted for by intersecting frames, the building geometric stiffness will not be correct. Further, even with intersecting frames the geometric stiffness will be correct only for frames intersecting at right angles. This is a similar problem to that which arises with the transformation of horizontal inertia effects from individual frames to the complete building. However, whereas the problem can be overcome for inertia effects by lumping all horizontal masses into the diaphragms, no similar solution is available for geometric stiffness. Hence, the inclusion of P-delta effects by means of geometric stiffnesses is at best approximate.

3.13 COMMON COLUMNS

Provision is made in the program for the axial forces in common columns to be computed as the sum of the forces for corresponding elements in the intersecting frames. The option is currently available only for beam-column elements, but could be incorporated into any new elements developed for the program if desired. The option is also currently applied only for axial forces. However, storage is allocated for moment information as well as axial force information to be transferred between common columns, so that elements incorporating yield criteria with biaxial bending could conceivably be developed in the future.

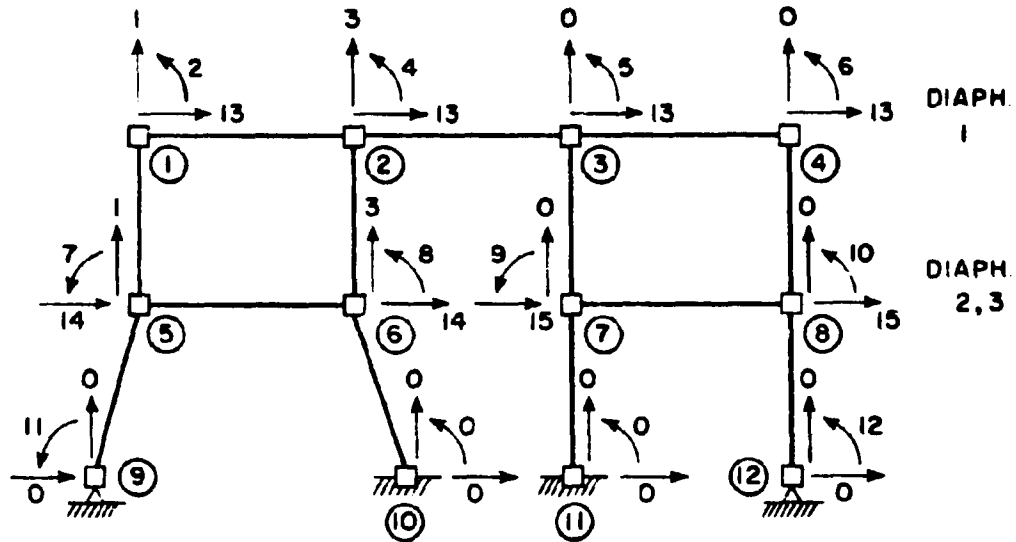
For initial force effects and static analyses, the axial forces for common elements are added, and the sum is then used for determining the geometric stiffness. Because initial and fixed end

forces are added, it is important to specify initial and fixed end axial forces for only one of the common elements, otherwise the combined value may be too large. For the static analysis and each step of the dynamic analysis, increments of axial force are computed and added to the existing force to give an updated value. When the yield criterion is being evaluated for a beam-column element, the axial force which is used is the value including the increment for that element and time step. If this is the first of two (or more) common elements to appear in the element sequence, then the axial force will not include the increments from the second (or later) common elements, and hence will be out of date by a portion of the step. Because the force increments within a time step should be small, this is not believed to be a serious error.

3.14 SOLUTION OF MODIFIED EQUILIBRIUM EQUATIONS

The equilibrium equations are solved using a subroutine derived from the efficient in-core equation solver OPTSOL [9]. If the response of any frame within a time step is such that no elements within the frame change their yield status (by yielding or unloading) then the triangularized stiffness matrix for that frame remains unchanged. If, however, status changes occur in one or more elements, a new triangularized form for the frame stiffness must be constructed. Because the change will generally not affect the entire stiffness matrix, computer time can be saved by carrying out triangularization operations on only that part of the matrix which changes. If the first degree of freedom to be affected by a change is in row j , then triangularization operations need to be carried out only on terms in and below row j . This is done automatically, subject to the following qualification.

Fig. 3.4 shows a frame stiffness, in which changes are to be made below row j . In the program, the un-triangularized frame stiffness is first modified by adding to it any changes resulting from status changes in the element. Then, if sufficient core storage is available to store the previously triangularized matrix, this matrix is set up, and all reduced coefficients above row j are transferred to the un-triangularized stiffness (the cross-hatched region in Fig. 3.4(a)). The triangularization then proceeds beginning at row j . If, however, there is insufficient storage to hold a duplicate frame stiffness, then the previously reduced coefficients must be recalled from disc rather than transferred from core. The triangularized stiffness is stored on disc columnwise as a single record. Hence, with a single READ statement it is possible to transfer only those reduced coefficients shown by cross hatching in Fig. 3.4(b). Thus, unless a more complicated reading procedure is used, it becomes necessary to triangularize all columns beyond column j . A more complicated reading scheme is believed to be undesirable, because the input/output cost is likely to exceed the cost of performing the extra triangularization operations. Hence, for the case in which a duplicate stiffness matrix cannot be held in core, the number of numerical operations required for re-triangularization is increased. The storage of duplicate stiffnesses is controlled by a user-selected storage option, as explained in Chapter 4.



(a) NODES AND DEGREES OF FREEDOM

NODE	H	Z	θ
1	13	1	2
2	13	3	4
3	13	0	5
4	13	0	6
5	14	1	7
6	14	3	8
7	15	0	9
8	15	0	10
9	0	0	11
10	0	0	0
11	0	0	0
12	0	0	12

(b) ID ARRAY

FIG. 3.1 CONSTRUCTION OF ID ARRAY

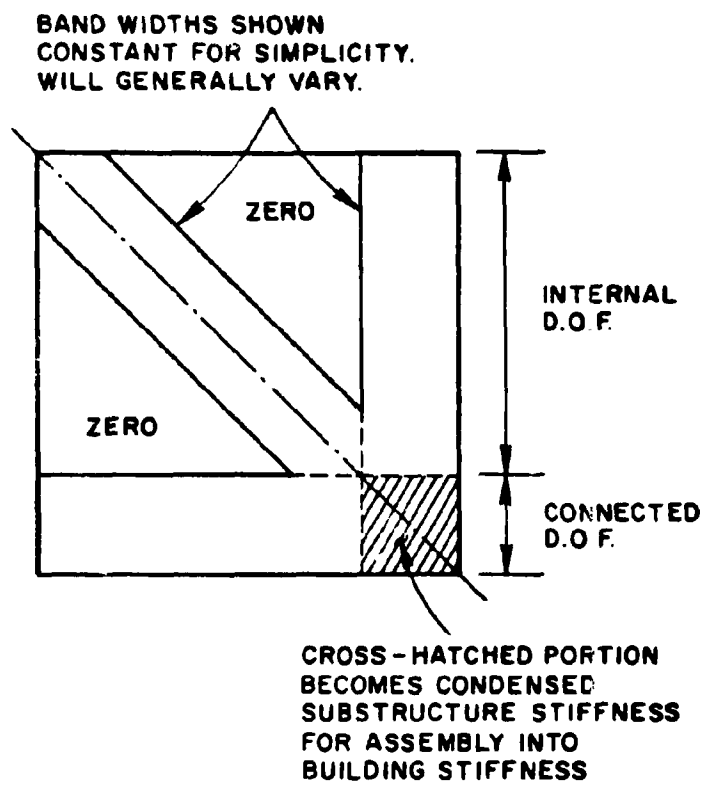


FIG. 3.2 FRAME STIFFNESS MATRIX

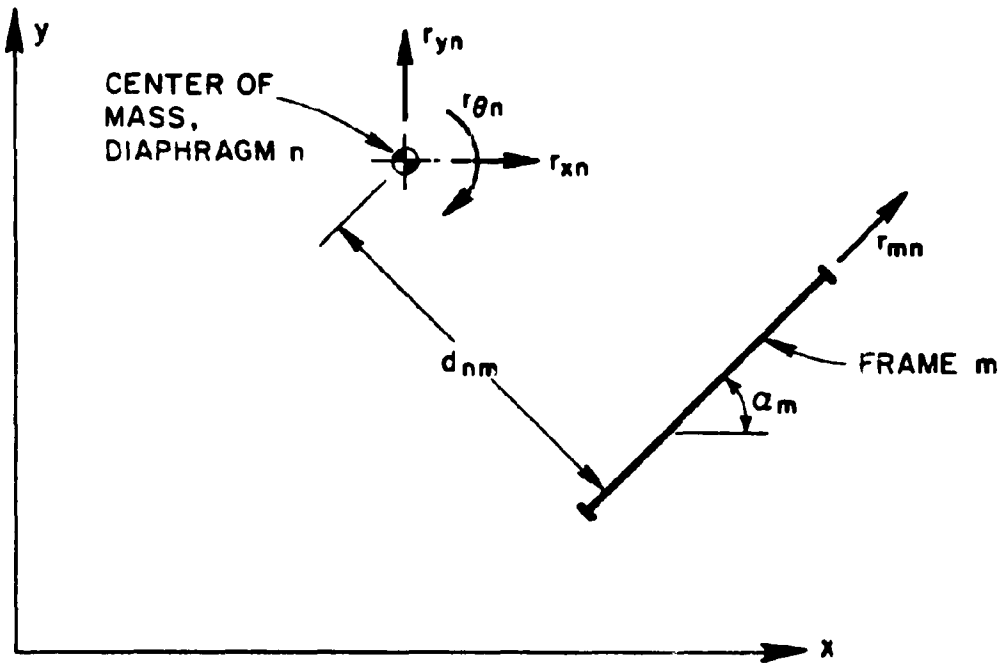
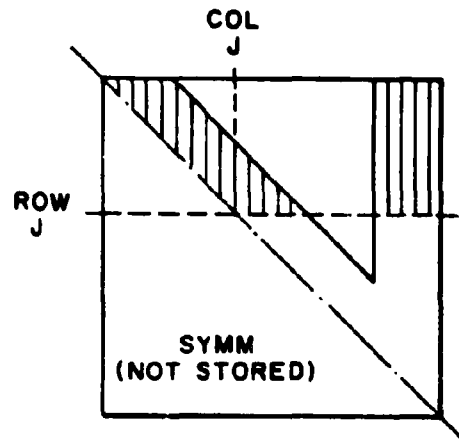
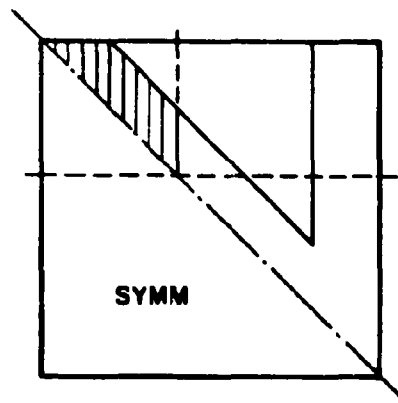


FIG. 3.3 DIAPHRAGM AND FRAME HORIZONTAL DISPLACEMENTS



(a) DUPLICATE STIFFNESS
IN CORE



(b) DUPLICATE STIFFNESS
NOT IN CORE

 = COEFFICIENTS WHICH DO NOT REQUIRE
RE-TRIANGULARIZATION

FIG. 3.4 SOLUTION OF MODIFIED EQUATIONS
FOR A FRAME

4. PROGRAM FEATURES

4.1 PROGRAM CONCEPTS

The program described in this report consists of a number of "base" subroutines which read and print the structure and loading data, allocate storage, carry out a variety of bookkeeping operations, assemble the frame and building stiffnesses and load vectors, solve the equilibrium equations, and determine the displacement response of the structure. This base program is combined with element subroutines to produce the complete program. Characteristics of the program are described in the following sections.

4.2 MODULARITY

The program is modular, so that subroutines for new elements can be developed and added with relative ease. All data input and output operations for the elements, and all element stiffness and state determination calculations, are carried out within the element subroutines. The requirements for transfer of information to and from the element subroutines, and the stiffness and state determination calculations, have been reduced to an essentially standard procedure which applies for elements of virtually all types.

4.3 STORAGE ALLOCATION

The program is organized to make efficient use of the available core storage. This has been achieved by using dynamic allocation for blank COMMON storage and by buffering data into large blocks for transfer to scratch (disc) storage. The required information for the computation is grouped in data blocks as follows. Arrays which may vary in size from problem to problem are stored in

blank COMMON, with the storage allocated at execution time.

- (1) General building data. This block contains information required to allocate storage and control the analysis, plus information on diaphragm motions, earthquake records and common columns.
- (2) Building stiffness matrix. This matrix is stored in compacted form, allowing for symmetry. With certain storage options (as described in the following Section) this core storage area may be shared with other data, the stiffness matrix being written on a scratch file and recalled as necessary. A duplicate stiffness matrix may also be stored, in order to reduce the execution time for re-triangularization of the stiffness as the structure yields.
- (3) Frame data. For each frame the following sub-blocks of data are stored. Depending on the storage option selected, the data for all frames may be held in core simultaneously, or for only one frame at a time.
 - (a) General frame data. This block contains information required to control the frame computations, plus information on the motions of the nodes.
 - (b) Frame stiffness matrix. This matrix is stored in compacted form. With certain storage options a duplicate frame stiffness matrix may also be stored, in order to reduce execution time.

- (c) Condensed frame stiffness matrix. The condensed frame stiffness is extracted from the partially reduced frame stiffness and copied into labelled COMMON/WORK/ to simplify the coding. This limits the number of diaphragms to a maximum of 61 for the currently assigned length of 2000 for this common block.
- (4) Element information. The element information is stored either in core or on scratch disc, depending on the problem size, the storage option selected by the user, and the available storage. If the data for all elements of all frames can be accommodated in core, then the data remains in core and no input/output operations to scratch storage are required. If storage on disc is necessary, the information is blocked to minimize the number of input/output operations. The storage used is that remaining after all other information has been stored in blank COMMON. The number of elements is not limited by the available core storage, except that the storage must be sufficient to hold the information for at least one of the elements requiring the largest number of data locations.

4.4 STORAGE OPTIONS

In order to provide both efficient computation for small problems and the capability to analyze large structures, a series of 10 different storage options is available to the program user. These options vary primarily according to whether or not the data for all frames are to be held permanently in core (so that input/output operations to scratch storage are minimized), and whether or

not duplicate stiffnesses are stored (so that the computational effort required to re-triangularize any modified stiffness matrices is minimized). Storage Option 1 requires most core but provides for the most rapid execution, whereas Option 10 requires the least core but may involve substantially larger execution times.

The storage options can be explained in terms of the following four storage codes.

(a) Storage Code A:

- (i) Code = 1 if the general frame data blocks for all frames are to be held in core simultaneously.
- (ii) Code = 2 if the general frame data block for only one frame at a time is to be held in core.

(b) Storage Code B:

- (i) Code = 1 if all frame stiffness matrices plus duplicates of all these matrices are to be held in core simultaneously.
- (ii) Code = 2 if all frame stiffness matrices are to be held in core simultaneously, but space is to be provided for only one duplicate matrix.
- (iii) Code = 3 if only one frame stiffness matrix plus one duplicate matrix are to be held in core at any time.
- (iv) Code = 4 if only one frame stiffness matrix is to be held in core at any time, with no provision to store a duplicate matrix.

(c) Storage Code C:

- (i) Code = 1 if the building stiffness matrix plus a duplicate matrix are to be held permanently in core.
- (ii) Code = 2 if the building stiffness matrix is to be held permanently in core, but no duplicate matrix.
- (iii) Code = 3 if the building stiffness matrix is to use the same storage area as the frame stiffness matrices.

The values of these codes are assigned by the program for each of the 10 different storage options as shown in Table 4.1.

For all storage options except Option 1 the stiffness matrices for different frames share the same storage area, and the allocated storage must be sufficient to accommodate the largest matrix. If the building has frames of different sizes, and if a storage option for which storage code B = 4 is selected, then for some of the smaller frames it may be possible to store both an original and a duplicate stiffness matrix in the storage allocated for the largest matrix. If this is so, then the program will automatically provide for a duplicate stiffness matrix for these smaller frames.

During data checking runs, the required data storage lengths for all 10 storage options are printed, to guide the program user in selecting an appropriate option.

4.5 OVERLAY STRUCTURE

The program is subdivided into overlays in order to reduce core storage requirements. The overlay procedure will operate on CDC 6400/6600 computers, and will require modification for other machines. However, because the overlay structure is simple, it is not anticipated that conversion will be difficult.

The overlay structure is as follows.

(0,0) overlay (i.e. main link): a small number of control subroutines.

(1,0) overlay (follows main link): subroutines for input of structure, frame and loading data.

(2,0) overlay (replaces (1,0) overlay): subroutines for execution of the step-by-step analysis.

(3,0) overlay (replaces (2,0) overlay): subroutines for organization of the time-history results.

Blank COMMON is allocated at the end of the program at each stage. Because the program length varies from stage to stage, the data in blank COMMON is automatically written to a scratch file at the end of each overlay, and recalled into its new location immediately after the next overlay is loaded.

4.6 FIELD LENGTH

The length of blank COMMON must be specified at execution time, and the program automatically sets the field length to the required length for each overlay. The subroutine which sets the field length on the CDC 6400 at the University of California, Berkeley, is subroutine MEMORY, which is system dependent. For other computers the call to MEMORY (from subroutine MEMRY) must be

changed to the appropriate system subroutine (e.g. XRFL for most CDC 6600 installations). This involves a change in only the MEMORY subroutine.

As previously noted, the required lengths of blank COMMON for all 10 storage options are printed by the program during data checking runs, so that the selection of an appropriate length for any execution run should not be a serious problem.

4.7 RANDOM ACCESS REQUIREMENT

At any time step of the dynamic analysis, it can be expected that only some (or none) of the frame stiffness matrices will need to be modified. Hence, for storage options in which the frame stiffnesses are stored on disc, it is inefficient to access the frames sequentially, because this would require reading and re-writing the stiffness matrices for all frames each time any one stiffness required modification. Hence, to reduce input/output cost, the frame stiffnesses are recalled from disc using a random access feature. This feature is system dependent, so that program changes will be needed to convert the program to other computer installations.

Details of the Berkeley random access routine (TSDISK) are given in Table 4.2.

For other computers it will be necessary to write four subroutines, namely OPTDISK, CLDISK, WRDISK, RDDISK, which will perform the same functions as described in this table. These subroutines should be simple, requiring probably only one executable statement.

Subroutines OPTDISK and CLDISK are called from Program SDRAIN in Overlay (2,0). The maximum record length (LSTR) required for the largest frame stiffness is available in SDRAIN, and can be used to open the random fileset if required by the specific computer. During data checking runs (which do not require random access), the required number of random access records and the maximum record length is printed. This information can be used for execution runs if special control cards are required to use random access.

Subroutines WRDISK and RDDISK are called from Subroutine SFRAME in Overlay (2,0). It is expected that it will be not necessary to change SFRAME, but merely to develop two simple subroutines. The present computer coding contains comment cards which specify the functions of these subroutines and show from where they are called.

4.8 REORGANIZED TIME-HISTORIES

During the step-by-step analysis, time-histories of results are generated for all nodes and elements at one step, followed by all nodes and elements at the next step, etc. The printout of the time-histories in this form is inconvenient, and hence provision is made for them to be reorganized to give all time steps for one node or element followed by all time steps for the next node or element, etc. This reorganization is coded using standard FORTRAN input/output commands, and requires a large amount of disc record manipulation. However, the program length in the reorganization phase (Overlay (3,0)) is short, so that peripheral processor costs in this phase will be low for those billing algorithms which base

PP cost on central memory requirements.

This phase of the program could undoubtedly be re-coded to provide more efficient execution on particular computer systems.

TABLE 4.1
DATA STORAGE OPTIONS

OPTION	1	2	3	4	5	6	7	8	9	10
CODE A	1	1	1	1	1	1	2	2	2	2
CODE B	1	2	2	3	4	4	3	3	4	4
CODE C	1	1	2	2	2	3	1	2	2	3

TABLE 4.2

CDC 6400 RANDOM ACCESS FEATURE

I9 CAL TSDISK: FEB 1972

IDENTIFICATION

Code Designation:	I9 CAL TSDISK
Title:	Random Access Disk Input/Output Subroutine
Author:	Thos Sumner, Computer Center, University of California, Berkeley
Date:	Revision: February 1972
Environment:	Machine: CDC 6000 Series Operating System: CALIDOSCOPE Coding Language: COMPASS Linkage Convention: <u>RUN2</u> Fortran
Program Name:	TSDISK
Entry Points:	CLDISK, LRDISK, MTDISK, OPDISK, RDDISK, WRDISK, NMDISK
Availability:	CLDR System Library (Public Fileset SUBRLIB)

PURPOSE

To provide random access disk storage for the Fortran programmer.

USAGE

The user must provide an index array whose size is greater than the number of distinct records to be written on the random access fileset. The fileset RANDOM is normally used by TSDISK and should not be accessed by any other routine. TSDISK may only be used to read a fileset which it has created.

To open fileset RANDOM

and provide access to the index array by TSDISK, OPDISK is normally called exactly once before any other call is made to TSDISK in the job step.

```
CALL OPDISK(N,INDEX)
```

N - the dimension of the array INDEX, $N \geq 2$. At most N-1 indexed records (identified by the integers from 1 to N-1) may be written on RANDOM. If RANDOM was written in a previous job or job step, N must be at least equal to the value used when it was written.

INDEX - the array declared in the user's program to hold the random file index.

TABLE 4.2
(CONTD)To write on fileset RANDOM

CALL WRDISK(IW,ARRAY,KW)

IW - the record identification number for the record to be written. IW is an integer and $1 \leq IW \leq N$. The values allowed for IW may, but need not, be used in their natural order (IW = j, followed by IW = j+1).

ARRAY - the first word of the block of data to be stored on the disk as the record identified by IW.

KW - the number of words to be written in the record identified by IW.

If a record has already been written for the given value of IW, it will be superseded by the new record, and the disk space used by the old record will be reused provided the new record is not longer than the old one. If the new record is longer, the disk space occupied by the old record is wasted.

To read from fileset RANDOM

CALL RDDISK(IR,ARRAY,KR)

IR - the identification number of the record to be read.

ARRAY - first word of the block into which the data in the indicated record are to be read.

KR - the number of words to be read from the indicated record. This must be less than or equal to the number of words written in the record.

To close the fileset RANDOM

CALL CLDISK

This call is used to close the fileset RANDOM after it is written if it is to be preserved for use in a subsequent job or job step. CLDISK need not be called if RANDOM is only being read in the current job step.

5. ADDITION OF ELEMENTS TO PROGRAM

5.1 GENERAL

New elements may be added to the program by developing five subroutines for each element, as described in this chapter.

Information is transmitted to and from the subroutines through short argument lists and labelled COMMON blocks. The procedures which must be observed are described in the following sections. These procedures are essentially the same as those for the DRAIN-2D program, but are extended to include changes since Reference [1] was published.

The five subroutines required are as follows. The "X" at the end of the subroutine name must be a number (currently from 1 to 10) to identify the element type.

- (1) INELX: Input and initialization of element data.
- (2) STIFX: Calculation of element tangent stiffness.
This may be termed the "linearization" phase.
- (3) RESPX: Determination of increments of element deformations (strains) and forces (stresses), determination of yield status, and output of time history results. This may be termed the "state determination" phase.
- (4) OUTX: Output of final envelope values for element deformations and forces.
- (5) THPRX: Output of re-ordered time histories for element results.

5.2 CONTROL INFORMATION

In the input data for each frame, the elements must be arranged in groups such that all elements within each group are of the same type. The base program then establishes certain variables to maintain control over the subroutine calling sequence and other parts of the computation. The control variables are as follows. The procedures for using them are described subsequently.

- (1) KCONT: KCONT (1) contains the element type number for the group of elements currently being input. KCONT (2) contains the number of elements in this group. KCONT (3) through KCONT (10) may contain other integer information on the element group, as desired by the programmer. The use of this array is explained in Section 5.8.
- (2) FCONT: This array is similar in purpose to KCONT, but contains real information on the element group, up to a maximum of 3 values.
- (3) NN: An undimensioned variable, equal to the total number of nodes in the frame.
- (4) NINFC: NINFC (N) contains the number of words of information stored for each element in the Nth group of elements. This number is the length of the labelled COMMON block INFEL created for elements of this type, as described in Section 5.3. NINFC (N) is transferred to subroutines INELX, STIFX, RESPX and OUTX through the argument lists, and is treated within these subroutines as an undimensioned variable.

- (5) NDOF: NDOF (N) contains the number of element degrees of freedom for each element in the Nth group. NDOF (N) is transferred to subroutines INELX, STIFX and RESPX through the argument lists as an undimensioned variable.
- (6) MSTEP: An undimensioned variable set equal to the step number in the step-by-step integration sequence. For the static load case, MSTEP = 0.
- (7) KPR: An undimensioned variable which indicates whether or not time history information for the current element is to be printed.
- (8) KST: An undimensioned variable which informs the base program whether or not the tangent stiffness matrix has changed for the current element at the current time step. Each time an element stiffness changes, the frame and building stiffnesses are modified as necessary.
- (9) KBAL: An undimensioned variable which informs the base program whether or not unbalanced loads have developed within the current element at the current time step. Any unbalanced loads must be assembled into the frame load vector for the next time step, to re-establish equilibrium. In general, unbalanced loads will be produced within an element at the same time that changes occur in the element tangent stiffness. However, separate indicators (KST and KBAL) are used to permit the programmer to suppress either the stiffness change or the unbalanced load correction if this is desired.

- (10) ITHP, NELTH, ISAVE, ISE, NF7: Variables for control of the reorganized time-history printout. Their meanings and uses are explained in Section 5.4.

5.3 ELEMENT INFORMATION BLOCK

For each element a block of information must be created and continually updated. All information to be retained for any element must be contained within this block.

For the purposes of adding element subroutines to the program, the element information is contained within the labelled COMMON block INFEL. This block may be divided in any way the programmer wishes, subject to the following restrictions.

- (1) The maximum length currently permitted is 200 words. This could easily be changed.
- (2) The first variable must be the number of the element within the element group. This is a single integer variable, for which the name IEL is suggested. The value of this variable is set in the INELX subroutine, and must not be changed subsequently.
- (3) The second variable must be KST, which must be set as explained subsequently.
- (4) The third variable must be the "LM" matrix for the element. This is an integer array with a dimensioned length equal to the number of element degrees of freedom. The name LM is suggested. The values in the array must be set in the INELX subroutine, as explained subsequently.

The labelled COMMON block INFEL must appear within subroutines INELX, STIFX, RESPX and OUTX. During the execution of subroutine INELX, the information in this block must be initialized for each element. This initialized information is then transferred, by the base program, to a scratch storage file. After all elements have been considered, the information is brought back into core for all elements and stored compactly in the available storage. If this storage becomes filled, the elements stored constitute an output block. This output block is written on a scratch file, and a new output block begun. At each cycle of the subsequent calculation, the output blocks must be returned to core, updated and rewritten on scratch files. The blocking and input/output operations are all carried out by the base program. If the information for all elements can be held in core, then there is no input/output of output blocks to scratch storage, and input/output costs are reduced.

Within subroutines STIFX, RESPX and OUTX, the element information is not available directly within COMMON block INFEL, but is stored in an array COMS transferred through the argument list. The address assigned to COMS in the base program is the address of the first word of the data for the corresponding element in the output block storage area. In order to transfer the data to INFEL, statements essentially as follows should appear at the beginning of each of the STIFX, RESPX and OUTX subroutines.

```
COMMON/INFEL/IEL, KST, IM(..), . . . .
DIMENSION COMS (1), COM (1)
EQUIVALENCE (IEL, COM (1))
DO 10 N = 1, NINFC
10 COM (N) = COMS (N)
```

These statements transfer the element information from the output block storage area to COMMON block INFEL. Note that because COMS and COM are both real variables, the transfer of data from COMS to COM is made without mode change. This is, it is immaterial whether the data being transferred is ultimately to be interpreted as real or integer.

At the end of the subroutine RESPX, any updated element information must be transferred back to the output block storage area. This may be achieved by the following statements

```
DO 100 N = 1, NINFC
100 COMS (N) = COM (N)
```

However, it is common for only a part of the element information to be changed, and hence it may be more efficient to transfer the modified data selectively. For example, if only the 23rd through 30th items in the INFEL block have been changed, the coding could be

```
DO 100 N = 23, 30
100 COMS (N) = COM (N)
```

Similarly, in transferring data from the output block storage area to the INFEL block the programmer may wish to save computer time by transferring only selected words. In some cases, it may even be desirable to use and modify the data with direct reference to variable COMS, without transfer to the INFEL block. In general, however, the coding of the element subroutines is greatly simplified if appropriate variable names are created within the INFEL block. The computer time required to transfer the data within core should be a relatively small part of the total execution time. Also,

because the arrays in INFEL are in fixed storage, whereas the address of COMS is transferred through the argument list, the compiled code for reference to arrays within the INFEL block is likely to be more efficient than if reference were made directly to COMS. This greater efficiency should compensate for the time required to transfer the data.

The information in the INFEL block may be updated only in the RESPX subroutines. That is, subroutines STIFX may use the data but may not change it. This restriction is also present in DRAIN-2D, and is imposed to reduce the number of input/output operations to scratch storage for cases in which the information for all elements cannot be held in core.

5.4 TIME HISTORY BLOCK

The labelled COMMON block THIST is used to control re-ordering of the time history results. This block is as follows.

```
COMMON/THIST/ITHOUT(10), THOUT(20), ITHP, ISAVE, NELTH, NSTH, NF7, ISE
```

Variables ITHP and ISE are set from user-specified input data, as follows.

- (a) ITHP: An undimensioned variable which indicates whether time history printout is required (1) only as the computation progresses (i.e. not re-ordered), (2) both as the computation progresses and at the end of the computation (i.e. re-ordered), or (3) only at the end of the computation. ITHP is set to values 0, 1 and 2, respectively, for these cases.

- (b) ISE: Tape storage code, to indicate whether or not the re-ordered time histories are to be saved on storage unit TAPE7 for subsequent off-line processing (e.g. plotting using user-developed plot routines).

The remaining variables are set or used as follows:

- (c) NF7: Storage unit number, currently set to 7.
- (d) NELTH: This indicator counts, for each frame, the number of elements for which time history results are requested. It is initialized to zero for each new frame by the base program and incremented in the INELX subroutines as explained in Section 5.8.
- (e) ITHOUT (10), THOUT(20): Areas used to transfer time-history data from the RESPX subroutines for subsequent re-ordering. The procedure is explained in Section 5.10.
- (f) ISAVE: This variable informs the base program whether or not a re-ordered time history printout is required for the current element. It is set to zero by the base program for each element before entry to RESPX, and must be set within RESPX as explained subsequently.
- (g) NSTH: The variable counts the total number of time steps in the time-history printouts. It is set automatically by the base program.

The values of the variables in the THIST block must not be set in the element subroutines except as indicated in the subroutine descriptions.

5.5 WORK BLOCK

Labelled COMMON block WORK provides a work area for use by the programmer. The block currently has a maximum length of 2000 words, and may be subdivided in any way.

The WORK block should not be used to transfer data between subroutines, because it will generally be used differently for different elements, and the data will therefore be continually changed. Data stored in this block for one element in subroutines INELX and OUTX is available for use by any subsequent element in the same group for the same subroutine at the current time step, but not in any other subroutine or at any other time step.

5.6 OTHER ARRAYS

The following arrays are generated by the base program, and transferred to some or all of the element subroutines.

- (1) X, Y: X(I), Y(I) contain the H, Z coordinates, respectively, of node I.
- (2) ID: ID(I,1), ID(I,2) and ID(I,3) contain the numbers of the frame degrees of freedom corresponding to the H displacement, Z displacement and rotation, respectively, at node I. For computational convenience in the base program, a value equal to one larger than the total number of structure degrees of freedom is assigned if the corresponding displacement is constrained to be zero. This is not important from the programmer's point of view.

5.7 COMMON COLUMN PROCEDURE

Storage with a length of $NCOLC * NESPL * NFORT$ words is reserved at the beginning of blank COMMON, for transfer of common column forces and related information, where:

$NCOLC$ = Total number of common column lines in building;

$NESPL$ = Largest number of elements in any common column line,

and

$NFORT$ = Maximum number of items (forces, moments, or displacements) to be transferred between common columns.

$NCOLC$ and $NESPL$ are required to be specified by the program user. The value of $NFORT$ is set equal 2 by the base program for the current version of the program, corresponding to the axial forces at ends i and j of a beam column element (these forces may be different because of the initial and/or fixed end forces). The value of $NFORT$ could be modified if more column actions were to be transferred. The storage area is initialized to zero by the base program, corresponding to zero force values.

In addition the following labelled COMMON block is established:

```
COMMON/CMCOL/NCOLC, NESPL, NFORT, NKDATA, CMCOL(6)
```

The variables $NCOLC$, $NESPL$ and $NFORT$ are specified as indicated above. The variable $NKDATA$ is a data checking variable which is initialized to zero by the base program and should be set to 1 in $INELX$ if an error in the common column specification is found, so that only a data checking run will be performed. $CMCOL(6)$ is an array that may be used in any way desired by the programmer.

The base program merely reserves the storage area, initializes it to zero, and establishes the variables NCOLC, NESPL, NFORT and NKDATA. All other computations are performed in the element subroutines. The procedure for beam-column elements (Type 2) is as follows. Other elements will be similar.

- (a) For each beam-column element which is also a common column element, the number of the common column line (NCOLL) and the story number within the common column line (NSCL) must be input (see User's Guide).
- (b) NFORT words of common column space are available for each pair of values of NCOLL and NSCL. The starting address of this space within blank COMMON is computed in the INEL2 subroutine as:

$$ICB = (NCOLL-1)*NESPL*NFORT+(NSCL-1)*NFORT+1$$
and its value is stored in COMMON/INFEL/, so that it is available for subroutines STIF2, RESP2, and OUT2.
- (c) In subroutine INEL2, the initial forces and fixed end axial forces at ends i and j are added to locations ICB and ICB+1 in blank COMMON.
- (d) At each step of the analysis in the RESP2 substructures the axial force increments are added to locations ICB and ICB+1 in blank COMMON, with appropriate signs. These updated values of axial force are then used for the state determination computations and time-history output in subroutine RESP2, and also for geometric stiffness determination in subroutine STIF2.

5.8 SUBROUTINE INELX

The argument list for this subroutine is as follows:

KCONT, FCONT, NDOF, NINFC, ID, X, Y, NN

The following arrays must be dimensioned:

KCONT(10), FCONT(3), ID(NN,3), X(NN), Y(NN), COM(200)

Dummy dimensions may be used if desired. (e.g., ID(NN,1),
Y(1))

Labelled COMMON blocks THIST and INFEL must be defined.

Labelled COMMON block CMCOL and blank COMMON must be used for common columns. Labelled COMMON block WORK may be used if desired. The variable COM(1) must be equivalenced to the first variable (IEL) of the INFEL block. If element fixed end forces are to be specified, a one-dimensional array DD with a dimension equal to NDOF for this element type must be set up (typically within/WORK/). This array is used to set up the fixed end forces in the H, Z coordinate system, as explained subsequently. An INELX subroutine is called once for each group of elements, early in the execution. The purpose of the subroutine is to read the input data for the elements in the group, and initialize a number of variables.

The variables ID, X, Y and NN are established within the base program before INELX is called, and must not be modified. Variables NDOF and NINFC are to be set within subroutine INELX. Variables KCONT and FCONT are established within the base program by reading the first data card of each element group on a (10I5, 3E10.0) format, one field for each term in KCONT and FCONT. This first data card of each element group must contain the following information:

Columns 1 - 5: A number (1 through 10) identifying the type of element in the group. For example, if 4 is entered, the subroutines called for this group will be named INEL4, STIF4, RESP4, OUT4, and THPR4.

6 - 10: Number of elements in group.

11 - 50: Up to 8 fields, each I5, containing any other integer information on the group that the programmer may wish to store in KCONT. In the elements developed to date, these fields have been used for entry of the number of element stiffness types, number of fixed end force patterns, etc.

51 - 80: Up to 3 fields, each E10.0, containing any real information on the group that the programmer may wish to store in FCONT.

All subsequent data cards for the element group are read within INELX, and may be of any type desired by the programmer.

The following steps must be performed within the subroutine.

- (1) Set the values of NDOF and NINFC.
- (2) If desired, establish reference tables of stiffnesses, strengths, fixed end forces and initial forces for subsequent use in specifying the individual elements. The

storage arrays should preferably be defined within COMMON block WORK, to avoid requesting new storage.

- (3) Specify the properties of each element in the group. The element node numbers and appropriate stiffness, strength and fixed end force data must be specified. Use may be made of any reference tables established in Step 2. Generation options may be incorporated, provided the elements are generated in element number sequence and information for only one element at a time is stored in COMMON block INFEL.

- (4) For each element, the following initialization operations must be carried out.

- (a) Set up the LM array within COMMON block INFEL. This is easily done, with reference to the element node numbers and the ID array.

- (b) Initialize all element information variables within INFEL to appropriate values corresponding to the initial state of the structure. Set IFEL to the element number within the group and set KST to zero.

- (c) Call subroutine BAND with the statement

CALL BAND

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

- (d) Set up the element fixed end forces, if any, in array DD. These forces must be in the H, Z coordinate system for the frame, and must be the forces acting on the element necessary to clamp its ends. The element

internal forces must also be initialized to their values in the clamped condition. Call subroutine SFORCE with the statement

```
CALL SFORCE (DD)
```

This subroutine transfers the fixed end forces from DD to the static load vector for the frame. This call should be omitted if there are no fixed end forces.

(e) Establish, within COMMON block INFEL, a variety of indicators to permit the current status and any status changes of the element to be monitored. Also set a variable, KGEOM, to a nonzero value if a geometric stiffness is to be computed (otherwise zero) and a variable KOUTDT to a nonzero value if time history results are to be printed (otherwise zero).

(f) Add one to variable NELTH whenever the value of KOUTDT for an element is nonzero. This counts the number of elements in the current frame for which time histories are requested. The value of NELTH is initialized to zero by the base program for each new frame.

(g) Because arrays X and Y will not be available at later times in the execution, save information on the element size and orientation in space, so that subsequent stiffness and state determination calculations can be carried out. A convenient means of storing element orientation information is to retain a displacement transformation relating nodal displacements and

element deformations, although other procedures may be used.

(h) Call subroutine FINISH with the statement

```
CALL FINISH
```

This call must be made after the INFEL block for the element has been fully initialized. The subroutine transfers the data from INFEL to scratch storage.

Ultimately the data is brought back into core by the base program and compacted into output blocks.

(i) If this is a common column element, initialize the axial force values for common columns. First, calculate the location, ICB, of the force values in blank COMMON as

$$ICB = (NCOLL-1)*NESPL*NFORT+(NSCL-1)*NFORT+1$$

where the variables are defined in Section 5.7.

Second, add any initial and fixed end axial forces at ends i and j of the element into blank COMMON locations ICB and ICB+1, respectively. It is also advisable to store ICB in the INFEL block to avoid recalculating it in the STIFX and RESPX subroutines. If an error is detected in the specification of the column line and story number, print an error message and set NKDATA = 1.

5.9 SUBROUTINE STIFX

The argument list for this subroutine is as follows:

```
MSTEP, NDOF, NINFC, COMS, FK, DFAC
```

The following arrays must be dimensioned:

```
COMS(200), FK (NDOF, NDOF), COM(200)
```

Dummy dimensions may be used if desired.[†] Labelled COMMON block INFEL must be defined, and block WORK may be defined if desired. The variable COM(1) must be equivalenced to the first variable (IEL) of the INFEL block.

The purpose of this subroutine is to return, in array FK, the change in element stiffness. The subroutine is called once in each time step for each element which changes its status, except for the cases MSTEP = 0 and MSTEP = 1, when it is called once for each element as explained below.

The size of the FK array is currently limited, by a DIMENSION statement in subroutine STIFF of the base program, to a maximum size of 10 x 10.

The following operations must be performed within the subroutine.

Case 1: MSTEP = 0.

Form the initial elastic stiffness of the element, and return in array FK.

Case 2: MSTEP = 1.

(a) If the geometric stiffness of the element is to be included (KGEOM \neq 0), form this stiffness and return in array FK. If this is a common column element, use the axial forces scored in blank COMMON addresses ICB and ICB+1.

(b) To consider $\beta_0 k_0$ damping, augment the element stiffness, k_0 , by the amount DFAC * k_0 . The value of DFAC is set in the base program. It may be noted that it would be possible to specify, for each element group or even for each element, a scaling factor to be applied to β_0 , allowing it to vary from one element to the next.

such an option has not been included in the element subroutines developed to date, but could easily be added if needed.

Case 3: MSTEP > 1.

Form the change in stiffness and return in array FK. The status codes for the previous and current state of the element, as established in subroutine RESPX, should be used to determine the nature of the stiffness change. Set the previous code equal to the current code.

In each case, appropriate information must be transferred from the storage array COMS to the INFEL block. No modification of data in the INFEL block is allowed in this subroutine.

5.10 SUBROUTINE RESPX

The argument list for this subroutine is as follows:

NDOF, NINFC, KBAL, KPR, COMS, DDISM, DD, TIME, VELM, DFAC, DELTA

The following arrays must be dimensioned:

COMS(200), DDISM(NDOF), DD(NDOF), COM(200), VELM(NDOF)

Dummy dimensions may be used if desired. Labelled COMMON blocks THIST and INFEL must be defined. Block WORK may be defined if desired, and block CMCOL must be used for common columns. The variable COM(1) must be equivalenced to the first variable (IEL) of the INFEL block.

The purpose of the subroutine is to determine the increments of element force and deformation, check the yield status for the element, set the status codes as necessary, and determine any unbalanced forces resulting from overshoot or reversal. The subroutine is called once for each element at each time step.

The following steps must be carried out within the subroutine:

- (1) Transfer the element data from the storage array COMS to the INFEL block.
- (2) The increments of nodal displacement are transferred from the base program in array DDISM. Using these increments, determine the increments of element deformation and element forces. The general procedure is described in Chapter 3 of Reference [1]. Common column elements must be treated as described in Section 5.7.
- (3) The value of KST is set to zero in the base program. If the status of the element changes, so that its stiffness must be modified in the next time step, set $KST = 1$. The values of the status indicators for the current state must also be set. These will be compared, in subroutine STIFX, with the values for the previous state to determine the nature of the stiffness change.
- (4) The value of KBAL is set to zero in the base program. If unbalanced loads are developed because of overshoot or reversal, set $KBAL = 1$ and add the magnitudes of the unbalanced loads to array DD. These unbalanced loads, and any damping loads as described in the following sections, must be the loads to be assembled directly into the structure load vector for the next time step, without transformation or change of sign. The theory is explained in Reference [1] and the details

can be inferred by studying the subroutines for the existing elements. If KBAL remains zero, no data need be stored in array DD.

- (5) The nodal velocities are transferred from the base program in array VELM and may be used to obtain element deformation increments from DDISE. To consider original stiffness proportional damping determine a load term equal to $DFAC * \underline{k}_0 * \dot{\underline{r}}$, where DFAC is set in the basic program and transmitted as argument; \underline{k}_0 is the original element elastic stiffness matrix; and $\dot{\underline{r}}$ the deformation rate vector at the beginning of the time step. Add this load to the vector DD. Set KBAL = 1.
- (6) To consider "structural" damping the parameter DELTA is used. This parameter is defined by the user and read by the base program. A set of damping forces must be computed at the end of each integration time step, as follows.

The damping force generated in, for example, a truss bar is

$$S_d = - DELTA * |S| \frac{|\dot{V}|}{\dot{V}}$$

in which S_d = damping axial force; S = current force (which for an elastic element will be proportional to the extension); and \dot{V} = rate of extension. The damping forces for other elements are determined similarly. Transform the force to correspond to the nodal displacements and add to DD. Set KBAL = 1.

- (7) Retain appropriate envelope values of element forces and deformations for subsequent printing. The information retained will depend on the type of element.
- (8) The value of KPR is set to the element group number in the base program if time history results are to be printed or saved for the current time step, otherwise it is set to zero, except for the static load case where it is set to minus the group number. If $KPR < 0$, then this is the static load case and results for all elements should be printed. If $KPR > 0$, this is an element for which time history data is required. If $ITHP \leq 1$, this is an element for which a time history is required as computation progresses. Care must be taken in printing the headings for time history data. A heading may be printed when $IEL = 1$, but there will be no corresponding data unless a time history print is required for at least one element in the group. Hence, it is recommended that a heading indicator, IHED, be established within COMMON block WORK, and initialized to zero when $IEL = 1$, $KPR \neq 0$ and $ITHP \geq 1$; or to 1 when $IEL = 1$ and either $KPR = 0$ or $ITHP < 1$. Then, when an element is encountered for which a time history print is required (i.e. $KOUTDT \neq 0$), print the heading and set $IHED = 1$ if $IHED = 0$, otherwise do not print the heading. The existing element subroutines use this procedure.

- (9) If the value of $ITHP \geq 1$ and $KOUTDT \neq 0$, this is an element for which a reorganized time history is to be printed at the end of the computation. The results for this element must therefore be saved on a scratch file, for subsequent re-ordering by the base program and printout in subroutine THPRX. The arrays ITHOUT(10) and THOUT(20) in COMMON block THIST are provided for transfer of results to the base program, which then writes the data on the scratch file. The following rules must be followed in setting up these two arrays.
- (a) The value of ITHOUT(2) must be the element group number, because it will be used by the base program for final printing in subroutine THPRX.
 - (b) The remainder of ITHOUT and all of THOUT can be used to transfer any desired information needed for the reorganized time history printout, with integer information in ITHOUT and real information in THOUT.
 - (c) Set ISAVE = 1, to indicate to the base program that the arrays are to be saved. ISAVE is initialized to zero by the base program for each element at each time step.
- (10) Transfer the updated element data from the INFEL block back to the storage array COMS.

5.11 SUBROUTINE OUTX

The argument list for this subroutine is as follows:

COMS, NINFC

The following arrays must be dimensioned:

COMS(200), COM(200)

Dummy dimensions may be used if desired. Labelled COMMON block INFEL must be defined, and COMMON block WORK may be used if desired. The variable COM(1) must be equivalenced to the first variable (IEL) of the INFEL block.

The purpose of the subroutine is to print final results for each element. The subroutine is called once for each element, at user specified time intervals. The information in the INFEL block must not be modified in this subroutine.

The following steps must be carried out within the subroutine:

- (1) Transfer the element data from the storage array COMS to the INFEL block.
- (2) If IEL = 1, print appropriate output headings for the element group.
- (3) Print the results, typically envelope values of forces and deformations, for each element.

5.12 SUBROUTINE THPRX

The argument list for this subroutine is as follows:

NS, IFR

The labelled COMMON block THIST must be defined.

The purpose of the subroutine is to print reorganized time history results for the specified elements. This subroutine is called after all other computation is complete.

The element time history results are recalled from a scratch file by the base program, and transferred through arrays ITHOUT and THOUT, exactly as these arrays were created in the corresponding subroutine RESPX.

The variable NS is set by the base program to the output interval number, starting with 1 for the initial static condition, and may be used in the printout if desired. The subroutine is called once for each output interval.

The variable IFR is the frame number to which the element belongs and may be used in the printout if desired. Its value must not be changed.

The following steps must be carried out within the subroutine:

- (1) Print an appropriate heading if NS = 1.
- (2) Print time history results using the data transferred in arrays ITHOUT and THOUT.
- (3) If ISE > 0, write the time history results (arrays ITHOUT and THOUT, or any other desired information) on file NF7. The variable NF7 is set to 7 in the current version of the program, so that this file is TAPE7. The file is rewound by the base program before the saving of time history results begins, but is not rewound at the end. Any desired data may be saved (for example, the arrays ITHOUT and THOUT), and the data may be formatted or unformatted. There will be one record for each output interval and each element. The data is not processed further by the program, being intended for processing by user-provided data processing programs.

6. EXAMPLE

The hypothetical example structure shown in Figs. 6.1 through 6.4 exercises most of the options in the program. The example illustrates data preparation procedures for the program, and may be used to check its execution following implementation on a new computer installation. Program decks received through the National Information Service for Earthquake Engineering will include a data deck and computer output for this example.

The input cards for the structure are listed in Table 6.1, including identification by the corresponding section in the User's Guide. A user who is unfamiliar with the program will be able to obtain guidance in data preparation procedures by studying this sample data.

The example structure does not represent any real structure, and the structural idealization used for the example is not necessarily a sound one for the analysis of real structures of similar type.

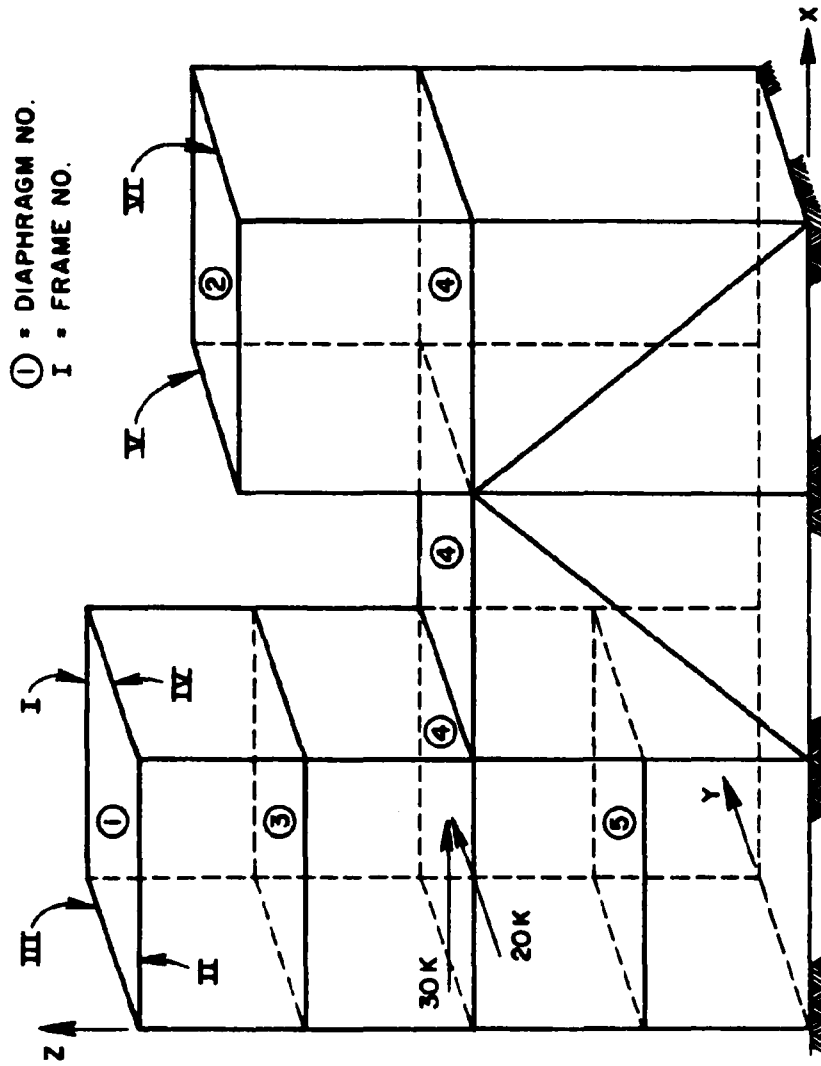


FIG. 6.1 EXAMPLE STRUCTURE

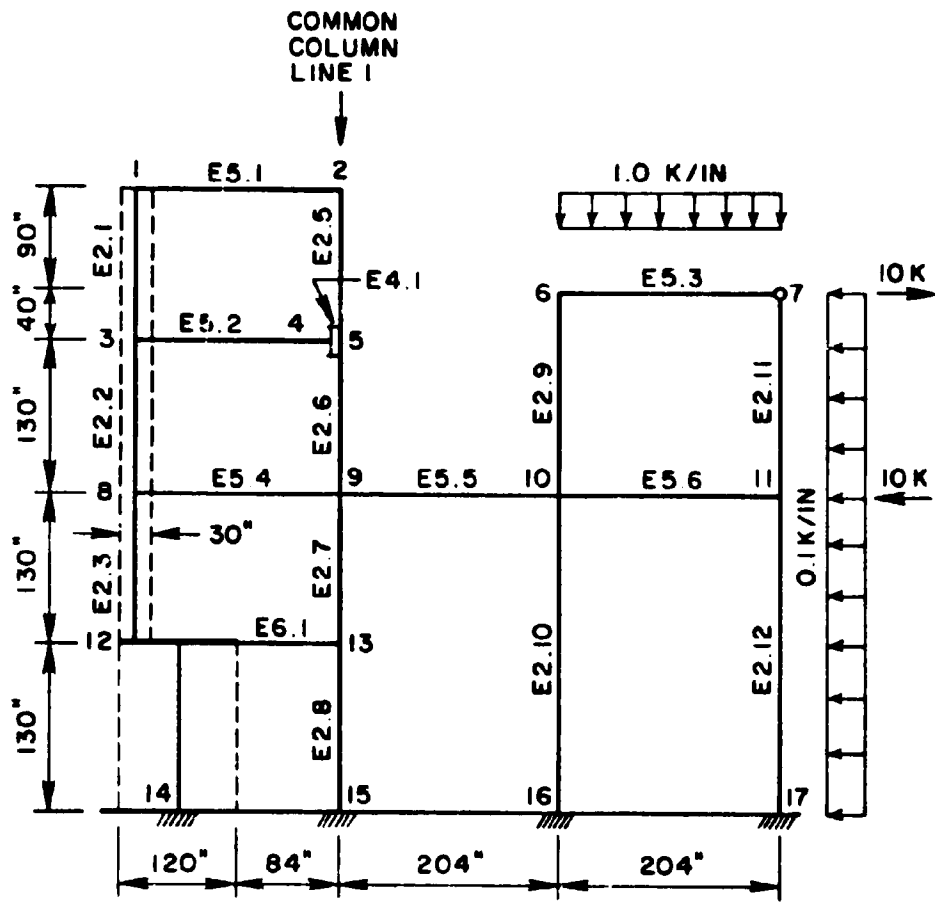


FIG. 6.2 FRAME I

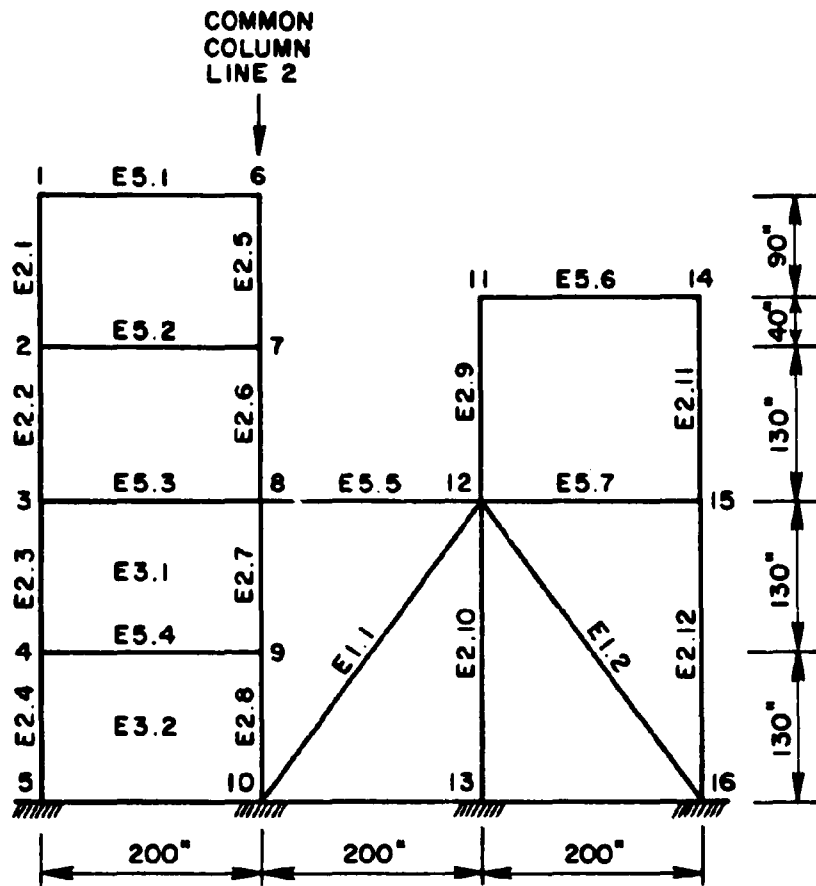


FIG. 6.3 FRAME II

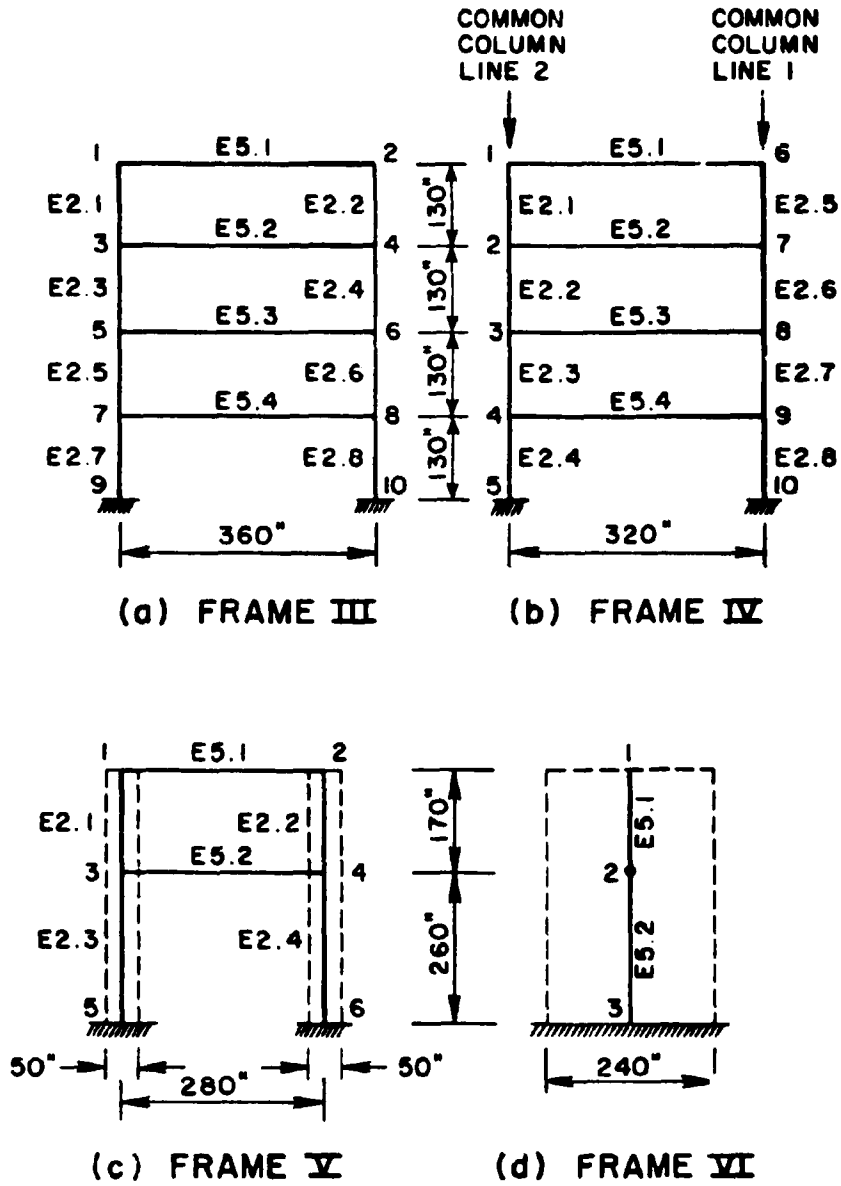


FIG. 6.4 FRAME III - VI

TABLE 6.1

LISTING OF INPUT DATA FOR
EXAMPLE STRUCTURE

The numbers on the right of the input data in the following table are the corresponding sections in the Program User's Guide. These numbers are not to be punched in the data cards.

7. CONCLUSION

The computer program described in this report is believed to be a useful practical tool for analysis of the inelastic earthquake response of three-dimensional buildings. The major restriction on the procedure is that frame interaction through common columns is not taken into account, so that the program is not applicable to tube-type buildings.

Because of the TABS type idealization and the use of substructuring techniques, the analysis will be substantially more efficient than an analysis based on a full three-dimensional idealization. The program is also able to consider both large and small structures efficiently, because of the several available data storage options. The program has some system-dependent features, but these are minor and adaptation of the program to other computer systems should not be difficult.

The program represents a substantial advance over the DRAIN-2D program, and bridges the gap between two-dimensional and full three-dimensional idealizations of building structures.

APPENDIX A
PROGRAM USER'S GUIDE

UNIVERSITY OF CALIFORNIA
BERKELEY

DIVISION OF STRUCTURAL ENGINEERING
AND STRUCTURAL MECHANICS

Computer Programming Series

IDENTIFICATION

DRAIN-TABS: computer program for the inelastic dynamic analysis of three-dimensional buildings under earthquake excitation.

Programmed: R. Guendelman, University of California, Berkeley, 1976.

PURPOSE

The program determines the inelastic dynamic response of three-dimensional buildings of essentially arbitrary configuration due to ground motions. Two independent horizontal motions plus vertical motion may be specified. Out-of-phase support motions cannot be considered. Static loads may be applied to the structure prior to the application of the dynamic loading, but the behavior under static load must be elastic.

The structure may be composed of elements having a variety of behavior patterns and yielding characteristics. The program is designed to permit new elements to be added with a relatively small amount of coding effort.

IDEALIZATION

The building is idealized as a series of plane frames interconnected by horizontal rigid diaphragms, with no enforcement

of compatibility for vertical and rotational displacements at joints common to two or more frames. Each frame can be of arbitrary geometry and include structural elements of a variety of types. It is not necessary for all frames to connect to all diaphragms, so that structures with independent diaphragms at some levels can be idealized.

Provision is made in the program for the axial forces in common columns to be computed as the sum of the forces for corresponding elements in intersecting frames. This option is currently available only for beam-column elements, but could be incorporated into new elements developed for the program if desired.

For each frame the elements must be divided into "groups." All elements in any group must be of the same type (e.g. truss, beam-column, etc.) and typically all elements of a single type will be included in a single group. However, elements of the same type may be subdivided into separate groups if desired. Within any group, elements must be numbered sequentially starting with 1.

The behavior of any element is determined by the programmer developing the subroutines for the element. This behavior may be of virtually any degree of complexity, depending on the purpose for which the element is intended. Characteristics of the available element library at the time of writing are described in Appendices B of Reference [2].

Displacements and coordinates are defined with respect to a right-handed global coordinate system (X, Y, Z) for the diaphragms, with Z vertically upwards and a local system (H, Z) for each frame, with H positive to the right.

CAPACITY LIMITATIONS

Some capacity limitations resulting from the use of fixed dimensions are associated with the input of element data, as explained in Section E. Some restrictions also result from the length of labelled COMMON/WORK/ as explained in this section. However, most data is stored in arrays with variable dimensions, with the required core storage established in blank COMMON and allocated dynamically at execution time.

The minimum required length of blank COMMON depends on the storage option selected, number of diaphragms, number of frames, number of nodes in each frame, the form of the building and frame stiffness matrices, number of integration time steps, and number of diaphragm and frame nodes for time-history printouts of displacements. Because there are several storage options and the stiffness matrices are stored in compacted form, the storage requirements are difficult to determine by hand. Hence, storage computations are made by the program and printed during data checking runs. It is assumed that one or more data checking runs will be required before any extensive computation is carried out. If a blank COMMON length in excess of that required is specified, the current version of the program for the CDC 6400 will reduce it to the required length.

For data checking runs the storage is independent of the storage option, and only a small amount of information is held in core. The storage required for data checking runs can be estimated as follows:

Let:

NFRAME = Number of frames.
 NDIAP = Number of diaphragms.
 NCOLC = Number of common column lines.
 NESPL = Maximum number of stories per common column line.
 NFORT = Maximum number of forces to be transmitted between common column. For the current version of the program this is set to 2.
 NSTEPS = Number of integration times steps.
 NDX, NDY, NDR = Number of diaphragms for which time history prints of X, Y and rotational displacements, respectively, are required.
 NDXR, NDYR = Number of pairs of diaphragms for which relative time history prints of X and Y displacements, respectively, are required.
 NEARQ = Number of earthquake motion components in X, Y, Z coordinates (1, 2 or 3).

For each frame let:

NJTS = Number of joints in the frame.
 NHOUT, NVOUT, NROUT = Numbers of nodes for which time history prints of H, Z and rotational displacements, respectively, are required.
 NHR, NVR = Number of node pairs for which time history prints of relative H and Z displacements, respectively, are required.
 NELFR = total number of elements in the frame.

Then the required blank COMMON for a data checking run is:

$NCOLC * NESPL * NFORT + 53 * NDIAP + NEARQ * NSTEPS + 9 * NFRAME + NDX + NDY + NDR + 2 * (NDXR + NDYR) + 31 + \text{Maximum for any frame of:}$

$(184 + 20 * NJTS + NROUT + NVOUT + NHOUT + 2 * (NHR + NVR))$

The storage requirements for each storage option will be printed as follows:

- (1) Minimum blank COMMON length for the option, without element data.

- (2) Required blank COMMON length required to hold all element information in core. In addition to the minimum storage in (1), space must be provided in blank COMMON to store information on the elements. The number of storage locations required for each different type of element is noted in section E. The element information is compacted into the available core, and if sufficient blank COMMON length is provided, the information for all elements will be retained permanently in core. The input/output cost for the computer run will then be minimized. If insufficient length is provided, the element information will automatically be divided into blocks, with as many elements as possible in each block. The blocks will be saved on scratch storage, and a considerable amount of input output effort may be involved.

In addition a message will be printed, indicating a needed increase in COMMON block WORK if the length of this block, LWORK, is less than

$$n_w = (NDIAP *(NDIAP +1))/2 + NDIAP + 6$$

The allocated storage can be modified by changing the following two cards:

- (1) LWORK = n_w in the main program.
- (2) COMMON/WORK/W(n_w) in the SDRAIN program.

INPUT DATA

The following punched cards define the problem to be solved.
Consistent units must be used throughout.

Sections A and B specify data for the complete building.
Sections C, D and E specify data for individual frames, and must be
repeated for each frame (i.e. Sections C, D, E for first frame;
Sections C, D, E for second frame, etc).

A1. PROBLEM INITIATION AND TITLE (A5,3X,18A4) - ONE CARD.

- Cols. 1 - 5: Punch the word START.
6 - 8: Blank
9 - 80: Problem title, to be printed with output.

A2. STRUCTURE CONTROL INFORMATION (7I5,5X,110) - ONE CARD.

- Cols. 1 - 5: Total number of frames in structure (NFRAM).
6 - 10: Number of floor diaphragms (NDIAP).
15: Code for allowable diaphragm displacement, as follows.
(a) 1: Full 3D action (X, Y and rotational displacements).
(b) 2: X displacement only.
(c) 3: Y displacement only.
(d) 4: X and Y displacements but no rotation.
19 - 20: Data checking code. Punch 1 if only a data checking run is required. Leave blank or punch zero if the problem is to be executed. Punch -1 if the problem is to be executed provided the number of element information blocks does not exceed one. This last option prevents execution, with excessive input/output cost, in cases where in-core operation is intended but errors are made in specifying the required storage.
24 - 25: Data storage option (1 through 10). See Table 4.1 for available options. Option 1 requires most core storage but least execution time; option 10 requires least storage but most execution time. The required storage lengths for all options are printed during data checking runs.
26 - 30: Number of common column lines (NCOLC). See NOTE 9 for explanation.

Cols. 31 - 35: Maximum number of column elements in any common column line. Leave blank if NCOLC = 0. If zero or blank and NCOLC > 0, assumed equal to the number of diaphragms, NDIAP. See NOTE 9.

41 - 50: Blank COMMON length to be assumed. If zero or blank, the value compiled into the program will be assumed. See discussion of capacity limitations.

A3. FLOOR DIAPHRAGM INFORMATION (6E10.0) - NDIAP CARDS, ONE FOR EACH DIAPHRAGM.

Cols. 1 - 10: X coordinate of center of mass.

11 - 20: Y coordinate of center of mass.

21 - 30: Translational mass for X direction.

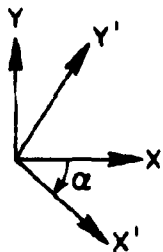
31 - 40: Translational mass for Y direction.

41 - 50: Rotational moment of inertia about Z axis.

51 - 60: Modifying factor by which masses are to be divided. If blank or zero, the factor from the preceding diaphragm is assumed, so that if the same factor applies for all diaphragms, it needs to be specified for the first diaphragm only. This factor will typically be equal to g, in which case the mass values will be given as weights. If zero or blank on the first card, assumed to be 1.0.

B1. LOAD CONTROL INFORMATION (2I5,E10.0,I5,5X,2E10.0) - ONE CARD.

- Cols. 5: Static load code. Punch 1 if static loads are to be applied before the dynamic loads. Leave blank or punch zero if static loads are not to be applied.
- 5 - 10: Number of static horizontal loads applied to diaphragms (NLDP). See Section B3.
- 11 - 20: Earthquake angle (degrees). Angle from global X axis to earthquake X' axis, measured clockwise from X to X' in plan (see figure).
- 21 - 25: Number of integration time steps to be considered in the dynamic analysis.
- 31 - 40: Integration time step, Δt .
- 41 - 50: Absolute value of the maximum displacement permitted before the structure can be assumed to have collapsed. The execution is terminated if this value is exceeded at any step. If zero or blank, assumed to be very high.



B2. EARTHQUAKE SCALE FACTORS (6E10.0) - ONE CARD.

This card may be left blank if all scale factors are 1.0.

- Cols. 1 - 10: Magnification factor to be applied to ground accelerations specified for the X' direction. See NOTE 5 for explanation.
- 11 - 20: Magnification factor to be applied to time scale of the acceleration record specified for the X' direction. See NOTE 5.
- 21 - 30: Magnification factor for ground accelerations in Y' direction.
- 31 - 40: Magnification factor for time scale in Y' direction.
- 41 - 50: Magnification factor for ground accelerations in Z (i.e. vertical) direction.
- 51 - 60: Magnification factor for time scale in Z direction.

B3. HORIZONTAL DIAPHRAGM LOADS (I5,5X,4E10.0) - NLDP CARDS, ONE FOR EACH DIAPHRAGM LOAD.

Omit if NLDP, Card B1, is zero. More than one load may be applied to any diaphragm if desired.

Cols. 1 - 5: Diaphragm number. Diaphragms are numbered in the sequence in which they were defined in Section A3.

6 - 10: Blank

11 - 20: X force, positive in the +X direction.

21 - 30: Y force, positive in the +Y direction.

31 - 40: X coordinate of point of application.

41 - 50: Y coordinate of point of application.

B4. TIME HISTORY OUTPUT SPECIFICATION.

Printed time histories of selected diaphragm displacements, frame nodal displacements and element results at selected time intervals may be obtained if desired. Envelope values of all nodal displacements and element results are automatically printed at the end of the computation and if the specified maximum displacement should be exceeded. Intermediate results envelopes may be printed at selected time intervals.

Time history values may be printed as the computation progresses, at the end of the computation only, or in both of these forms. The printouts at the end of the computation are ordered element by element, node by node, or diaphragm by diaphragm rather

than time step by time step, and hence are both more compact and more convenient for use in plotting time history graphs. These re-ordered time histories may also be saved on tape for subsequent machine plotting or other processing.

B4.1 CONTROL INFORMATION (1115) - ONE CARD.

- Cols. 1 - 5: Time interval for printout of diaphragm and node displacement time histories, expressed as a multiple of the time step Δt . Leave blank for no printout. The diaphragms for which time histories are required are specified in Sections B4.2 through B4.6. The nodes for which time histories are required are specified in Sections D2 through D6.
- 6 - 10: Time interval for printout of time histories of element results, expressed as a multiple of the time step Δt . Leave blank for no printout. The elements for which time histories are required are specified in Section E.
- 11 - 15: Time interval for intermediate printout of envelope values, expressed as a multiple of the time step Δt . Leave blank for no intermediate printout. Envelope values are automatically printed at the end of the response period.
- 16 - 20: Number of diaphragms (NDXOUT) for which X displacement time histories are required.
- 21 - 25: Number of diaphragms (NDYOUT) for which Y displacement time histories are required.
- 26 - 30: Number of diaphragms (NDROUT) for which rotation time histories are required.
- 31 - 35: Number of pairs of diaphragms (NDXR) for which time histories of relative X displacement are required.
- 36 - 40: Number of pairs of diaphragms (NDYR) for which time histories of relative Y displacement are required.

- Cols. 45: Time history print code for diaphragm displacement time histories, as follows:
- (a) Zero or blank: time history printout only as the computation progresses.
 - (b) 1: both a printout as the computation progresses and a re-ordered printout at the end of the computation.
 - (c) 2: only a re-ordered printout at the end of the computation.
- 50: Time history print code for relative diaphragm displacement time histories. Zero, 1 or 2, as for the preceding code.
- 51 - 55: Tape storage code for saving re-ordered time histories of diaphragm displacements, as follows:
- (a) Zero or blank: the re-ordered time histories are printed but not saved on tape.
 - (b) Nonzero: the re-ordered time histories are printed and also written on output unit TAPE7. See NOTE 8 for explanation.

B4.2 LIST OF DIAPHRAGMS FOR X DISPLACEMENT TIME HISTORIES (1015).

As many cards as needed to specify NDXOUT diaphragm numbers, punched ten to a card. Omit if NDXOUT is zero.

B4.3 LIST OF DIAPHRAGMS FOR Y DISPLACEMENT TIME HISTORIES (1015).

As many cards as needed to specify NDYOUT diaphragm numbers, punched ten to a card. Omit if NDYOUT is zero.

B4.4 LIST OF DIAPHRAGMS FOR ROTATION TIME HISTORIES (10I5).

As many cards as needed to specify NDROUT diaphragm numbers, punched ten to a card. Omit if NDROUT is zero.

B4.5 LIST OF DIAPHRAGMS FOR REALTIVE X DISPLACEMENT TIME HISTORIES (10I5).

As many cards needed to specify NXDR pairs of diaphragms, punched 5 pairs to a card. Omit if NXDR is zero. The printed displacement is the displacement of the first diaphragm of any pair minus the displacement of the second diaphragm.

B4.6 LIST OF DIAPHRAGMS FOR RELATIVE Y DISPLACEMENT TIME HISTORIES (10I5).

As many cards as needed to specify NYDR pairs of diaphragms, punched 5 pairs to a card. Omit if NDYR is zero.

B5. ACCELERATION RECORDS - ONE CONTROL CARD, FOLLOWED BY AS MANY CARDS AS NEEDED TO SPECIFY THE ACCELERATION TIME HISTORIES.

B5.1 CONTROL INFORMATION (5I5,5X,8A6) - ONE CARD.

Cols. 1 - 5: Number of time-acceleration pairs defining ground motion in X' direction (NPTX). Punch zero or leave blank for no ground motion in this direction.

6 - 10: Number of time-acceleration pairs defining ground motion in Y' direction (NPTY). Punch zero or leave blank for no ground motion in this direction.

11 - 15: Number of time-acceleration pairs defining ground motion in Z (vertical) direction (NPTZ). Punch zero or leave blank for no ground motion in this direction.

- Cols. 20: Code for printing accelerations as input. Leave blank or punch zero for no printout. Punch 1 to get listing of acceleration record.
- 25: Code for printing of accelerations as interpolated at intervals of Δt . Leave blank or punch zero for no printout. Punch 1 to get listing of acceleration record.
- 31 - 78: Optional title to identify records, to be printed with output.

B5.2 GROUND ACCELERATION TIME HISTORY IN X' DIRECTION (12F6.0).

As many cards as needed to specify NPTX time-acceleration pairs, 6 pairs to a card, assumed to be in acceleration units (not multiples of the acceleration due to gravity). Omit if NPTX equals zero. Note that both the acceleration and time scales may be scaled if desired (see Section B.2). If the record is input in terms of the acceleration due to gravity, the acceleration magnification factor must be g to convert to acceleration units.

B5.3 GROUND ACCELERATION TIME HISTORY IN Y' DIRECTION (12F6.0).

As many cards as needed to specify NPTY time-acceleration pairs. Omit if NPTY equals zero.

B5.4 GROUND ACCELERATION TIME HISTORY IN Z DIRECTION (12F6.0).

As many cards as needed to specify NPTZ time-acceleration pairs. Omit if NPTZ equals zero.

B6. DAMPING INFORMATION (4F10.0) - ONE CARD.

See NOTE 6 for explanation.

- Cols. 1 - 10: Mass proportional damping factor, α
- 11 - 20: Stiffness proportional damping factor, β , for current tangent stiffness.
- 21 - 30: Stiffness proportional damping factor, β_0 , for original elastic stiffness
- 31 - 40: "Structural" damping factor, δ . It is recommended that this factor be left blank.

C1. FRAME LOCATION AND TITLE (4E10.0,4A8) - ONE CARD.

See NOTE 10 for explanation of frame location and orientation.

Cols. 1 - 10: X coordinate of point 1 on frame.
 11 - 20: Y coordinate of point 1.
 21 - 30: X coordinate of point 2 on frame.
 31 - 40: Y coordinate of point 2.
 41 - 72: Optional frame title, to be printed with output.

C2. FRAME CONTROL INFORMATION (9I5) - ONE CARD.

Cols. 1 - 5: Number of nodes in frame.
 6 - 10: Number of "control nodes" for which coordinates are specified directly. See Section C3.
 11 - 15: Number of node coordinate generation commands. See Section C4.
 16 - 20: Number of commands specifying nodes with zero displacements. See Section C5.
 21 - 25: Number of commands specifying nodes connected to floor diaphragms. See Section C6.
 26 - 30: Number of commands specifying nodes with identical displacements. See Section C7.
 31 - 35: Number of commands specifying lumped masses at nodes. See Section C8.
 36 - 40: Number of commands specifying static loads applied directly at the nodes. See Section C9. Leave blank or punch zero if there are no static loads applied directly at the nodes.
 41 - 45: Number of different groups of elements in this frame. See Section E. Maximum 10.

- C3. CONTROL NODE COORDINATES (I5,2F10,0) - ONE CARD FOR EACH CONTROL NODE. See NOTE 1 for explanation.

Cols. 1 - 5: Node number. The nodes for each frame must be numbered starting with 1, with no numbers omitted. However, the control nodes may be input in any convenient sequence.

6 - 15: X coordinate of node.

16 - 25: Z coordinate of node.

- C4. COMMANDS FOR STRAIGHT LINE GENERATION OF NODE COORDINATES (4I5,F10.0) - ONE CARD FOR EACH GENERATION COMMAND.

Omit if there are no generation commands. See NOTE 1 for explanation.

Cols. 1 - 5: Node number at beginning of generation line.

6 - 10: Node number at end of generation line.

11 - 15: Number of nodes to be generated along line.

16 - 20: Node number difference (constant) between any two successive nodes on line. If blank or zero, assumed to be equal to 1.

21 - 30: Spacing between successive generated nodes. If greater than or equal to 1.0, assumed to be the actual spacing. If less than 1.0, assumed to be the actual spacing divided by the length of the generation line. If zero or blank, the nodes are automatically spaced uniformly along the generation line.

- C5. COMMANDS FOR NODES WITH ZERO DISPLACEMENTS (6I5) - ONE CARD FOR EACH COMMAND.

Omit if no nodes are constrained to have zero displacements.

See NOTE 2 for explanation.

- Cols. 1 - 5: Node number, or number of first node in a series of nodes covered by this command.
- 10: Code for H displacements. Punch 1 if constrained to be zero, otherwise leave blank or punch zero.
- 15: Code for Z displacement.
- 20: Code for rotation.
- 21 - 25: Number of last node in series. Leave blank for a single node.
- 26 - 30: Node number difference (constant) between successive nodes in series. If blank or zero, assumed to be equal to 1.

C6. COMMANDS FOR NODES CONNECTED TO FLOOR DIAPHRAGMS (1615) - ONE CARD FOR EACH COMMAND

See Chapter 2 and 3 for explanation.

- Cols. 1 - 5: Diaphragm number.
- 6 - 10: Number of nodes covered by this command (max. 14 - See NOTE 3 for procedure when more than 14 nodes are connected to one diaphragm).
- 11 - 80: Up to 14 fields, each 15. List of nodes, in increasing numerical order.

C7. COMMANDS FOR NODES WITH IDENTICAL DISPLACEMENTS (1615) - ONE CARD FOR EACH COMMAND.

Omit if no nodes are constrained to have identical displacements.

See NOTE 3 for explanation.

- Cols. 5: Displacement code, as follows.
- Punch 1 for H displacement.
- Punch 2 for Z displacement.

- Cols. 5: Punch 3 for rotation.
- 6 - 10: Number of nodes covered by this command
(max. 14 - See NOTE 3 for procedure when more
than 14 nodes have identical displacements).
- 11 - 80: Up to 14 fields, each I5. List of nodes, in
increasing numerical order.

C8. COMMANDS FOR LUMPED MASSES AT NODES (I5,3F10.0,2I5,F10.0) - ONE
CARD FOR EACH COMMAND

See NOTE 4 for explanation.

- Cols. 1 - 5: Node number, or number of first node in a
series of nodes covered by this command.
- 6 - 15: Mass associated with H displacement. See NOTE
11. Should be nonzero only in special cases.
- 16 - 25: Mass associated with Z displacement. May be
zero.
- 26 - 35: Rotary Inertia. May be zero.
- 36 - 40: Number of last node in series. Leave blank
for a single node.
- 41 - 45: Node number difference between successive
nodes in series. If blank or zero, assumed
to be equal to 1.
- 46 - 55: Modifying factor by which masses are to be
divided. If blank or zero, the factor from
the preceding command is assumed, so that if
the same factor applies for all commands, it
needs to be specified for the first command
only. This factor will typically be equal
to g, in which cases the mass values will
be given as weights. If zero or blank for
the first command, assumed to be 1.0.

C9. COMMANDS FOR STATIC LOADS APPLIED DIRECTLY AT NODES (I5,3F10.0, 2I5) - ONE CARD FOR EACH COMMAND.

Omit if there are no static loads applied directly at nodes.

If the static load code (Card B1) is zero or blank the loads will be read and printed, but are otherwise ignored.

- Cols. 1 - 5: Node number, or number of first node in a series of nodes covered by this command.
- 6 - 15: Load in H direction, the same on all nodes in the series.
- 16 - 25: Load in Z direction, the same on all nodes in the series.
- 26 - 35: Moment load (counterclockwise positive).
- 36 - 40: Number of last node in series. Leave blank for a single node.
- 41 - 45: Node number difference (constant) between successive nodes in series. If blank or zero, assumed to be equal to 1.

Note: A single node may appear in two or more commands if desired. In such a case, the total loads applied at the node will be the sum of the loads from the separate commands.

Note that Z loads are positive upwards.

D. FRAME TIME HISTORY OUTPUT SPECIFICATION

D1. CONTROL INFORMATION (1015) - ONE CARD.

- Cols. 1 - 5: Number of nodes (NHOUT) for which H displacement time histories are required.
- 6 - 10: Number of nodes (NZOUT) for which Z displacement time histories are required.
- 11 - 15: Number of nodes (NROUT) for which rotation time histories are required.
- 16 - 20: Number of pairs of nodes (NHR) for which time histories of relative H displacement are required.
- 21 - 25: Number of pairs of nodes (NZR) for which time histories of relative Z displacement are required.
- 30: Time history print code for nodal displacement time histories, as follows:
- (a) Zero or blank: time history printout only as the computation progresses.
 - (b) 1: both a printout as the computation progresses and a re-ordered printout at the end of the computation.
 - (c) 2: only a re-ordered printout at the end of the computation.
- 35: Time history print code for relative nodal displacement time histories. Zero, 1 or 2, as for the preceding code.
- 40: Time history print code for element results time histories. Zero, 1 or 2, as for the preceding code.
- 41 - 45: Tape storage code for saving re-ordered time histories of nodal displacements and relative displacements, as follows:
- (a) Zero or blank: The re-ordered time histories are printed but not saved on tape.

Cols. 41 - 45: (b) Nonzero: The re-ordered time histories are printed and also written on output unit TAPE7. See NOTE 8 for further explanation.

46 - 50: Tape storage code for saving re-ordered time histories of element results. Zero or nonzero, as for preceding code. See NOTE 8 for further explanation.

D2. LIST OF NODES FOR H DISPLACEMENT TIME HISTORIES (1015).

As many cards as needed to specify NHOUT node numbers, punched ten to a card. Omit if NHOUT equals zero.

D3. LIST OF NODES FOR Z DISPLACEMENT TIME HISTORIES (1015).

As many cards as needed to specify NZOUT node numbers, punched ten to a card. Omit if NZOUT equals zero.

D4. LIST OF NODES FOR ROTATION TIME HISTORIES (1015).

As many cards as needed to specify NROUT node numbers, punched ten to a card. Omit if NROUT equals zero.

D5. LIST OF NODES FOR RELATIVE H DISPLACEMENT TIME HISTORIES (1015).

As many cards as needed to specify NHR pairs of node numbers, 5 pairs to a card. Omit if NHR equals zero. The printed displacement is the displacement of the first node of any pair minus the displacement of the second node.

D6. LIST OF NODES FOR RELATIVE Z DISPLACEMENT TIME HISTORIES (1015).

As many cards as needed to specify NZR pairs of node numbers,
5 pairs to a card. Omit if NZR equals zero.

E. ELEMENT SPECIFICATION

For input and output, the elements of each frame must be divided into groups. All elements in any group must be of the same type, and typically all elements of a single type will be included in a single group. However, elements of the same type may be subdivided into separate groups if desired.

Element groups may be input in any convenient sequence. Within any group, the elements must be numbered in sequence beginning with 1.

The number of element groups for any frame may not exceed 10.

E1. TRUSS ELEMENTS

See Reference [2] for description of element. Number of words of information per element = 35.

E1(a) CONTROL INFORMATION FOR GROUP (4I5) - ONE CARD.

Cols. 5: Punch 1 (to indicate that group consists of truss elements).
 6 - 10: Number of elements in group.
 11 - 15: Number of different element stiffness types (max. 10). See Section E1(b).
 16 - 20: Number of different fixed end force patterns (max. 40). See Section E1(c).

E1(b) STIFFNESS TYPES (15,5F10.0,15) - ONE CARD FOR EACH STIFFNESS TYPE.

Cols. 1 - 5: Stiffness type number, in sequence beginning with 1.
 6 - 15: Young's modulus of elasticity.
 16 - 25: Strain hardening modulus, as a proportion of Young's modulus.
 26 - 35: Average cross sectional area.
 36 - 45: Yield stress in tension.
 46 - 55: Yield stress or elastic buckling stress in compression.
 60: Buckling code. Punch 1 if element buckles elastically in compression. Punch zero or leave blank if element yields in compression, without buckling.

E1(c) FIXED END FORCE PATTERNS (2I5,4F10.0) - ONE CARD FOR EACH
FIXED END FORCE PATTERN.

Omit if there are no fixed end forces.

Cols. 1 - 5: Pattern number, in sequence beginning with 1.

10: Axis code, as follows.

Code = 0: Forces are in the element coordinate system, as in Fig. B1.5a, Reference [2].

Code = 1: Forces are in the global coordinate system, as in Fig. B1.5b, Reference [2].

11 - 20: Clamping force F_1 .

21 - 30: Clamping force V_1 .

31 - 40: Clamping force F_j .

41 - 50: Clamping force V_j .

E1(d) ELEMENT GENERATION COMMANDS (9I5,2F5.0,F10.0) - ONE CARD
FOR EACH GENERATION COMMAND.

Elements must be specified in increasing numerical order.

Cards for the first and last elements must be included. See

NOTE 7 for explanation of generation procedure.

Cols. 1 - 5: Element number, or number of first element in a sequentially numbered series of elements to be generated by this command.

6 - 10: Node number at element end i.

11 - 15: Node number at element end j.

16 - 20: Node number increment for element generation. If zero or blank, assumed to be equal to 1.

- Cols. 21 - 25: Stiffness type number.
- 30: Code for including geometric stiffness. Punch 1 if geometric stiffness is to be included. Leave blank or punch zero if geometric stiffness is to be ignored. See Chapter 3, Section 3.12 for warning on geometric stiffness.
- 35: Time history output code. If a time history of element results is not required for the elements covered by this command, punch zero or leave blank. If a time history printout, at the intervals specified on Card B4.1, is required, punch 1.
- 36 - 40: Fixed end force pattern number for static dead loads on element. Leave blank if there are no dead loads. See Note below.
- 41 - 45: Fixed end force pattern number for static live loads on element. Leave blank if there are no live loads.
- 46 - 50: Scale factor to be applied to fixed end forces due to static dead loads. Leave blank if there are no dead loads.
- 51 - 55: Scale factor to be applied to fixed end forces due to static live loads. Leave blank if there are no live loads.
- 56 - 65: Initial axial force on element, tension positive.

Note: If the static load code, Card B1, is zero but fixed end forces are still specified for some elements, an inconsistency results. In effect, any such fixed end forces will be treated as initial element forces.

E2. BEAM-COLUMN ELEMENTS

See Reference [2] for description of element. Number of words of information per element = 144.

E2(a) CONTROL INFORMATION FOR GROUP (7I5) - ONE CARD.

- | | | |
|-------|----------|---|
| Cols. | 5: | Punch 2 (to indicate that group consists of beam column elements). |
| | 6 - 10: | Number of elements in group. |
| | 11 - 15: | Number of different element stiffness types (max. 40). See Section E2(b). |
| | 16 - 20: | Number of different end eccentricity types (max. 15). See Section E2(c). |
| | 21 - 25: | Number of different yield interaction surfaces for cross sections (max. 40). See Section E2(d). |
| | 26 - 30: | Number of different fixed end force patterns (max. 35). See Section E2(e). |
| | 31 - 35: | Number of different initial element force patterns (max. 30). See Section E2(f). |

E2(b) STIFFNESS TYPES (I5,4F10.0,3F5.0,2F10.0) - ONE CARD FOR EACH STIFFNESS TYPE.

- | | | |
|-------|----------|--|
| Cols. | 1 - 5: | Stiffness type number, in sequence beginning with 1. |
| | 6 - 15: | Young's modulus of elasticity. |
| | 16 - 25: | Strain hardening modulus, as a <u>proportion</u> of Young's modulus. |
| | 26 - 35: | Average cross sectional area. |
| | 36 - 45: | Reference moment of inertia. |
| | 46 - 50: | Flexural stiffness factor k_{ii} . |

- Cols. 51 - 55: Flexural stiffness factor k_{jj} .
- 56 - 60: Flexural stiffness factor k_{ij} .
- 61 - 70: Effective shear area. Leave blank or punch zero if shear deformations are to be ignored, or if shear deformations have already been taken into account in computing the flexural stiffness factors.
- 71 - 80: Poisson's ratio (used for computing shear modulus, and required only if shear deformations are to be considered).

E2(c) END ECCENTRICITIES (15,4F10.0) - ONE CARD FOR EACH END ECCENTRICITY TYPE.

Omit if there are no end eccentricities. See Fig. B2.6, Reference [2] for explanation. All eccentricities are measured from the node to the element end.

- Cols. 1 - 5: End eccentricity type number, in sequence beginning with 1.
- 6 - 15: $H_i = H$ eccentricity at end i.
- 16 - 25: $H_j = H$ eccentricity at end j.
- 26 - 35: $Z_i = Z$ eccentricity at end i.
- 36 - 45: $Z_j = Z$ eccentricity at end j.

E2(d) CROSS SECTION YIELD INTERACTION SURFACES (215,4F10.0,4F5.0) - ONE CARD FOR EACH YIELD SURFACE.

See Fig. B2.3, Reference [2] for explanation.

- Cols. 1 - 5: Yield surface number, in sequence beginning with 1.
- 10: Yield surface shape code, as follows.

- Cols. 10: Punch 1: Beam type, without P-M interaction.
 Punch 2: Steel I-beam type.
 Punch 3: Reinforced concrete column type.
- 11 - 20: Positive (sagging) yield moment, M_{y+} .
- 21 - 30: Negative (hogging) yield moment, M_{y-} .
- 31 - 40: Compression yield force, P_{yc} . Leave blank if shape code = 1.
- 41 - 50: Tension yield force, P_{yt} . Leave blank if shape code = 1.
- 51 - 55: M coordinate of balance point A, as a proportion of M_{y+} . Leave blank if shape code = 1.
- 56 - 60: P coordinate of balance point A, as a proportion of P_{yc} . Leave blank if shape code = 1.
- 61 - 65: M coordinate of balance point B, as a proportion of M_{y-} . Leave blank if shape code = 1.
- 66 - 70: P coordinate of balance point B, as a proportion of P_{yc} . Leave blank if shape code = 1.

E2(e) FIXED END FORCE PATTERNS (2I5,7F10.0) - ONE CARD FOR EACH FIXED END FORCE PATTERN.

Omit if there are no fixed end forces.

Cols. 1 - 5: Pattern number, in sequence beginning with 1.

10: Axis code, as follows.

Code = 0: Forces are in the element coordinate system, as in Fig. B2.5a, Reference [2].

Code = 1: Forces are in the global coordinate system, as in Fig. B2.5b, Reference [2].

Cols. 11 - 20: Clamping force, F_1 .
 21 - 30: Clamping force, V_1 .
 31 - 40: Clamping moment, M_1 .
 41 - 50: Clamping force, F_j .
 51 - 60: Clamping force, V_j .
 61 - 70: Clamping moment, M_j .
 71 - 80: Live load reduction factor, for computation
 of live load forces to be applied to nodes.
 See Section B2.5, Reference [2] for explanation.

E2(f) INITIAL ELEMENT FORCE PATTERNS (15,6F10.0) - ONE CARD FOR
 EACH INITIAL FORCE PATTERN

Omit if there are no initial forces. See Fig. B2.5a, Reference [2].

Cols. 1 - 5: Pattern number, in sequence beginning with 1.
 6 - 15: Initial axial force, F_1 .
 16 - 25: Initial shear force, V_1 .
 26 - 35: Initial moment, M_1 .
 36 - 45: Initial axial force, F_j .
 46 - 55: Initial shear force, V_j .
 56 - 65: Initial moment, M_j .

E2(g) ELEMENT GENERATION COMMANDS (8I5,2(I3,I2),2I5,2F5.0,I5,F5.0) -
 ONE CARD FOR EACH GENERATION COMMAND.

Elements must be specified in increasing numerical order.

Cards for the first and last elements must be included. See

NOTE 7 for explanation of generation procedure.

Col's. 1 - 5: Element number, or number of first element in a sequentially numbered series of elements to be generated by this command.

6 - 10: Node number at element end i.

11 - 15: Node number at element end j.

16 - 20: Node number increment for element generation. If zero or blank, assumed to be equal to 1.

21 - 25: Stiffness type number.

26 - 30: End eccentricity type number. Leave blank or punch zero if there is no end eccentricity.

31 - 35: Yield surface number for element end i.

36 - 40: Yield surface number for element end j.

42 - 43: Number of common column line for this element. Leave blank for no common column effect. See NOTE 9 and NOTE 7.

44 - 45: Number of column in column line. Leave blank for no common column effect.

48: Code for including geometric stiffness. Punch 1 if geometric stiffness is to be included. Leave blank or punch zero if geometric stiffness is to be ignored.

50: Time history output code. If a time history of element results is not required for the element covered by this command, punch zero or leave blank. If a time history printout, at the intervals specified on card B4.1, is required, punch 1.

51 - 55: Fixed end force pattern number for static dead loads on element. Leave blank or punch zero if there are no dead loads. See Note below.

56 - 60: Fixed end forces pattern number for static live loads on element. Leave blank or punch zero if there are no live loads.

61 - 65: Scale factor to be applied to fixed end forces due to static dead loads.

- Cols. 66 - 70: Scale factor to be applied to fixed end forces due to static live loads.
- 71 - 75: Initial force pattern number. Leave blank or punch zero if there are no initial forces.
- 76 - 80: Scale factor to be applied to initial element forces.

Note: If the static load code, Card B1, is zero but fixed end forces are still specified for some elements, an inconsistency results. In effect, any such fixed end forces will be treated as initial element forces.

E3. INFILL PANEL ELEMENTS

See Reference [2] for description of element. Number of words of information per element = 42.

E3(a) CONTROL INFORMATION FOR GROUP (3I5) - ONE CARD.

- Cols. 5: Punch 3 (to indicate that group consists of infill panel elements).
- 6 - 10: Number of elements in group.
- 11 - 15: Number of different element stiffness types (max. 40). See Section E3(b).

E3(b) STIFFNESS TYPES (15,5F10.0,15) - ONE CARD FOR EACH STIFFNESS TYPE.

- Cols. 1 - 5: Stiffness type number, in sequence beginning with 1.
- 6 - 15: Shear modulus of elasticity.
- 16 - 25: Strain hardening shear modulus, as a proportion of shear modulus of elasticity.
- 26 - 35: Average thickness of panel.
- 36 - 45: Yield stress in shear.
- 46 - 55: Strain at complete failure, as a proportion of strain at yield. This must not be less than 1.0.
- 60: Failure code, governing type of behavior after failure. Punch 1 if strength and stiffness are to be reduced to zero after failure. Punch zero or leave blank if strength and stiffness of elastic (strain hardening) component is to be retained after failure.

E3(c) ELEMENT GENERATION COMMANDS (815,F10.0) - ONE CARD FOR EACH GENERATION COMMAND

Elements must be specified in increasing numerical order.

Cards for the first and last elements must be included. See

NOTE 7 for explanation of generation procedure.

- Cols. 1 - 5: Element number, or number of first element in a sequentially numbered series of elements to be generated by this command.
- 6 - 10: Node number i. See Note below
- 11 - 15: Node number j.
- 16 - 20: Node number k.
- 21 - 25: Node number l.
- 26 - 30: Node number increment for element generation. If zero or blank assumed to be equal to 1.
- 31 - 35: Stiffness type number.
- 40: Time history output code. If a time history of element results is not required for elements covered by this command, punch zero or leave blank. If a time history printout, at the intervals specified on Card B4.1, is required, punch 1.
- 41 - 50: Initial shear stress in element.

Note: Node i may be any of the four nodes to which the element connects. Nodes j, k and l must then be counterclockwise from i around the element.

E4. SEMI-RIGID CONNECTION ELEMENTS

See Reference [2] for description of element. Number of words of information per element = 25.

E4(a) CONTROL INFORMATION FOR GROUP (3I5) - ONE CARD.

Cols. 5: Punch 4 (to indicate that group consists of semi-rigid connection elements).

 6 - 10: Number of elements in group.

 11 - 15: Number of different element stiffness types (max. 40). See Section E4(b).

E4(b) STIFFNESS TYPES (I5,4F10.0) - ONE CARD FOR EACH STIFFNESS TYPE.

Cols. 1 - 5: Stiffness type number, in sequence beginning with 1.

 6 - 15: Initial rotational stiffness (moment per radian).

 16 - 25: Strain hardening stiffness, as a proportion of initial rotational stiffness.

 26 - 35: Positive yield moment. See Note below.

 36 - 45: Negative yield moment.

Note: Positive rotation is rotation of node i counterclockwise relative to node j. A positive moment in the connection tends to produce positive rotation.

E4(c) ELEMENT GENERATION COMMANDS (6I5,F10.0) - ONE CARD FOR EACH GENERATION COMMAND.

Elements must be specified in increasing numerical order.

Cards for the first and last elements must be included. See

NOTE 7 for explanation of generation procedure.

Cols. 1 - 5: Element number, or number of first element in a sequentially numbered series of elements to be generated by this command.

6 - 10: Node number i.

11 - 15: Node number j.

16 - 20: Node number increment for element generation. If zero or blank, assumed to be equal to 1.

21 - 25: Stiffness type number.

30: Time history output code. If a time history of element results is not required for elements covered by this command, punch zero or leave blank. If a time history printout, at the intervals specified on Card B4.1, is required, punch 1.

31 - 40: Initial moment in connection.

E5. BEAM ELEMENTS

The beam element is identical to the beam column element except that only a beam type yield code is permitted.

The input data is identical to that for the beam column element (Section E2), except as follows.

- (1) Punch 5 in column 5 of Card E2(a), to indicate that the groups consists of beam elements.
- (2) The term "yield moment values" is substituted for "yield interaction surfaces".
- (3) The yield surface data on Cards E2(d) is unchanged. However, any data in columns 5 - 10 and 31 - 80 are ignored, and a beam type yield surface is automatically assumed.
- (4) The column line and column numbers, Card E2(g) Cols. 42-43 and 44 - 45, are ignored.

Number of words of information per element = 97.

E6. BEAM ELEMENTS WITH DEGRADING STIFFNESS

See Reference [10] for description of element. Number of words of information per element = 165.

Except for Sections E6(a) and (b) below, the input data is identical to that for beam-column elements (Section E2). See Note on sign convention at end of this section.

E6(a) CONTROL INFORMATION FOR GROUP (715) - ONE CARD.

Cols.	5: Punch 6 (to indicate that group consists of beam elements with degrading stiffness).
	6 - 10: Number of elements in group.
	11 - 15: Number of different element stiffness types (max. = 40). See Section E6(b).
	16 - 20: Number of different end eccentricity types (max. = 15). See Section E2(c).
	21 - 25: Number of different yield moment values for cross sections (max. = 40). See Section E2(d).
	26 - 30: Number of different fixed end force patterns (max. = 34). See Section E2(e).
	31 - 35: Number of different initial element force patterns (max. = 30). See Section E2(f).

E6(b) STIFFNESS TYPES - TWO CARDS FOR EACH STIFFNESS TYPE.

CARD 1: BEAM PROPERTIES CARD (I5,3F10.0,3F5.0,2F10.0)

Cols.	1 - 5: Stiffness type number, in sequence beginning with 1.
-------	---

- Cols. 6 - 15: EI, reference flexural stiffness.
- 16 - 25: EA, effective axial stiffness.
- 26 - 35: GA', effective shear stiffness. If blank or zero, shear deformations are neglected.
- 36 - 40: Flexural stiffness factor k_{ii} .
- 41 - 45: Flexural stiffness factor k_{jj} .
- 46 - 50: Flexural stiffness factor k_{ij} .
- 51 - 60: Strain hardening ratio for inelastic flexure at node i. If a nonzero hinge stiffness is specified for node i on CARD 2, columns 6 - 15, then this strain hardening ratio will apply directly to the hinge moment-rotation relationship. Otherwise, this ratio will apply to the overall beam end moment-rotation relationship or cantilever P- δ relationship.
- 61 - 70: Strain hardening ratio for inelastic flexure at node j. As with node i, a zero or nonzero hinge stiffness for node j on CARD 2, columns 16 - 25 will control whether this ratio is directly applied to the hinge alone or to the beam as a whole.

CARD 2: HINGE PROPERTIES CARD (15,7F10.0)

Leave blank to obtain Takeda model.

- Cols. 1 - 5: Stiffness type number, in sequence beginning with 1 and corresponding to the stiffness type number on the preceding BEAM PROPERTIES CARD.
- 6 - 15: Hinge stiffness at node i. Leave blank or zero if hinge properties are to be determined by the program (these properties are marked with * in the output). If blank or zero, the following field for node j must also be blank or zero. If nonzero, the following field must also be nonzero.

- Cols. 16 - 25: Hinge stiffness at node j.
- 26 - 35: α_i , unloading stiffness parameter for end i. Leave blank or zero for unloading according to Takeda model.
- 36 - 45: α_j , unloading stiffness parameter for end j. Leave blank or zero for unloading according to Takeda model.
- 46 - 55: β_i , loading parameter for end i. Leave blank or zero for reloading according to Takeda model.
- 56 - 65: β_j , loading parameter for end j. Leave blank or zero for reloading according to Takeda model.
- 66 - 75: N, loading exponential parameter.

E6(c) through E6(g) These sections are identical to sections E2(c) through E2(g).

NOTE ON SIGN CONVENTION

The sign convention used for positive bending moments for the degrading stiffness beam element differs from that for the beam-column and beam elements, as follows.

- (1) Beam-column and beam elements:



- (2) Degrading stiffness element:



F. NEXT PROBLEM

The data for a new problem may follow immediately, starting with Card A.

G. TERMINATION CARD (A4) - ONE CARD TO TERMINATE THE COMPLETE DATA DECK.

Cols. 1 - 4: Punch the word STOP.

USER'S GUIDE NOTESNOTE 1. NODE COORDINATE SPECIFICATION

The "control node" coordinates must be defined with respect to the frame H, Z coordinate system, with H positive to the right and Z positive upwards. The coordinates of the remaining nodes may be generated using straight line generation commands (Section B3). The number of nodes generated by each command may be one or any larger number. The coordinates of the two nodes at the beginning and end of the generation line must have been previously defined, either by direct specification or by previous straight line generation.

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 lines joining the specified nodes immediately preceding and following the group. That is, any generation command with equal spacing and a node number difference of one is superfluous.

NOTE 2. NODES WITH ZERO DISPLACEMENTS

Each node of the structure may have up to three degrees of freedom, namely H displacement, Z displacement and rotation. These are all displacements relative to the ground.

Initially the program assumes that all three degrees of freedom are present at all nodes (code = 0), and initializes the data arrays accordingly. If this assumption is correct, the cards of Section C5 should be omitted. In some cases, however, either (a) certain nodes may be fixed relative to the ground in certain directions or (b) it may be reasonable to assume zero displacement. Any degree of freedom which is fixed is to be assigned a code = 1, and cards must be included in Section C5 to specify those nodes and degrees of freedom for which the codes are equal to 1.

If there is any doubt, it should be assumed that all nodes can displace with all three degrees of freedom (i.e. all codes = 0). If however, certain degrees of freedom can be eliminated, the computer time may be significantly reduced. See Chapter 3 for further discussion.

NOTE 3. NODES WITH IDENTICAL DISPLACEMENTS

It may often be reasonable to assume that certain nodes displace identically in certain directions. Identical displacements may be specified by the commands of Section C7. The input format for this Section limit to 14 the number of nodes covered by any single command. If more than 14 nodes are to be assigned identical displacements, two or more commands should be used, with the nodes in increasing numerical order in each command, and with the smallest numbered node common to all commands. The procedure if more than 14 nodes are connected to a floor diaphragm is the same.

As with displacements which are constrained to be zero,

greater computational efficiency may be achieved by specifying identical displacements. However, whereas the specification of zero displacements will always decrease the structure stiffness band width or leave it unchanged, specification of identical displacements may increase this band width. The effect may be to increase the required structure stiffness storage and/or the computational effort required to solve the equilibrium equations. Identical displacements should therefore be specified with caution, and their effects on storage requirements and execution times should be investigated.

NOTE 4. SPECIFICATION OF LUMPED MASSES

The specification commands for lumped masses will generally permit the user to input the nodal masses with only a few cards. Any node may, if desired, appear in more than one specification command. In such cases the mass associated with any degree of freedom will be the sum of the masses specified in the separate commands. If certain nodes are constrained to have an identical displacement, the mass associated with this displacement will be the sum of the masses specified for the individual nodes.

NOTE 5. SCALING OF EARTHQUAKE RECORDS

The acceleration scale factors may be used to increase or decrease the ground acceleration, or to convert from multiples of the acceleration due to gravity to acceleration units. Modification of the earthquake intensity by scaling the acceleration values is a

common practice in research investigations, but should be undertaken cautiously in practical applications. When the accelerations are scaled, the ground velocities and displacements are scaled in the same proportion.

Provision is also made to modify the time scale. If a time scale factor equal to, say, f , is specified, all input times are multiplied by f before obtaining the interpolated accelerations at intervals equal to the integration time step. If the ground accelerations remain unchanged, the effect is to increase the ground velocities by f and the ground displacements by f^2 , and to alter the frequency content of the earthquake. Time scale modifications should not be made without carefully considering their influence on the ground motion.

NOTE 6. DAMPING

Damping of four different types may be specified, singly or in any combination. These are as follows:

(1) Mass-proportional damping, in which a viscous damping matrix $C = \alpha M$ is assumed, where M = mass matrix. The procedure for calculating α 's reviewed in Reference 1.

(2) Stiffness proportional damping, in which a viscous damping matrix $C = \beta K_T$ is assumed, where K_T = current tangent stiffness matrix at any time, including any geometric stiffnesses.

(3) Stiffness proportional damping, in which a viscous damping matrix $C = \beta_o K_o$ is assumed, where K_o = original elastic stiffness matrix, ignoring the geometric stiffness. See Reference

1 for further explanation.

(4) "Structural" damping, in which damping forces are assumed such that (a) the magnitudes of the damping forces originating within any element are a multiple, δ , of the absolute values of the element actions (axial forces, end moments, etc), and (b) the direction of each damping action is such that it opposes the rate of change of the corresponding element deformation. No viscous damping matrix is assumed. Instead, damping forces in any time step are applied, based on the member actions and rates of deformation existing at the end of the previous step. The member actions used to calculate the damping forces are the total actions, including both static and dynamic effects. The damping forces are first applied in the second time step. See Reference 1 for further explanation.

Little use appears to have been made of the "structural" damping concept, and hence little experience is available, especially for inelastic structures. A possible problem is that the damping forces may tend to accentuate small oscillations in the numerical computations, because the damping forces in any step are always based on the state at the end of the previous step. Until experience is gained, this option should be used cautiously. It is included in the program primarily for research reasons.

NOTE 7. ELEMENT DATA GENERATION

In the element generation commands, the elements must be specified in increasing numerical order. Cards may be provided for

sequentially numbered elements, in which case each card specifies one element and the generation option is not used. Alternatively, the cards for a group of elements may be omitted, in which case the data for the missing group is generated as follows.

1. All elements are assigned the same stiffnesses, strengths, element load data, output codes, etc., as the element preceding the missing group.

2. The numbers of the nodes for each missing element are obtained by adding the specified node number increment to the node numbers for the preceding element. The node number increment is that specified for the element preceding the missing group.

In the printout of the element data, generated data is identified by an asterisk at the beginning of the printed line. See NOTE 9 for common column generation.

NOTE 8. TAPE STORAGE OF TIME HISTORIES

If the tape storage code is nonzero, each printed line of time history data in the re-ordered time history (i.e. the print-out excluding any headings) is written, non-formatted, on I/O unit TAPE7, one record per printed line. That is, the printed time histories are directly reproduced on the file TAPE7.

This option should be used if routines are to be developed for computer plotting of the time history data. In such a case, TAPE7 will usually be equivalenced to a physical tape unit, so that the results will be saved on magnetic tape for future processing.

At the beginning of any computer run, TAPE7 is rewound. If more than one problem is executed in the run, the time histories will appear sequentially on TAPE7, in the same sequence as the printed time histories.

The program does not write any problem identification data on TAPE7, nor does it explicitly place an end-of-file mark after the last record.

To provide a record of the library number of the physical tape which has been used for any results set, it is suggested that the tape storage code be set equal to the library number. The printed value of the code will then provide a record of the tape assigned to the subsequent time history output.

NOTE 9. COMMON COLUMNS

Common columns are identified by the number of the column line and the number of the column within the line. Column lines may be numbered arbitrarily. The columns within a column line may be numbered from top to bottom or bottom to top, or even arbitrarily, depending on user preference and on whether or not beam-column elements are generated. If elements are generated, the column line number remains the same for all elements in the missing group, and the column number within the line is increased by one for each successive element.

Node numbers i and j for common columns must be the same for all elements making up a common column. That is, if node i is at the top of the column for the element, it must also be at the top of the column for the other element(s).

NOTE 10. FRAME LOCATION

All frames must be located with respect to a fixed, right-handed, X, Y, Z coordinate system, with Z vertically upwards. Any frame is positioned by specifying the X, Y coordinates of any two points on the H axis of the frame. The H coordinate of Point 2 must be larger than that of Point 1.

NOTE 11. H MASSES AT FRAME NODES

As noted in Chapter 3, Section 3.5, the program may not correctly account for horizontal masses specified at the frame level, and it is recommended that all horizontal inertia effects be lumped in the diaphragms. A user who wishes to specify H masses may do so, but should study Section 3.5 to ensure that the program will model the structure in the correct way.

REFERENCES

- [1] Kanaan, A. and Powell, G. H., "General Purpose Computer Program for Inelastic Dynamic Response of Plane Structures," Earthquake Engineering Research Center, Report No. EERC 73-6, University of California, Berkeley, 1973.
- [2] Powell, G. H., "DRAIN-2D User's Guide," Earthquake Engineering Research Center, Report No. EERC 73-22, University of California, Berkeley, 1973.
- [3] Wilson, E. L. and Dovey, H. H., "Three-Dimensional Analysis of Building Systems - TABS," Report No. EERC 72-8, Earthquake Engineering Research Center, University of California, Berkeley, December 1972.
- [4] Clough, R. W., King, I. P., and Wilson, E. L., "Structural Analysis of Multistory Buildings," Journal of the Structural Division, ASCE, August 1963.
- [5] Wilson, E. L., and Dovey, H. H., "Static and Earthquake Analysis of Three-Dimensional Frame and Shearwall Buildings," Report No. EERC 72-1, Earthquake Engineering Research Center, University of California, Berkeley, May 1972.
- [6] Weaver, W., and Nelson, M. F., "Three-Dimensional Analysis of Tier Buildings," Journal of the Structural Division, ASCE, Vol. 93, No. ST2, April 1967.
- [7] Wilson, E. L., Hollings, J. P., and Dovey, H. H., "Three-Dimensional Analysis of Building Systems (Extended Version)," Report No. EERC 75-13, Earthquake Engineering Research Center, University of California, Berkeley, April 1975.
- [8] Nair, R. S., "Linear Structural Analysis of Multistory Buildings," Journal of the Structural Division, ASCE, Vol. 101, No. ST3, March 1975.
- [9] Mondkar, D. P., and Powell, G. H., "Towards Optimal In-core Equation Solving," International Journal Computers and Structures," Vol. 4, pp. 531-548, 1974.
- [10] Litton, R. W., "A Contribution to the Analysis of Concrete Structures Under Cyclic Loading, Part I," Ph.D. Thesis, University of California, Berkeley, 1975. Included in DRAIN-2D documentation package distributed by NISEE.

EARTHQUAKE ENGINEERING RESEARCH CENTER REPORTS

- EERC 67-1 "Feasibility Study Large-Scale Earthquake Simulator Facility," by J. Penzien, J. G. Bouwkamp, R. W. Clough and D. Rea - 1967 (PB 187 905)
- EERC 68-1 Unassigned
- EERC 68-2 "Inelastic Behavior of Beam-to-Column Subassemblages Under Repeated Loading," by V. V. Bertero - 1968 (PB 184 888)
- EERC 68-3 "A Graphical Method for Solving the Wave Reflection-Refraction Problem," by H. D. McNiven and Y. Mengi 1968 (PB 187 943)
- EERC 68-4 "Dynamic Properties of McKinley School Buildings," by D. Rea, J. G. Bouwkamp and R. W. Clough - 1968 (PB 187 902)
- EERC 68-5 "Characteristics of Rock Motions During Earthquakes," by H. B. Seed, I. M. Idriss and F. W. Kiefer - 1968 (PB 188 338)
- EERC 69-1 "Earthquake Engineering Research at Berkeley," - 1969 (PB 187 906)
- EERC 69-2 "Nonlinear Seismic Response of Earth Structures," by M. Dibaj and J. Penzien - 1969 (PB 187 904)
- EERC 69-3 "Probabilistic Study of the Behavior of Structures During Earthquakes," by P. Ruiz and J. Penzien - 1969 (PB 137 886)
- EERC 69-4 "Numerical Solution of Boundary Value Problems in Structural Mechanics by Reduction to an Initial Value Formulation," by N. Distefano and J. Schujman - 1969 (PB 187 942)
- EERC 69-5 "Dynamic Programming and the Solution of the Biharmonic Equation," by N. Distefano - 1969 (PB 187 941)

Note: Numbers in parenthesis are Accession Numbers assigned by the National Technical Information Service. Copies of these 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 the reports (PB --- ---) and remittance must accompany each order. (Foreign orders, add \$2.50 extra for mailing charges.) Those reports without this information listed are not yet available from NTIS. Upon request, EERC will mail inquirers this information when it becomes available to us.

- EERC 69-6 "Stochastic Analysis of Offshore Tower Structures," by A. K. Malhotra and J. Penzien - 1969 (PB 187 903)
- EERC 69-7 "Rock Motion Accelerograms for High Magnitude Earthquakes," by H. B. Seed and I. M. Idriss - 1969 (PB 187 940)
- EERC 69-8 "Structural Dynamics Testing Facilities at the University of California, Berkeley," by R. M. Stephen, J. G. Bouwkamp, R. W. Clough and J. Penzien - 1969 (PB 189 111)
- EERC 69-9 "Seismic Response of Soil Deposits Underlain by Sloping Rock Boundaries," by H. Dezfulian and H. B. Seed - 1969 (PB 189 114)
- EERC 69-10 "Dynamic Stress Analysis of Axisymmetric Structures under Arbitrary Loading," by S. Chosh and E. L. Wilson - 1969 (PB 189 026)
- EERC 69-11 "Seismic Behavior of Multistory Frames Designed by Different Philosophies," by J. C. Anderson and V. V. Bertero - 1969 (PB 190 662)
- EERC 69-12 "Stiffness Degradation of Reinforcing Concrete Structures Subjected to Reversed Actions," by V. V. Bertero, B. Bresler and H. Ming Liao - 1969 (PB 202 942)
- EERC 69-13 "Response of Non-Uniform Soil Deposits to Travel Seismic Waves," by H. Dezfulian and H. B. Seed - 1969 (PB 191 023)
- EERC 69-14 "Damping Capacity of a Model Steel Structure," by D. Rea, R. W. Clough and J. G. Bouwkamp - 1969 (PB 190 663)
- EERC 69-15 "Influence of Local Soil Conditions on Building Damage Potential during Earthquakes," by H. B. Seed and I. M. Idriss - 1969 (PB 191 036)
- EERC 69-16 "The Behavior of Sands under Seismic Loading Conditions," by M. L. Silver and H. B. Seed - 1969 (AD 714 982)
- EERC 70-1 "Earthquake Response of Concrete Gravity Dams," by A. K. Chopra - 1970 (AD 709 640)
- EERC 70-2 "Relationships between Soil Conditions and Building Damage in the Caracas Earthquake of July 29, 1967," by H. B. Seed, I. M. Idriss and H. Dezfulian - 1970 (PB 195 762)

- EERC 70-3 "Cyclic Loading of Full Size Steel Connections," by E. P. Popov and R. M. Stephen - 1970 (PB 213 545)
- EERC 70-4 "Seismic Analysis of the Charaima Building, Caraballeda, Venezuela," by Subcommittee of the SEAONC Research Committee: V. V. Bertero, P. F. Fratessa, S. A. Mahin, J. H. Sexton, A. C. Scordelis, E. L. Wilson, L. A. Wyllie, H. B. Seed and J. Penzien, Chairman - 1970 (PB 201 455)
- EERC 70-5 "A Computer Program for Earthquake Analysis of Dams," by A. K. Chopra and P. Chakrabarti - 1970 (AD 723 994)
- EERC 70-6 "The Propagation of Love Waves across Non-Horizontally Layered Structures," by J. Lysmer and L. A. Drake - 1970 (PB 197 896)
- EERC 70-7 "Influence of Base Rock Characteristics on Ground Response," by J. Lysmer, H. B. Seed and P. B. Schnabel - 1970 (PB 197 897)
- EERC 70-8 "Applicability of Laboratory Test Procedures for Measuring Soil Liquefaction Characteristics under Cyclic Loading," by H. B. Seed and W. H. Peacock - 1970 (PB 198 016)
- EERC 70-9 "A Simplified Procedure for Evaluating Soil Liquefaction Potential," by H. B. Seed and I. M. Idriss - 1970 (PB 198 009)
- EERC 70-10 "Soil Moduli and Damping Factors for Dynamic Response Analysis," by H. B. Seed and I. M. Idriss - 1970 (PB 197 869)
- EERC 71-1 "Koyna Earthquake and the Performance of Koyna Dam," by A. K. Chopra and P. Chakrabarti - 1971 (AD 731 496)
- EERC 71-2 "Preliminary In-Situ Measurements of Anelastic Absorption in Soils Using a Prototype Earthquake Simulator," by R. D. Borcherdt and P. W. Rodgers - 1971 (PB 201 454)
- EERC 71-3 "Static and Dynamic Analysis of Inelastic Frame Structures," by F. L. Porter and G. H. Powell - 1971 (PB 210 135)
- EERC 71-4 "Research Needs in Limit Design of Reinforced Concrete Structures," by V. V. Bertero - 1971 (PB 202 943)
- EERC 71-5 "Dynamic Behavior of a High-Rise Diagonally Braced Steel Building," by D. Rea, A. A. Shah and J. G. Bouwkamp - 1971 (PB 203 584)

- EERC 71-6 "Dynamic Stress Analysis of Porous Elastic Solids Saturated with Compressible Fluids," by J. Ghaboussi and E. L. Wilson - 1971 (PB 211 396)
- EERC 71-7 "Inelastic Behavior of Steel Beam-to-Column Subassemblages," by H. Krawinkler, V. V. Bertero and E. P. Popov - 1971 (PB 211 335)
- EERC 71-8 "Modification of Seismograph Records for Effects of Local Soil Conditions," by P. Schnabel, H. B. Seed and J. Lysmer - 1971 (PB 214 450)
- EERC 72-1 "Static and Earthquake Analysis of Three Dimensional Frame and Shear Wall Buildings," by E. L. Wilson and H. H. Dovey - 1972 (PB 212 904)
- EERC 72-2 "Accelerations in Rock for Earthquakes in the Western United States," by P. B. Schnabel and H. B. Seed - 1972 (PB 213 100)
- EERC 72-3 "Elastic-Plastic Earthquake Response of Soil-Building Systems," by T. Minami - 1972 (PB 214 868)
- EERC 72-4 "Stochastic Inelastic Response of Offshore Towers to Strong Motion Earthquakes," by M. K. Kaul - 1972 (PB 215 713)
- EERC 72-5 "Cyclic Behavior of Three Reinforced Concrete Flexural Members with High Shear," by E. P. Popov, V. V. Bertero and H. Krawinkler - 1972 (PB 214 555)
- EERC 72-6 "Earthquake Response of Gravity Dams Including Reservoir Interaction Effects," by P. Chakrabarti and A. K. Chopra - 1972 (AD 762 330)
- EERC 72-7 "Dynamic Properties on Pine Flat Dam," by D. Rea, C. Y. Liaw and A. K. Chopra - 1972 (AD 763 928)
- EERC 72-8 "Three Dimensional Analysis of Building Systems," by E. L. Wilson and H. H. Dovey - 1972 (PB 222 438)
- EERC 72-9 "Rate of Loading Effects on Uncracked and Repaired Reinforced Concrete Members," by S. Mahin, V. V. Bertero, D. Rea and M. Atalay - 1972 (PB 224 520)
- EERC 72-10 "Computer Program for Static and Dynamic Analysis of Linear Structural Systems," by E. L. Wilson, K.-J. Bathe, J. E. Peterson and H. H. Dovey - 1972 (PB 220 437)

- EERC 72-11 "Literature Survey - Seismic Effects on Highway Bridges," by T. Iwasaki, J. Penzien and R. W. Clough - 1972 (PB 215 613)
- EERC 72-12 "SHAKE-A Computer Program for Earthquake Response Analysis of Horizontally Layered Sites," by P. B. Schnabel and J. Lysmer - 1972 (PB 220 207)
- EERC 73-1 "Optimal Seismic Design of Multistory Frames," by V. V. Bertero and H. Kamil - 1973
- EERC 73-2 "Analysis of the Slides in the San Fernando Dams during the Earthquake of February 9, 1971," by H. B. Seed, K. L. Lee, I. M. Idriss and F. Makdisi - 1973 (PB 223 402)
- EERC 73-3 "Computer Aided Ultimate Load Design of Unbraced Multistory Steel Frames," by M. B. El-Hafez and G. H. Powell - 1973
- EERC 73-4 "Experimental Investigation into the Seismic Behavior of Critical Regions of Reinforced Concrete Components as Influenced by Moment and Shear," by M. Celebi and J. Penzien - 1973 (PB 215 884)
- EERC 73-5 "Hysteretic Behavior of Epoxy-Repaired Reinforced Concrete Beams," by M. Celebi and J. Penzien - 1973
- EERC 73-6 "General Purpose Computer Program for Inelastic Dynamic Response of Plane Structures," by A. Kanaan and G. H. Powell - 1973 (PB 221 260)
- EERC 73-7 "A Computer Program for Earthquake Analysis of Gravity Dams Including Reservoir Interaction," by P. Chakrabarti and A. K. Chopra - 1973 (AD 766 271)
- EERC 73-8 "Behavior of Reinforced Concrete Deep Beam-Column Subassemblages under Cyclic Loads," by O. Kustu and J. G. Bouwkamp - 1973
- EERC 73-9 "Earthquake Analysis of Structure-Foundation Systems," by A. K. Vaish and A. K. Chopra - 1973 (AD 766 272)
- EERC 73-10 "Deconvolution of Seismic Response for Linear Systems," by R. B. Reimer - 1973 (PB 227 179)
- EERC 73-11 "SAP IV: A Structural Analysis Program for Static and Dynamic Response of Linear Systems," by K.-J. Bathe, E. L. Wilson and F. E. Peterson - 1973 (PB 221 967)
- EERC 73-12 "Analytical Investigations of the Seismic Response of Long, Multiple Span Highway Bridges," by W. S. Tseng and J. Penzien - 1973 (PB 227 816)

- EERC 73-13 "Earthquake Analysis of Multi-Story Buildings Including Foundation Interaction," by A. K. Chopra and J. A. Gutierrez - 1973 (PB 222 970)
- EERC 73-14 "ADAP: A Computer Program for Static and Dynamic Analysis of Arch Dams," by R. W. Clough, J. M. Raphael and S. Majtahedi - 1973 (PB 223 763)
- EERC 73-15 "Cyclic Plastic Analysis of Structural Steel Joints," by R. B. Pinkney and R. W. Clough - 1973 (PB 226 843)
- EERC 73-16 "QUAD-4: A Computer Program for Evaluating the Seismic Response of Soil Structures by Variable Damping Finite Element Procedures," by I. M. Idriss, J. Lysmer, R. Hwang and H. B. Seed - 1973 (PB 229 424)
- EERC 73-17 "Dynamic Behavior of a Multi-Story Pyramid Shaped Building," by R. M. Stephen and J. G. Bouwkamp - 1973
- EERC 73-18 "Effect of Different Types of Reinforcing on Seismic Behavior of Short Concrete Columns," by V. V. Bertero, J. Hollings, O. Kustu, R. M. Stephen and J. G. Bouwkamp - 1973
- EERC 73-19 "Olive View Medical Center Material Studies, Phase I," by B. Bresler and V. V. Bertero - 1973 (PB 235 986)
- EERC 73-20 "Linear and Nonlinear Seismic Analysis Computer Programs for Long Multiple-Span Highway Bridges," by W. S. Tseng and J. Penzien - 1973
- EERC 73-21 "Constitutive Models for Cyclic Plastic Deformation of Engineering Materials," by J. M. Kelly and P. P. Gillis - 1973 (PB 226 024)
- EERC 73-22 "DRAIN - 2D User's Guide," by G. H. Powell - 1973 (PB 227 016)
- EERC 73-23 "Earthquake Engineering at Berkeley - 1973" - 1973 (PB 226 033)
- EERC 73-24 Unassigned
- EERC 73-25 "Earthquake Response of Axisymmetric Tower Structures Surrounded by Water," by C. Y. Liaw and A. K. Chopra - 1973 (AD 773 052)
- EERC 73-26 "Investigation of the Failures of the Olive View Stairtowers during the San Fernando Earthquake and Their Implications in Seismic Design," by V. V. Bertero and R. G. Collins - 1973 (PB 235 106)

- EERC 73-27 "Further Studies on Seismic Behavior of Steel Beam-Column Subassemblages," by V. V. Bertero, H. Krawinkler and E. P. Popov - 1973 (PB 234 172)
- EERC 74-1 "Seismic Risk Analysis," by C. S. Oliveira - 1974 (PB 235 920)
- EERC 74-2 "Settlement and Liquefaction of Sands under Multi-Directional Shaking," by R. Pyke, C. K. Chan and H. B. Seed - 1974
- EERC 74-3 "Optimum Design of Earthquake Resistant Shear Buildings," by D. Ray, K. S. Pister and A. K. Chopra - 1974 (PB 231 172)
- EERC 74-4 "LUSH - A Computer Program for Complex Response Analysis of Soil-Structure Systems," by J. Lysmer, T. Udaka, H. B. Seed and R. Hwang - 1974 (PB 236 796)
- EERC 74-5 "Sensitivity Analysis for Hysteretic Dynamic Systems: Applications to Earthquake Engineering," by D. Ray - 1974 (PB 233 213)
- EERC 74-6 "Soil-Structure Interaction Analyses for Evaluating Seismic Response," by H. B. Seed, J. Lysmer and R. Hwang - 1974 (PB 236 519)
- EERC 74-7 Unassigned
- EERC 74-8 "Shaking Table Tests of a Steel Frame - A Progress Report," by R. W. Clough and D. Tang - 1974
- EERC 74-9 "Hysteretic Behavior of Reinforced Concrete Flexural Members with Special Web Reinforcement," by V. V. Bertero, E. P. Popov and T. Y. Wang - 1974 (PB. 236 797)
- EERC 74-10 "Applications of Reliability-Based, Global Cost Optimization to Design of Earthquake Resistant Structures," by E. Vitiello and K. S. Pister - 1974 (PB 237 231)
- EERC 74-11 "Liquefaction of Gravelly Soils under Cyclic Loading Conditions," by R. T. Wong, H. B. Seed and C. K. Chan - 1974
- EERC 74-12 "Site-Dependent Spectra for Earthquake-Resistant Design," by H. B. Seed, C. Ugas and J. Lysmer - 1974

- EERC 74-13 "Earthquake Simulator Study of a Reinforced Concrete Frame," by P. Hidalgo and R. W. Clough - 1974 (PB 241 944)
- EERC 74-14 "Nonlinear Earthquake Response of Concrete Gravity Dams," by N. Pal - 1974 (AD/A006583)
- EERC 74-15 "Modeling and Identification in Nonlinear Structural Dynamics, I - One Degree of Freedom Models," by N. Distefano and A. Rath - 1974 (PB 241 548)
- EERC 75-1 "Determination of Seismic Design Criteria for the Dumbarton Bridge Replacement Structure, Vol. 1: Description, Theory and Analytical Modeling of Bridge and Parameters," by F. Baron and S.-H. Pang - 1975
- EERC 75-2 "Determination of Seismic Design Criteria for the Dumbarton Bridge Replacement Structure, Vol. 2: Numerical Studies and Establishment of Seismic Design Criteria," by F. Baron and S.-H. Pang - 1975
- EERC 75-3 "Seismic Risk Analysis for a Site and a Metropolitan Area," by C. S. Oliveira - 1975
- EERC 75-4 "Analytical Investigations of Seismic Response of Short, Single or Multiple-Span Highway Bridges," by Ma-chi Chen and J. Penzien - 1975 (PB 241 454)
- EERC 75-5 "An Evaluation of Some Methods for Predicting Seismic Behavior of Reinforced Concrete Buildings," by Stephen A. Mahin and V. V. Bertero - 1975
- EERC 75-6 "Earthquake Simulator Study of a Steel Frame Structure, Vol. 1: Experimental Results," by R. W. Clough and David T. Tang - 1975 (PB 243 981)
- EERC 75-7 "Dynamic Properties of San Bernardino Intake Tower," by Dixon Rea, C.-Y. Liaw, and Anil K. Chopra - 1975 (AD/A008406)
- EERC 75-8 "Seismic Studies of the Articulation for the Dumbarton Bridge Replacement Structure, Vol. 1: Description, Theory and Analytical Modeling of Bridge Components," by F. Baron and R. E. Hamati - 1975
- EERC 75-9 "Seismic Studies of the Articulation for the Dumbarton Bridge Replacement Structure, Vol. 2: Numerical Studies of Steel and Concrete Girder Alternates," by F. Baron and R. E. Hamati - 1975

- EERC 75-10 "Static and Dynamic Analysis of Nonlinear Structures," by Digambar P. Mondkar and Graham H. Powell - 1975 (PB 242 434)
- EERC 75-11 "Hysteretic Behavior of Steel Columns," by E. P. Popov, V. V. Bertero and S. Chandramouli - 1975
- EERC 75-12 "Earthquake Engineering Research Center Library Printed Catalog" - 1975 (PB 243 711)
- EERC 75-13 "Three Dimensional Analysis of Building Systems," Extended Version, by E. L. Wilson, J. P. Hollings and H. H. Dovey - 1975 (PB 243 989)
- EERC 75-14 "Determination of Soil Liquefaction Characteristics by Large-Scale Laboratory Tests," by Pedro De Alba, Clarence K. Chan and H. Bolton Seed - 1975
- EERC 75-15 "A Literature Survey - Compressive, Tensile, Bond and Shear Strength of Masonry," by Ronald L. Mayes and Ray W. Clough - 1975
- EERC 75-16 "Hysteretic Behavior of Ductile Moment Resisting Reinforced Concrete Frame Components," by V. V. Bertero and E. P. Popov - 1975
- EERC 75-17 "Relationships Between Maximum Acceleration, Maximum Velocity, Distance from Source, Local Site Conditions for Moderately Strong Earthquakes," by H. Bolton Seed, Ramesh Murarka, John Lysmer and I. M. Idriss - 1975
- EERC 75-18 "The Effects of Method of Sample Preparation on the Cyclic Stress-Strain Behavior of Sands," by J. Paul Mulilis, Clarence K. Chan and H. Bolton Seed - 1975
- EERC 75-19 "The Seismic Behavior of Critical Regions of Reinforced Concrete Components as Influenced by Moment, Shear and Axial Force," by B. Atalay and J. Penzien - 1975
- EERC 75-20 "Dynamic Properties of an Eleven Story Masonry Building," by R. M. Stephen, J. P. Hollings, J. G. Bouwkamp and D. Jurukovski - 1975
- EERC 75-21 "State-of-the-Art in Seismic Shear Strength of Masonry - An Evaluation and Review," by Ronald L. Mayes and Ray W. Clough - 1975
- EERC 75-22 "Frequency Dependencies Stiffness Matrices for Viscoelastic Half-Plane Foundations," by Anil K. Chopra, P. Chakrabarti and Gautam Dasgupta - 1975
- EERC 75-23 "Hysteretic Behavior of Reinforced Concrete Framed Walls," by T. Y. Wong, V. V. Bertero and E. P. Popov - 1975

- EERC 75-24 "Testing Facility for Subassemblages of Frame-Wall Structural Systems," by V. V. Bertero, E. P. Popov and T. Endo - 1975
- EERC 75-25 "Influence of Seismic History of the Liquefaction Characteristics of Sands," by H. Bolton Seed, Kenji Mori and Clarence K. Chan - 1975
- EERC 75-26 "The Generation and Dissipation of Pore Water Pressures during Soil Liquefaction," by H. Bolton Seed, Phillippe P. Martin and John Lysmer - 1975
- EERC 75-27 "Identification of Research Needs for Improving a Seismic Design of Building Structures," by V. V. Bertero - 1975
- EERC 75-28 "Evaluation of Soil Liquefaction Potential during Earthquakes," by H. Bolton Seed, I. Arango and Clarence K. Chan 1975
- EERC 75-29 "Representation of Irregular Stress Time Histories by Equivalent Uniform Stress Series in Liquefaction Analyses," by H. Bolton Seed, I. M. Idriss, F. Makdisi and N. Banerjee 1975
- EERC 75-30 "FLUSH - A Computer Program for Approximate 3-D Analysis of Soil-Structure Interaction Problems," by J. Lysmer, T. Udaka, C.-F. Tsai and H. B. Seed - 1975
- EERC 75-31 "ALUSH - A Computer Program for Seismic Response Analysis of Axisymmetric Soil-Structure Systems," by E. Berger, J. Lysmer and H. B. Seed - 1975
- EERC 75-32 "TRIP and TRAVEL - Computer Programs for Soil-Structure Interaction Analysis with Horizontally Travelling Waves," by T. Udaka, J. Lysmer and H. B. Seed - 1975
- EERC 75-33 "Predicting the Performance of Structures in Regions of High Seismicity," by Joseph Penzien - 1975
- EERC 75-34 "Efficient Finite Element Analysis of Seismic Structure - Soil - Direction," by J. Lysmer, H. Bolton Seed, T. Udaka, R. N. Hwang and C.-F. Tsai - 1975
- EERC 75-35 "The Dynamic Behavior of a First Story Girder of a Three-Story Steel Frame Subjected to Earthquake Loading," by Ray W. Clough and Lap-Yan Li - 1975
- EERC 75-36 "Earthquake Simulator Study of a Steel Frame Structure, Volume II - Analytical Results," by David T. Tang - 1975
- EERC 75-37 "ANSR-I General Purpose Computer Program for Analysis of Non-Linear Structure Response," by Digambar P. Mondkar and Graham H. Powell - 1975

- EERC 75-38 "Nonlinear Response Spectra for Probabilistic Seismic Design and Damage Assessment of Reinforced Concrete Structures," by Masaya Murakami and Joseph Penzien - 1975
- EERC 75-39 "Study of a Method of Feasible Directions for Optimal Elastic Design of Framed Structures Subjected to Earthquake Loading," by N. D. Walker and K. S. Pister - 1975
- EERC 75-40 "An Alternative Representation of the Elastic-Viscoelastic Analogy," by Gautam Dasgupta and Jerome L. Sackman - 1975
- EERC 75-41 "Effect of Multi-Directional Shaking on Liquefaction of Sands," by H. Bolton Seed, Robert Pyke and Geoffrey R. Martin - 1975
- EERC 76-1 "Strength and Ductility Evaluation of Existing Low-Rise Reinforced Concrete Buildings - Screening Method," by Tsuneo Okada and Boris Bresler - 1976
- EERC 76-2 "Experimental and Analytical Studies on the Hysteretic Behavior of Reinforced Concrete Rectangular and T-Beams," by Shao-Yeh Marshall Ma, Egor P. Popov and Vitelmo V. Bertero - 1976
- EERC 76-3 "Dynamic Behavior of a Multistory Triangular-Shaped Building," by J. Petrovski, R. M. Stephen, E. Gartenbaum and J. G. Bouwkamp - 1976
- EERC 76-4 "Earthquake Induced Deformations of Earth Dams," by Norman Serff and H. Bolton Seed - 1976
- EERC 76-5 "Analysis and Design of Tube-Type Tall Building Structures," by H. de Clercq and G. H. Powell - 1976
- EERC 76-6 "Time and Frequency Domain Analysis of Three-Dimensional Ground Motions, San Fernando Earthquake," by Tetsuo Kubo and Joseph Penzien - 1976
- EERC 76-7 "Expected Performance of Uniform Building Code Design Masonry Structures," by R. L. Mayes, Y. Omote, S. W. Chen and R. W. Clough - 1976
- EERC 76-8 "Cyclic Shear Tests on Concrete Masonry Piers, Part I - Test Results," by R. L. Mayes, Y. Omote and R. W. Clough 1976
- EERC 76-9 "A Substructure Method for Earthquake Analysis of Structure - Soil Interaction," by Jorge Alberto Gutierrez and Anil K. Chopra - 1976
- EERC 76-10 "Stabilization of Potentially Liquefiable Sand Deposits using Gravel Drain Systems," by H. Bolton Seed and John R. Booker - 1976

- EERC 76-11 "Influence of Design and Analysis Assumptions on Computed Inelastic Response of Moderately Tall Frames," by G. I. Powell and D. G. Row - 1976
- EERC 76-12 "Sensitivity Analysis for Hysteretic Dynamic Systems: Theory and Applications," by D. Ray, K. S. Pister and E. Polak - 1976 (PB 262 859)
- EERC 76-13 "Coupled Lateral Torsional Response of Buildings to Ground Shaking," by Christopher L. Kan and Anil K. Chopra - 1976
- EERC 76-14 "Seismic Analysis of the Banco de America," by V. V. Bertero, S. A. Mahin, and J. A. Hollings - 1976
- EERC 76-15 "Reinforced Concrete Frame 2: Seismic Testing and Analytical Correlation," by Ray W. Clough and Jawahar Gidwani - 1976 (PB 261 323)
- EERC 76-16 "Cyclic Shear Tests on Masonry Piers, Part II - Analysis of Test Results," by R. L. Mayes, Y. Omote and R. W. Clough 1976
- EERC 76-17 "Structural Steel Bracing Systems: Behavior Under Cyclic Loading," by E. P. Popov, K. Takanashi and C. W. Roeder 1976 (PB 260 715)
- EERC 76-18 "Experimental Model Studies on Seismic Response of High Curved Overcrossings," by David Williams and William G. Godden - 1976
- EERC 76-19 "Effects of Non-Uniform Seismic Disturbances on the Dumbarton Bridge Replacement Structure," by Frank Baron and Raymond E. Hamati - 1976
- EERC 76-20 "Investigation of the Inelastic Characteristics of a Single Story Steel Structure using System Identification and Shaking Table Experiments," by Vernon C. Matzen and Hugh D. McNiven 1976 (PB 258 453)
- EERC 76-21 "Capacity of Columns with Splice Imperfections," by E. P. Popov, R. M. Stephen and R. Philbrick - 1976 (PB 260 378)
- EERC 76-22 "Response of the Olive View Hospital Main Building during the San Fernando Earthquake," by Stephen A. Mahin, Robert Collins, Anil K. Chopra and Vitelmo V. Bertero - 1976
- EERC 76-23 "A Study on the Major Factors Influencing the Strength of Masonry Prisms," by N. M. Mostaghel, R. L. Mayes, R. W. Clough and S. W. Chen - 1976
- EERC 76-24 "GADFLEA - A Computer Program for the Analysis of Pore Pressure Generation and Dissipation during Cyclic or Earthquake Loading," by J. R. Booker, M. S. Rahman and H. Bolton Seed - 1976

- EERC 76-25 "Rehabilitation of an Existing Building: A Case Study," by B. Bresler and J. Axley - 1976
- EERC 76-26 "Correlation Investigations on Theoretical and Experimental Dynamic Behavior of a Model Bridge Structure," by Kazuhiko Kawashima and Joseph Penzien - 1976 (PB 263 388)
- EERC 76-27 "Earthquake Response of Coupled Shear Wall Buildings," by Thirawat Srichatrapimuk - 1976
- EERC 76-28 "Tensile Capacity of Partial Penetration Welds," by Egor P. Popov and Roy M. Stephen - 1976 (PB 262 899)
- EERC 76-29 "Analysis and Design of Numerical Integration Methods in Structural Dynamics," by Hans M. Hilber - 1976
- EERC 76-30 "Contribution of a Floor System to the Dynamic Characteristics of Reinforced Concrete Buildings," by L. J. Edgar and V. V. Bertero - 1976
- EERC 76-31 "The Effects of Seismic Disturbances on the Golden Gate Bridge," by Frank Baron, Metin Arikian and Raymond E. Hamati 1976
- EERC 76-32 "Infilled Frames in Earthquake Resistant Construction," by R. E. Klinger and V. V. Bertero - 1976
- UCB/EERC-77/01 "PLUSH - A Computer Program for Probabilistic Finite Element Analysis of Seismic Soil-Structure Interaction," by Miguel P. Romo Organista, John Lysmer and H. Bolton Seed - 1977
- UCB/EERC-77/02 "Soil-Structure Interaction Effects at the Humbolt Bay Power Plant in the Ferndale Earthquake of June 7, 1975," by J. E. Valera, H. Bolton Seed, C. F. Tsai and J. Lysmer 1977
- UCB/EERC-77/03 "Influence of Sample Disturbance on Sand Response to Cyclic Loading," by Kenji Mori, H. Bolton Seed and Clarence K. Chan 1977
- UCB/EERC-77/04 "Seismological Studies of Strong Motion Records," by Jafar Shoja-Taheri - 1977
- UCB/EERC-77/05 "Testing Facility for Coupled-Shear Walls," by Lee Li-Hyung, Vitelmo V. Bertero and Egor P. Popov - 1977
- UCB/EERC-77/06 "Developing Methodologies for Evaluating the Earthquake Safety of Existing Buildings," No. 1 - B. Bresler; No. 2 - B. Bresler, T. Okada and D. Zisling; No. 3 - T. Okada and B. Bresler; No. 4 - V. Bertero and B. Bresler - 1977

- 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
- 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