AD-A107 635 COMPUTERS/STRUCTURES INTERNATIONAL OAKLAND CA F/G 13/13
THEORETICAL BASIS FOR CTABS80: A COMPUTER PROGRAM FOR THREE-DIM--ETC(U)
SEP 81 E L WILSON, H H DOVEY, A HABIBULLAH
UNCLASSIFIED WES-TR-K-81-2 NL

1 OF 2
AD A
107635

LEVEL

12

TECHNICAL REPORT K-81-2

THEORETICAL BASIS FOR CTABS80: A COMPUTER PROGRAM FOR THREE-DIMENSIONAL ANALYSIS OF BUILDING SYSTEMS

by

Edward L. Wilson, H. H. Dovey
Ashraf Habibullah

Computers/Structures International
4009 Webster Street
Oakland, Calif. 94609

September 1981

Final Report

A report under the Computer-Aided Structural Engineering (CASE) Project

Approved For Public Release, Distribution Unlimited

Prepared for Office, Chief of Engineers, U. S. Army
Washington, D. C. 20314

Monitored by Automatic Data Processing Center
U. S. Army Engineer Waterways Experiment Station
P. O. Box 631, Vicksburg, Miss. 39180

Destroy this report when no longer needed. Do not return it to the originator.

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products.

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: Technical Report K-81-2	2. GOVT ACCESSION NO.: AD-A110 725
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): THEORETICAL BASIS FOR CTABS80: A COMPUTER PROGRAM FOR THREE-DIMENSIONAL ANALYSIS OF BUILDING SYSTEMS	5. TYPE OF REPORT & PERIOD COVERED: Final report
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Edward L. Wilson, H. H. Dovey, Ashraf Habibullah	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Computers/Stuctures International, 4009 Webster Street, Oakland, Calif. 94609	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Office, Chief of Engineers, U. S. Army, Washington, D. C. 20314	12. REPORT DATE: September 1981
	13. NUMBER OF PAGES: 128
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office): U. S. Army Engineer Waterways Experiment Station, Automatic Data Processing Center, P. O. Box 631, Vicksburg, Miss. 39180	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Available from National Technical Information Service, Springfield, Va. 22151. This report was prepared under the Computer-Aided Structural Engineering (CASE) Project. A list of published CASE reports is printed on the inside of the back cover.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Buildings
Computer programs
CTABS80 (Computer program)
Structural analysis
Structural members

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report presents the theoretical basis for CTABS80, a computer program for the linear three-dimensional structural analysis of multistory frame and shear wall buildings subjected to static or dynamic loadings.

In CTABS80, the building is idealized as an assemblage of vertical independent frame and shear wall systems interconnected by horizontal floor diaphragms which are rigid in their own plane. The frame and shear wall systems must basically be of rectangular geometry (in elevation) with

(Continued)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. ABSTRACT (Continued)

vertical columns (or piers) and horizontal beams (or spandrels). However, with special modeling techniques, very complex situations may be considered. A special shear panel element is developed to enable modeling of discontinuous shear walls and shear walls with arbitrary openings. A diagonal bracing system to model braced frames (X-braced, K-braced, or eccentrically braced systems) is also presented.

The column, shear panel, and diagonal formulations include the effects of bending, axial, and shear deformations. Bending and shear deformations are also included in the beam formulation; however, the effects of axial deformations are neglected.

The effects of the finite dimensions of the beams and columns on the stiffness of a frame or shear wall system are automatically included.

The buildings may be unsymmetrical and nonrectangular in plan. Torsional behavior and interstory compatibility are accurately reflected in the results.

Four independent vertical and two independent lateral static load conditions are possible in any one run. These six static load conditions may be combined in any ratio to each other or to a lateral dynamic earthquake input which may be specified as a time-dependent ground acceleration or as an acceleration response spectrum.

Three-dimensional mode shapes and frequencies are evaluated.

The unique solution procedure used by CTABS80 considers the frame and shear walls as substructures, reduced with a modified wave front technique. This method results in a significant reduction in the program data preparation, computational effort, and storage requirements.

The consecutive levels of each of the individual frames can be arbitrarily connected to any (sequential by not necessarily consecutive) level of the structure, thereby making it possible for frames to bypass certain story levels. This option gives the program the capability to model partial diaphragms and multidiaphragms at any level.

A user's guide for the program is presented in Waterways Experiment Station Instruction Report K-81-9.

PREFACE

This report presents the theoretical basis for a computer program called CTABS80 that can be used for static and dynamic analysis of multi-story frame and shear wall buildings. Dr. E. H. Wilson, University of California, Berkeley, was responsible for developing the original version of the program (TABS), sponsored mainly by a National Science Foundation Grant.

Modifications to the program to make it a more useful tool for Corps of Engineers' personnel were made by Mr. Ashraf Habibullah, Computers/ Structures International, Oakland, Calif. His work was sponsored with funds provided to the Automatic Data Processing (ADP) Center, U. S. Army Engineer Waterways Experiment Station (WES), Vicksburg, Miss., by the Military Programs Directorate of the Office, Chief of Engineers, U. S. Army (OCE), under the Computer-Aided Structural Engineering (CASE) Project. This report and a companion user's guide for CTABS80 are the work of Dr. Wilson and Messrs. H. H. Dovey and Habibullah.

Specifications for the modifications to TABS were provided by the members of the CASE Task Group on Building Systems. The following were members of the Task Group (though all may not have served for the entire period) during the period of modifications to the program:

Mr. Dan Reynolds, Sacramento District (Chairman)
Mr. Jerry Foster, Baltimore District
Mr. Joseph Hartman, St. Louis District
Mr. David Illias, Portland District
Mr. Sefton Lucas, Memphis District
Mr. Jun Ouchi, Pacific Ocean Division
Mr. David Raisanen, North Pacific Division
Mr. Pete Rossbach, Baltimore District
Mr. James Simmons, Baltimore District
Mr. Ollie Werner, Middle East Division
Mr. Gene Wyatt, Mobile District

Dr. N. Radhakrishnan, Special Technical Assistant, ADP Center, WES, and CASE Project Manager, and Mr. Paul K. Senter, Computer-Aided Design Group (CADG), ADP Center, coordinated and monitored the work. Ms. Deborah K. Martin, CADG, supported the Task Group in changing the program to accept free-field input. Mr. Seymour Schneider, Military Programs Directorate, was the OCE point of contact. Mr. Donald L. Neumann was Chief, ADP Center.

Directors of WES during this period were COL J. L. Cannon, CE, COL N. P. Conover, CE, and COL T. C. Creel, CE. Technical Director was Mr. F. R. Brown.

CONTENTS

CONVERSION FACTORS, INCH-POUND TO METRIC (SI) UNITS OF MEASUREMENT

Inch-pound units of measurement used in this report can be converted to metric (SI) units as follows:

Multiply	By	To Obtain
feet	0.3048	metres
inches	2.54	centimetres
kips (1000 lb force)	4.448222	kilonewtons
kips (force) per foot	14.593904	kilonewtons per metre
pounds (force) per square foot	47.880263	pascals
pounds (mass) per cubic foot	16.01846	kilograms per cubic metre

THEORETICAL BASIS FOR CTABS80: A COMPUTER PROGRAM FOR THREE-DIMENSIONAL ANALYSIS OF BUILDING SYSTEMS

CHAPTER I: INTRODUCTION

A. Purpose

This report presents the theoretical basis for CTABS80, a computer program for the linear three-dimensional structural analysis of multistory frame and shear wall buildings subjected to static and dynamic loadings. A user's guide for the program is presented in Waterways Experiment Station (WES) Instruction Report K-81-9[18].

B. General-Purpose Programs for Structural Analysis

There are many two- and three-dimensional computer programs for the linear analysis of complex structures [1,2]. Most of these programs can be used for the static and dynamic analysis of multistory frame and shear wall buildings. However, most of these programs do not give special recognition to the fact that building systems in themselves are a very special class of structures from an analytical point of view. The following are some of the characteristics that are inherent in the nature of a building analysis that a general-purpose analysis program may not recognize, thereby resulting in significant losses in man-hours, computer time, and possibly accuracy:

1. Most buildings are of simple geometry with horizontal beams and vertical columns. A simple rectangular grid can define such a geometry

vertical columns. A simple rectangular grid can define such a geometry with minimal input. See Figure 1.

2. Many of the frames and shear walls are typical. Most general-purpose programs do not recognize this fact; therefore, the input may be large, and some internal calculations may be unnecessarily duplicated.

3. The in-plane stiffnesses of the floor systems of most buildings are very high. General-purpose programs do not necessarily recognize this, resulting in a set of equilibrium equations which may be very large, and thereby causing an increase in computation effort by a factor of 10 to 100. Also, numerical errors may be introduced since the in-plane floor stiffnesses are several orders of magnitude greater than the story-to-story stiffnesses of the structure. Since these two stiffnesses are added in a direct stiffness approach, double precision may be required in the solution.

4. The loading in building systems is of a restricted form. Loads, in general, are either vertically down (dead or live) or lateral (wind or seismic). The vertical loads are usually applied on the beams, and the lateral loads are generated at the floor levels.

5. In many buildings, the dimensions of the members are large and have a significant effect on the stiffness of the frame. Therefore, corrections need to be applied to the member stiffnesses. Most general-purpose programs work on center-line dimensions, and stiffness corrections are usually very tedious to implement.

6. In the dynamic analysis of buildings, the mass of the structure can be accurately lumped at the floor levels. Recognizing this fact

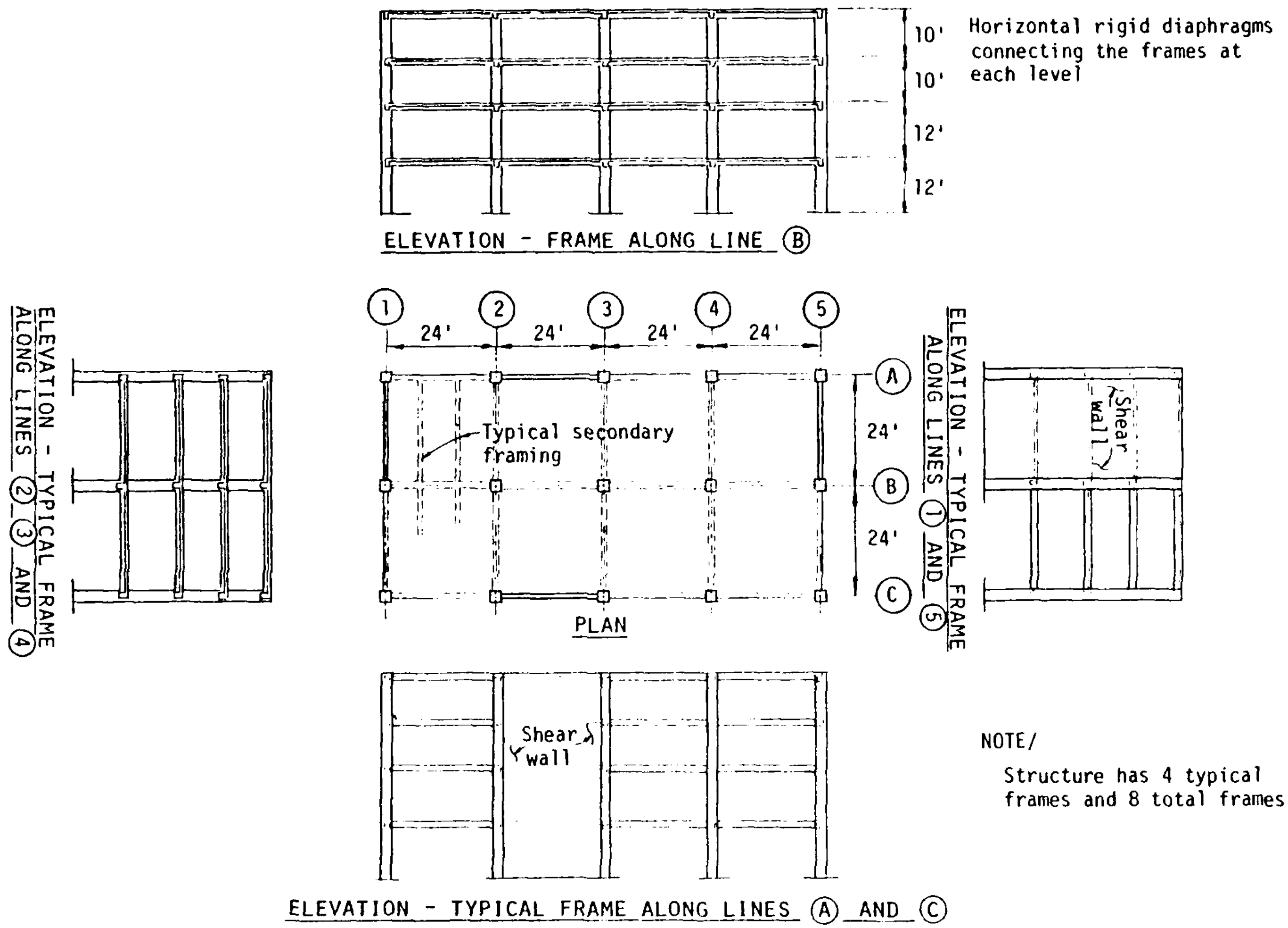


Figure 1. Typical frame and shear wall building

significantly reduces the size of the eigenvalue problem to be solved.

7. Various code loading requirements necessitate special options that allow convenient combinations of the vertical and lateral static and dynamic loadings. Also, the member forces need to be printed out at the support faces of the members. Such transformations are not automatic in general-purpose programs.

8. It is desirable to have a building analysis computer output printed in a special format; i.e., in terms of a particular frame, story, column, and beam. Also, special output such as story shears may be desirable.

In light of the above-mentioned and other reasons, the need for special-purpose programs for building analysis is apparent.

C. Special-Purpose Programs for Building Analysis

Various programs have been developed at the University of California at Berkeley for the linear analysis of multistory buildings in the past two decades [4,5,6]. These programs have been used in the profession on many major structures in many different countries. One of the major reasons for the development of computer program TABS [1,2,3] was the direct "feedback" from the profession in the use of these programs.

The first of these programs, FRMSTC, is a static load analysis program for symmetrical buildings with parallel frames and shear walls. Lateral mode shapes and frequencies are also evaluated.

Program FRMDYN is the same as FRMSTC except that the load input is ground accelerations due to a specified earthquake. Time-dependent displacements and member forces are produced but are not combined with static loads.

Program LATERAL is an extension of FRMSTC to the static analysis of a system of frames and shear walls which are not parallel. Three degrees of freedom exist at each story level. This program does not have dynamic options.

The first version of TABS was released in 1972, with the intent of replacing the computer programs described above. CTABS80 is an enhanced version of the original version of TABS and is intended to supercede other enhanced versions such as XTABS and TABS77.

The computer program ETABS [15] was released in 1975. The program allows three-dimensional frame input in which common column compatibility is enforced. The input data are more complex than those of TABS, and use of this program is only recommended if common column compatibility is important.

For buildings with other complexities, such as discontinuous or flexible diaphragms, sloped diaphragm, nonrectangular framing systems, etc., a general-purpose program such as SAPIV [12] or EASE2 [11] is still the most appropriate solution tool.

D. Disclaimer

Considerable time, effort, and expense have gone into the development and documentation of CTABS80, and the program has been thoroughly tested and used. In using the program, however, the user accepts and understands that no warrantly is expressed or implied, either by the sponsors, the developers, or the distributors, as to the accuracy or the reliability of the program. The user must clearly understand the basic assumptions of the program and must verify his own results.

CHAPTER II: STRUCTURAL IDEALIZATION

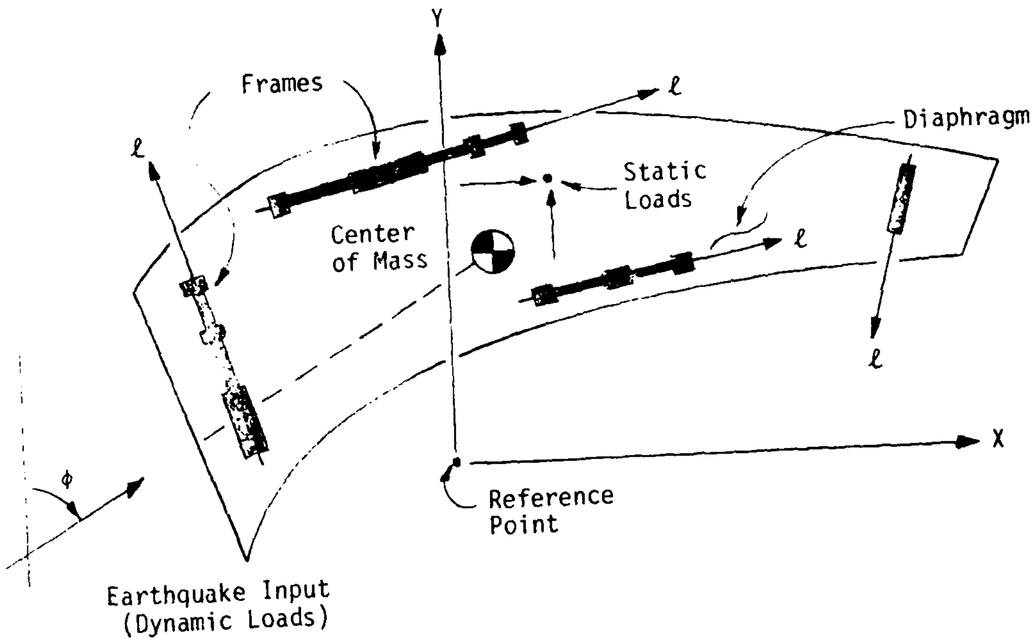
An exact three-dimensional structural analysis is required for only a limited number of buildings. For the majority of buildings the following approximations can be made. These approximations greatly simplify the preparation of input data and significantly reduce the computational efforts associated with the analysis of the structure.

1. The structure is idealized as an assemblage of vertical planar "frames". A frame consists of m columns and (m-1) beams. As long as shear and bending deformations are included in all members there is no need to distinguish between a shear wall/spandrel versus a beam/column system. See Figure 2.

2. The out-of-plane stiffnesses of all frames are assumed to be zero. Therefore each column at every floor has two degrees of freedom, a vertical displacement and a rotation. In addition, there is one lateral degree of freedom at every floor level of the frame.

3. Each floor is modeled as a horizontal diaphragm. This diaphragm is assumed to be infinitely stiff in-plane. The out-of-plane stiffness of this diaphragm is neglected. Bending stiffness of the floors may be included approximately in the modeling of the individual frames. It is apparent that axial deformation is not permitted in the beams. Floor levels must be the same for all frames. See Figure 2.

4. The floor diaphragm connects all the frames together at the corresponding level. The connection is only in a lateral sense. The frames otherwise are completely independent of each other. This also means that compatibility is not enforced with regard to



PLAN

Figure 2. Typical story level

displacements at columns which are common to more than one frame. Thus axial deformations in common columns will not be the same. As for joint rotations, if the frames with common members are perpendicular in plan view, then the rotations are uncoupled. This assumption invalidates the program for use in the analysis of structures in which the tubular effect or common column compatibility is important.

5. Vertical loads are applied to each frame on a tributary area basis. The diaphragm will not transfer any vertical load from one frame to another. However, no frame can sidesway independently without engaging the other frames.

6. Lateral loads are applied as loads for the complete floor at each level. The loads are applied at specified locations on the floor diaphragms and get distributed to the various frames in accordance with their corresponding stiffnesses and locations.

A. The Frame Substructure

The elevation of a typical frame is shown in Figure 3. The frame is basically of rectangular geometry with vertical column center lines and horizontal floor levels as the basic reference lines for the description of the frame.

The frame is an assemblage of column, beam, panel, and bracing elements. Vertical loading is applied to the individual frames by means of loading patterns associated with each beam.

The column and beam elements have options for rigid offsets at each end to compensate for the effects of the finite dimensions of the members on

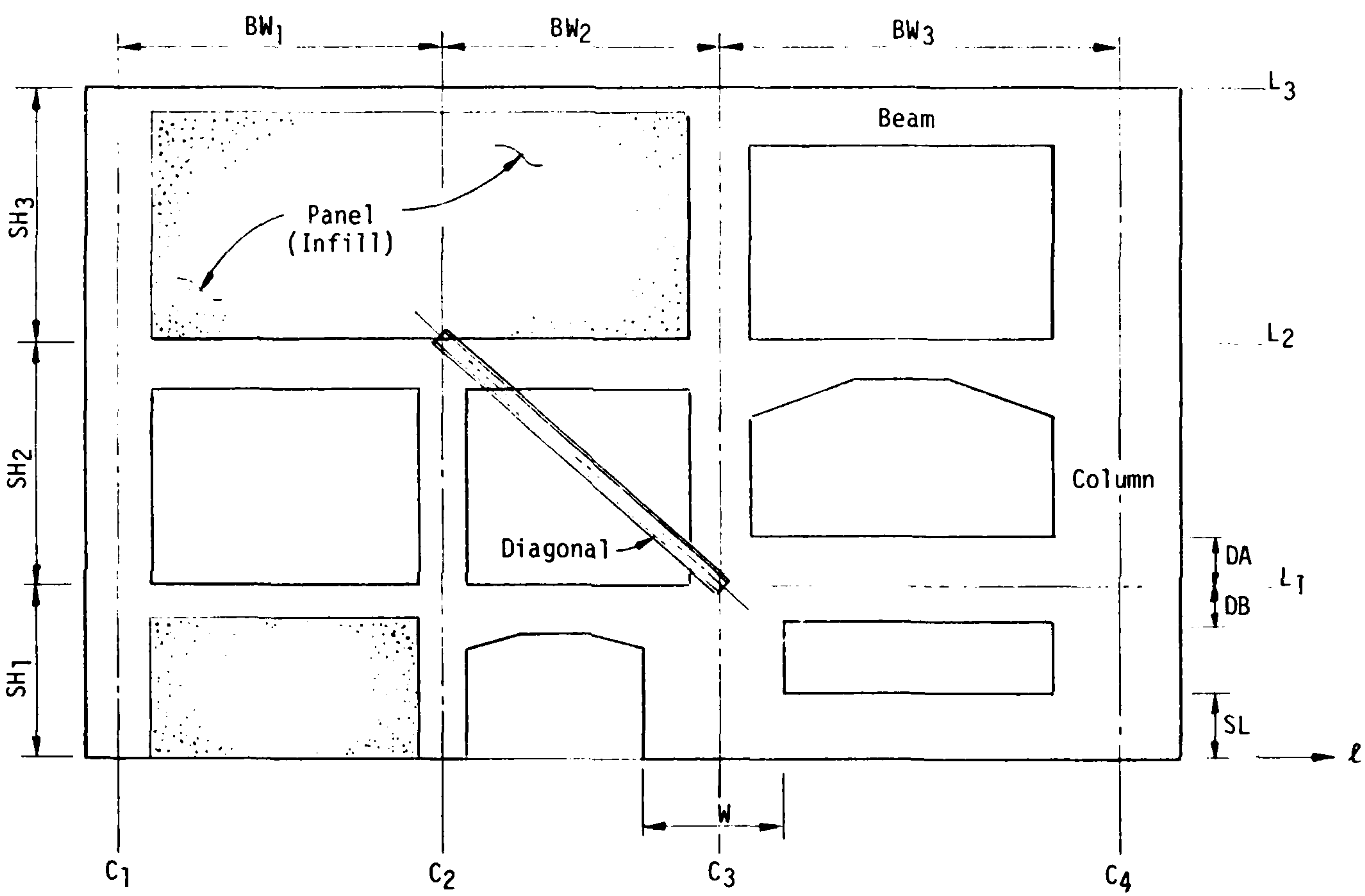


Figure 3. Elevation of typical frame

the stiffness of the system. The procedure used to set the lengths of these rigid offsets is presented later in this report.

(i). Individual Member Stiffnesses

The complete stiffness matrix of each frame is assembled by the direct stiffness technique. This involves calculating the local stiffness matrix, $\underset{\sim}{k}$, for each member along with a transformation matrix, $\underset{\sim}{a}$, which transforms the local displacements and forces, $\underset{\sim}{\phi}$, $\underset{\sim}{S}$, to global displacements and forces, $\underset{\sim}{r}$, $\underset{\sim}{R}$

$$\text{or:}\quad \underset{\sim}{\phi} = \underset{\sim}{a}\,\underset{\sim}{r} \qquad \text{also:}\quad \underset{\sim}{S} = \underset{\sim}{k}\,\underset{\sim}{\phi}$$

$$\underset{\sim}{S} = \underset{\sim}{a}\,\underset{\sim}{R} \qquad \text{and:}\quad \underset{\sim}{R} = \underset{\sim}{K}\,\underset{\sim}{r}$$

where $\underset{\sim}{K}$ is the stiffness matrix in global coordinates.

Substituting $\underset{\sim}{\phi} = \underset{\sim}{a}\,\underset{\sim}{r}$ and $\underset{\sim}{S} = \underset{\sim}{a}\,\underset{\sim}{R}$ into $\underset{\sim}{S} = \underset{\sim}{k}\,\underset{\sim}{\phi}$ we get:

$$\underset{\sim}{a}\,\underset{\sim}{R} = \underset{\sim}{k}\,\underset{\sim}{a}\,\underset{\sim}{r}$$

Premultiply both sides by $\underset{\sim}{a}^T$ and recognizing $\underset{\sim}{a}^T\,\underset{\sim}{a} = \underset{\sim}{I}$ we get:

$$\underset{\sim}{a}^T\,\underset{\sim}{a}\,\underset{\sim}{R} = \underset{\sim}{a}^T\,\underset{\sim}{k}\,\underset{\sim}{a}\,\underset{\sim}{r}$$

$$\underset{\sim}{R} = \underset{\sim}{a}^T\,\underset{\sim}{k}\,\underset{\sim}{a}\,\underset{\sim}{r}$$

As $\underset{\sim}{R} = \underset{\sim}{k}\,\underset{\sim}{r}$ we get $$\underset{\sim}{K} = \underset{\sim}{a}^T\,\underset{\sim}{k}\,\underset{\sim}{a}$$

Thus knowing the local stiffness matrix $\underset{\sim}{k}$ and the coordinate transformation matrix $\underset{\sim}{a}$ the global stiffness matrix may be evaluated.

The $\underset{\sim}{a}$ and $\underset{\sim}{k}$ matrices for the column, beam, panel, and brace elements are presented in Figures 4, 5, 6, and 7, respectively.

$$\begin{Bmatrix} M_i \\ M_j \\ S \end{Bmatrix} = \begin{bmatrix} S_a & S_b & 0 \\ S_b & S_a & 0 \\ 0 & 0 & S_c \end{bmatrix} \begin{Bmatrix} \phi_i \\ \phi_j \\ \delta \end{Bmatrix}$$

$$\underset{\sim}{S}_c = \underset{\sim}{k}_c \; \underset{\sim}{\Phi}_c$$

<u>FORCE/DEFORMATION TRANSFORMATION</u>

$$\begin{Bmatrix} \phi_i \\ \phi_j \\ \delta \end{Bmatrix} = \begin{bmatrix} 1 + \frac{b}{h} & \frac{1}{h} & \frac{a}{h} & -\frac{1}{h} & 0 & 0 \\ \frac{b}{h} & \frac{1}{h} & 1 + \frac{a}{h} & -\frac{1}{h} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{Bmatrix} \theta_B \\ u_B \\ \theta_T \\ u_T \\ v_B \\ v_T \end{Bmatrix}$$

$$\underset{\sim}{\Phi}_c = \underset{\sim}{a}_c \; \underset{\sim}{r}_c$$

<u>DEFORMATION/DISPLACEMENT TRANSFORMATION</u>

$$\underset{\sim}{K}_c = \underset{\sim}{a}_c^T \; \underset{\sim}{k}_c \; \underset{\sim}{a}_c$$

<u>COLUMN STIFFNESS MATRIX</u>
(6x6)

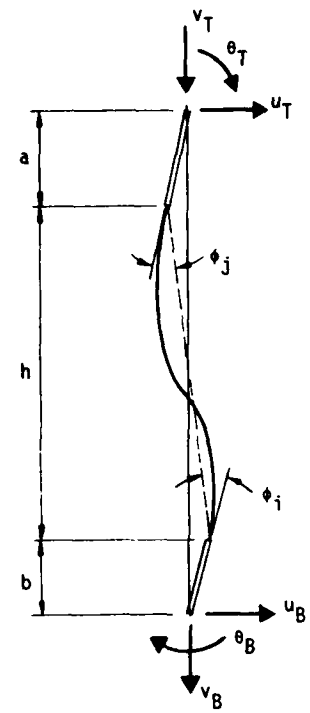


Figure 4

$$\begin{Bmatrix} M_i \\ M_j \end{Bmatrix} = \begin{bmatrix} S_a & S_b \\ S_b & S_a \end{bmatrix} \begin{Bmatrix} \phi_i \\ \phi_j \end{Bmatrix}$$

$$\underline{S}_b = \underline{k}_b \; \underline{\Phi}_b$$

<u>FORCE/DEFORMATION TRANSFORMATION</u>

$$\begin{Bmatrix} \phi_i \\ \phi_j \end{Bmatrix} = \begin{bmatrix} 1 + \frac{b}{L} & \frac{1}{L} & \frac{a}{L} & -\frac{1}{L} \\ \frac{b}{L} & \frac{1}{L} & 1 + \frac{a}{L} & -\frac{1}{L} \end{bmatrix} \begin{Bmatrix} \theta_L \\ v_L \\ \theta_R \\ v_R \end{Bmatrix}$$

$$\underline{\Phi}_b = \underline{a}_b \; \underline{r}_b$$

<u>DEFORMATION/DISPLACEMENT TRANSFORMATION</u>

$$\underline{K}_b = \underline{a}_b^T \; \underline{k}_b \; \underline{a}_b$$

<u>BEAM STIFFNESS MATRIX</u>

(4x4)

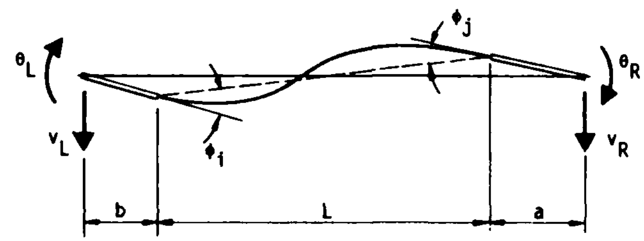


Figure 5

$$\begin{Bmatrix} M_i \\ M_j \\ S \end{Bmatrix} = \begin{bmatrix} S_a & S_b & 0 \\ S_b & S_a & 0 \\ 0 & 0 & S_c \end{bmatrix} \begin{Bmatrix} \phi_i \\ \phi_j \\ \delta \end{Bmatrix}$$

$$\underset{\sim}{S}_c = \underset{\sim}{k}_c \quad \underset{\sim}{\Phi}_c$$

<u>FORCE/DEFORMATION TRANSFORMATION</u>

$$\begin{Bmatrix} \phi_i \\ \phi_j \\ \delta \end{Bmatrix} = \begin{bmatrix} \frac{1}{h} & -\frac{1}{h} & -\frac{1}{L} & \frac{1}{L} & 0 & 0 \\ \frac{1}{h} & -\frac{1}{h} & 0 & 0 & -\frac{1}{L} & \frac{1}{L} \\ 0 & 0 & \frac{1}{2} & \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} \end{bmatrix} \begin{Bmatrix} u_B \\ u_T \\ v_{LB} \\ v_{RB} \\ v_{LT} \\ v_{RT} \end{Bmatrix}$$

$$\underset{\sim}{\Phi}_p = \underset{\sim}{a}_p \quad \underset{\sim}{r}_p$$

<u>DEFORMATION/DISPLACEMENT TRANSFORMATION</u>

$$\underset{\sim}{K}_p = \underset{\sim}{a}_p^T \quad \underset{\sim}{k}_p \quad \underset{\sim}{a}_p$$

<u>PANEL STIFFNESS MATRIX</u>
(6x6)

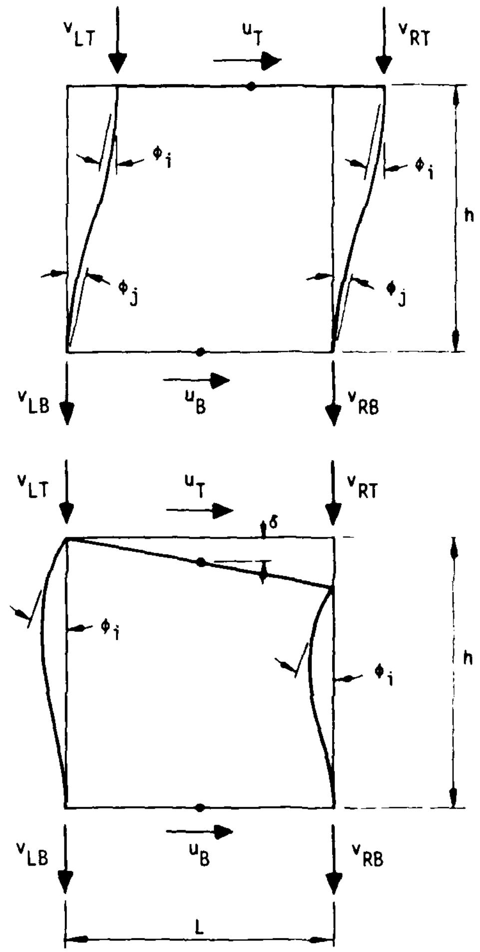


Figure 6

$$\begin{Bmatrix} M_i \\ M_j \\ S \end{Bmatrix} = \begin{bmatrix} S_a & S_b & 0 \\ S_b & S_a & 0 \\ 0 & 0 & S_c \end{bmatrix} \begin{Bmatrix} \phi_i \\ \phi_j \\ \delta \end{Bmatrix}$$

$$\underset{\sim}{S}_D = \underset{\sim}{k}_D \; \underset{\sim}{\Phi}_D$$

FORCE/DEFORMATION TRANSFORMATION

$$\begin{Bmatrix} \phi_i \\ \phi_j \\ \delta \end{Bmatrix} = \begin{bmatrix} 1 & \frac{h}{DD} & 0 & -\frac{h}{DD} & \frac{L}{DD} & -\frac{L}{DD} \\ 0 & \frac{h}{DD} & 1 & -\frac{h}{DD} & \frac{L}{DD} & -\frac{L}{DD} \\ 0 & -\frac{L}{D} & 0 & \frac{L}{D} & -\frac{h}{D} & \frac{h}{D} \end{bmatrix} \begin{Bmatrix} \theta_B \\ u_B \\ \theta_T \\ u_T \\ v_B \\ v_T \end{Bmatrix}$$

$$\underset{\sim}{\Phi}_D = \underset{\sim}{a}_D \; \underset{\sim}{r}_D$$

DEFORMATION/DISPLACEMENT MATRIX

$$\underset{\sim}{K}_D = \underset{\sim}{a}_D^T \; \underset{\sim}{k}_D \; \underset{\sim}{a}_D$$

DIAGONAL (BRACE) STIFFNESS MATRIX

(6x6)

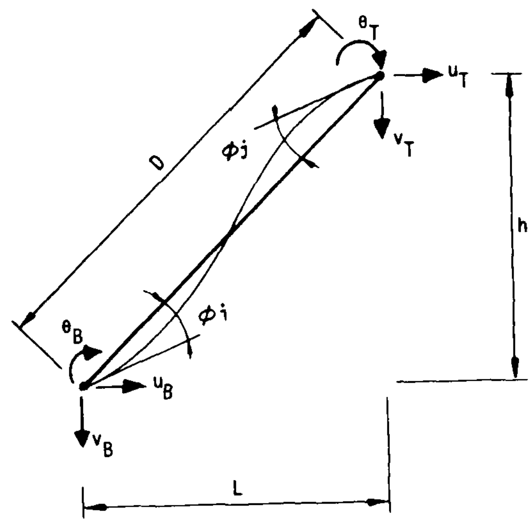


Figure 7

The column element formulation accounts for bending, axial and shear deformations. The basic stiffness matrix for such an element is shown in Figure 8. The column ends have options for rigid offsets. Figure 4 shows the six degrees of freedom associated with the column element and the deformation displacement transformation matrix linking the frame joint displacements to the column end deformations.

The beam element formulation is similar to that of the column except that the axial force component is dropped leaving a stiffness and transformation matrix as shown in Figure 5.

The panel element formulation is basically the same as that of the column except that each rotational degree of freedom is transformed into the two vertical displacements of the column lines bounding the panel element at each corresponding level. The axial degrees of freedom of the panel are also transformed as being an average of the vertical degrees of freedom of the two column lines bounding the panel element at each corresponding level. The degrees of freedom of the frame associated with the panel are therefore all translational. No rigid offsets are used. see Figure 6.

The diagonal element formulation is exactly the same as that of the column except that the brace is inclined and no rigid offsets are used. See Figure 7.

$$\begin{Bmatrix} M_i \\ M_j \\ S \end{Bmatrix} = \begin{bmatrix} S_a & S_b & 0 \\ S_b & S_a & 0 \\ 0 & 0 & S_c \end{bmatrix} \begin{Bmatrix} \phi_i \\ \phi_j \\ \delta \end{Bmatrix}$$

<u>FORCE/DEFORMATION TRANSFORMATION</u>

Where: $S_a = \frac{2EI}{L} \left(\frac{2 + \beta}{1 + 2\beta} \right)$

$S_b = \frac{2EI}{L} \left(\frac{1 - \beta}{1 + 2\beta} \right)$

$S_c = \frac{AE}{L}$

$\beta = \frac{6EI}{L^2 \bar{A} G}$

A = axial area

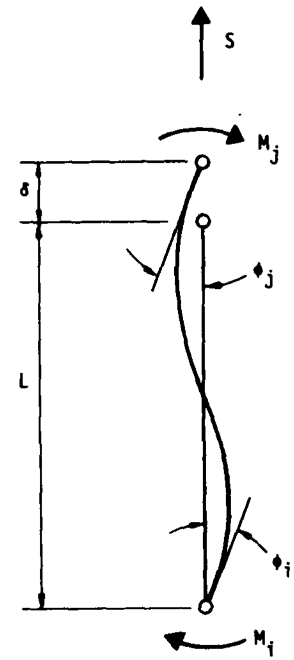
$\bar{A}$ = effective shear area

I = moment of inertia

E = elastic modulus

G = shear modulus

L = length



<u>FLEXURAL MEMBER WITH AXIAL DEFORMATIONS</u>

Figure 8

The complete stiffness matrix for each frame has two degrees of freedom for each beam-column intersection and one lateral degree of freedom per story.

(ii). Lateral Frame Stiffness

With the frame degrees of freedom appropriately ordered, the frame equilibrium equations have the form shown in Figure 9 . Where N is the number of stories in the frame, $\underset{\sim}{r}_n$ is the vector of joint displacements (that is vertical displacement and rotation) at story level n and $\underset{\sim}{r}_L$ is the vector of lateral story displacements. Lateral loads are applied to the complete structure and are considered when the lateral stiffness matrix for the complete building is assembled. Gaussian elimination is performed on the full system up to and including the equations:

$$\underset{\sim}{R}_N = \underset{\sim}{C}_{N-1}\,\underset{\sim}{r}_{N-1} + \underset{\sim}{K}_N\,\underset{\sim}{r}_N + \underset{\sim}{E}_N\,\underset{\sim}{r}_L$$

The last N equations ($\underset{\sim}{r}_L$ is a vector of order N) may now be written as:

$$\underset{\sim}{R}_L = \underset{\sim}{K}_L\,\underset{\sim}{r}_L$$

The vector $\underset{\sim}{R}_L$ is the lateral load submatrix of the frame and is modified by the elimination process due to vertical loading on the frame. These terms represent the sidesway effects under vertical loading. The matrix $\underset{\sim}{K}_L$ clearly represents the frame lateral stiffness matrix; i.e., the stiffness matrix of the frame in terms of only the lateral story displacements.

Within the computer program the following approach is adopted in order to reduce storage requirements. The assembly and reduction process is

$$\begin{bmatrix} \underline{R}_1 \\ \underline{R}_2 \\ \underline{R}_3 \\ \vdots \\ \underline{R}_n \\ \underline{R}_{n+1} \\ \vdots \\ \underline{R}_{N-1} \\ \underline{R}_N \\ \underline{R}_L \end{bmatrix} = \begin{bmatrix} \underline{K}_1 & \underline{C}_1 & & & & & & & & \underline{E}_1 \\ \underline{C}_1^T & \underline{K}_2 & \underline{C}_2 & & & & & & & \underline{E}_2 \\ & \underline{C}_2^T & \underline{K}_3 & \underline{C}_3 & & & & & & \underline{E}_3 \\ & & & \ddots & & & & & & \vdots \\ & & & & \underline{K}_n & \underline{C}_n & & & & \underline{E}_n \\ & & & & \underline{C}_n^T & \underline{K}_{n+1} & & & & \underline{E}_{n+1} \\ & & & & & & \ddots & & & \vdots \\ & & & & & & & \underline{K}_{N-1} & \underline{C}_{N-1} & \underline{E}_{N-1} \\ & & & & & & & \underline{C}_{N-1}^T & \underline{K}_N & \underline{E}_N \\ \underline{E}_1^T & \underline{E}_2^T & \cdots & & \underline{E}_n^T & \underline{E}_{n+1}^T & \cdots & \underline{E}_{N-1}^T & \underline{E}_N^T & \underline{K}_L \end{bmatrix} \begin{bmatrix} \underline{r}_1 \\ \underline{r}_2 \\ \underline{r}_3 \\ \vdots \\ \underline{r}_n \\ \underline{r}_{n+1} \\ \vdots \\ \underline{r}_{N-1} \\ \underline{r}_N \\ \underline{r}_L \end{bmatrix} \quad \begin{array}{c} \text{SIZE} \\ 2NC \\ 2NC \\ 2NC \\ \\ 2NC \\ 2NC \\ \\ 2NC \\ 2NC \\ NS+1 \end{array}$$

The equations in core at any one time are blocked out above.

NC = No. of Column Lines NS = No. of Stories

Figure 9. Complete equation system of frame substructure

carried out systematically story by story from the top of the structure such that at any level, n, we consider the system shown below:

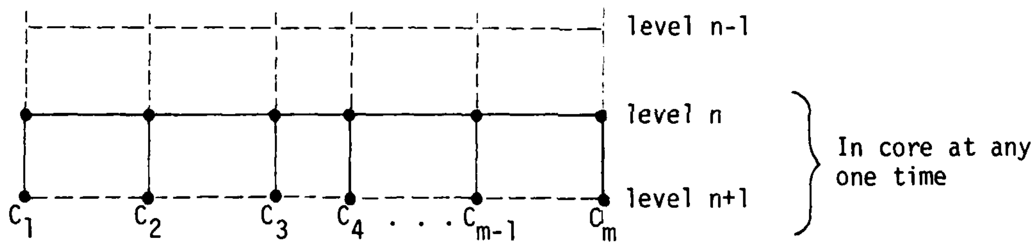
$$\begin{Bmatrix} \underset{\sim}{R}'_n \\ \underset{\sim}{R}'_{n+1} \\ \underset{\sim}{R}'_L \end{Bmatrix} = \begin{bmatrix} \underset{\sim}{K}'_n & \underset{\sim}{C}'_n & \underset{\sim}{E}'_n \\ \underset{\sim}{C}'^T_n & \underset{\sim}{K}'_{n+1} & \underset{\sim}{E}'_{n+1} \\ \underset{\sim}{E}'^T_n & \underset{\sim}{E}'^T_{n+1} & \underset{\sim}{K}'_L \end{bmatrix} \begin{Bmatrix} \underset{\sim}{r}_n \\ \underset{\sim}{r}_{n+1} \\ \underset{\sim}{r}_L \end{Bmatrix}$$

where the prime indicates that the submatrices may have been modified by previous elimination.

At each level the following steps are performed:

a. Add in the individual member stiffnesses for level n.

These are shown below:



b. Perform the elimination on the equations of the uppermost partition in the equations above

c. Save these reduced equations for subsequent back-substitution.

d. Rearrange the submatrices in the equation above appropriately in order to proceed to the next level. This rearrangement is as follows:

$$\begin{Bmatrix} \underline{R}'_{n+1} \\ \\ \underline{0} \\ \\ \underline{R}'_L \end{Bmatrix} = \begin{bmatrix} \underline{K}'_{n+1} & \underline{0} & \underline{E}'_{n+1} \\ \\ \underline{0} & \underline{0} & \underline{0} \\ \\ \underline{E}'^T_{n+1} & \underline{0} & \underline{K}'_L \end{bmatrix} \begin{Bmatrix} \underline{r}_{n+1} \\ \\ \underline{r}_{n+2} \\ \\ \underline{r}_L \end{Bmatrix}$$

e. Repeat the above steps for the next level. Thus after the elimination is completed for joint displacements at all story levels, we are left with the lateral stiffness matrix for the frame.

(iii). Rigid Joint Offset For Beams and Columns

The deformations within the joint, an area bounded by the finite dimensions of any beam and column intersection (shown shaded in Figure 10) are neglected. In other words, this area is assumed to be an infinitely rigid rectangular diaphragm.

This is achieved by providing rigid offsets at the ends of the beams equal in length to one half of the widths of the column below at each corresponding end. Rigid offsets are also provided at each end of the columns equal to the depth of the larger of the beams on either side of the column at the corresponding level.

It has been found that, in general, a reduction in the lengths of the rigid offsets to compensate for some deformation that may exist in the joint is justifiable and gives better results, especially in cases where the member dimensions are substantial. See Reference 9.

Reduction of the rigid link dimension has been coupled to the size of the member. In other words, the rigid link is calculated as described above and then is reduced by 25% of the dimension of the member, at each end.

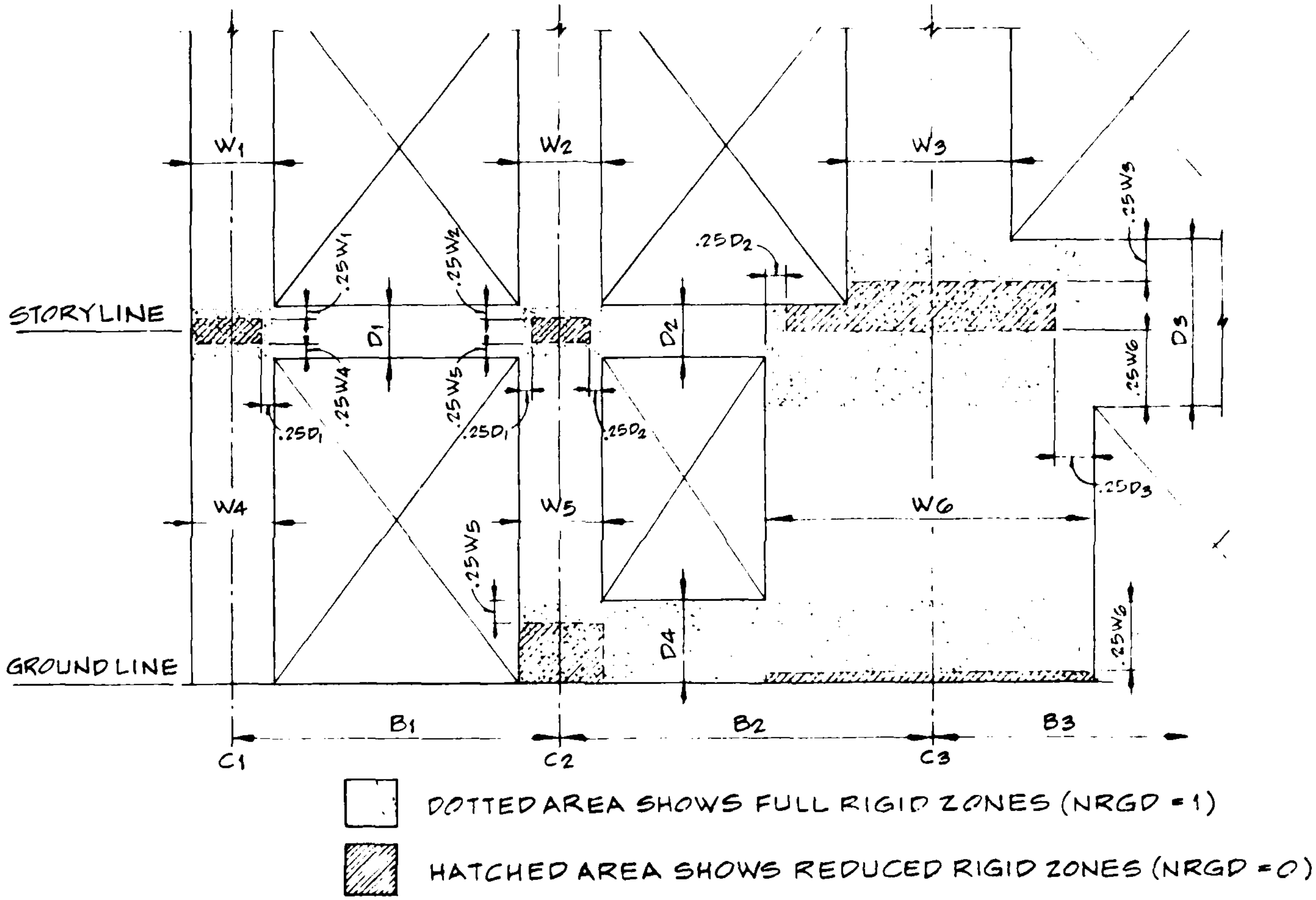


Figure 10. Illustration of rigid zone reduction methodology

Thus the beam rigid links are reduced by 25% of the beam depth and the column rigid links are reduced by 25% of the column width at each end. The reduction cannot, of course, result in a negative rigid link length. This reduction procedure is optional. If no column widths or beam lengths are input the rigid link lengths degenerate to zero and the analysis is carried out on the frame grid line basis.

B. The Complete Structure

In order to combine the frame lateral stiffness matrices into a complete structure lateral stiffness matrix, each of the frame stiffnesses must be transformed to a common displacement coordinate system (which will be referred to as the global system). The global system chosen is two translations and one rotation per story. The origin of these global displacement coordinates at each story level is taken at the center of mass of that story segment. This position may vary from story to story. Such a formulation will degenerate the mass matrix to a diagonal form, thus simplifying the eigen-value problem in the dynamics.

The first step is to develop the transformation between the frame lateral displacements and the global displacements. With reference to Figure 11, the transformation at any level, n, is as follows:

$$r_{Ln} = \langle \cos\alpha \quad \sin\alpha \quad -d_n \rangle \begin{Bmatrix} r_{xn} \\ r_{yn} \\ r_{\theta n} \end{Bmatrix}$$

or:

$$r_{Ln} = \underset{\sim}{a}_n \, \underset{\sim}{r}_n$$

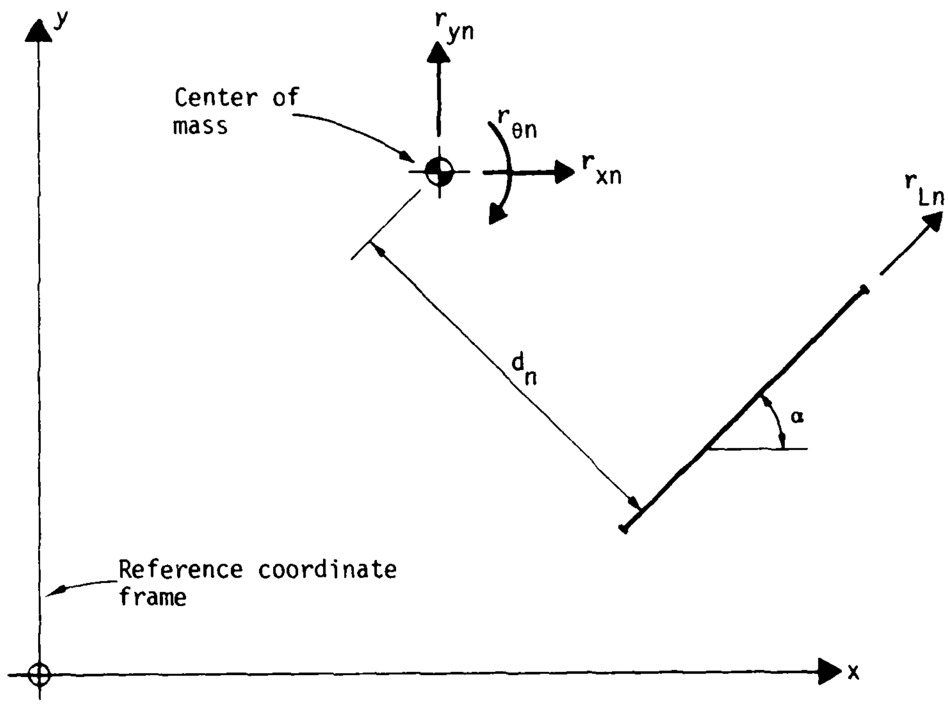


Figure 11. Structural global lateral displacements and frame local lateral displacements

Assembling the transformations for all floors, we obtain the complete transformation between frame lateral displacements and global displacement as follows:

$$\begin{Bmatrix} r_{L1} \\ r_{L2} \\ \cdot \\ \cdot \\ r_{Ln} \\ \cdot \\ \cdot \\ r_{LN} \end{Bmatrix} = \begin{bmatrix} \underset{\sim}{a}_1 & & & & & & \\ & \underset{\sim}{a}_2 & & & & & \\ & & \cdot & & & & \\ & & & \cdot & & & \\ & & & & \underset{\sim}{a}_n & & \\ & & & & & \cdot & \\ & & & & & & \cdot \\ & & & & & & & \underset{\sim}{a}_N \end{bmatrix} \begin{Bmatrix} \underset{\sim}{r}_1 \\ \underset{\sim}{r}_2 \\ \cdot \\ \cdot \\ \underset{\sim}{r}_n \\ \cdot \\ \cdot \\ \underset{\sim}{r}_N \end{Bmatrix}$$

or:

$$\underset{\sim}{r}_{Li} = \underset{\sim}{A}_i \ \underset{\sim}{r}$$

$\underset{\sim}{r}$ is the complete vector of global displacements. The frame lateral stiffness is transformed to the global system and becomes:

$$\underset{\sim}{K}_i = \underset{\sim}{A}_i^T \ \underset{\sim}{K}_{Li} \ \underset{\sim}{A}_i$$

where the subscript i denotes the ith frame.

The structure lateral stiffness is assembled by the addition of components from all frames: i.e.,

$$\underset{\sim}{K} = \sum_i \underset{\sim}{K}_i$$

The frame lateral load vector from sidesway effects must also be transformed to the global system. This transformation is shown by:

$$\underset{\sim}{R}_i = \underset{\sim}{A}_i^T \ \underset{\sim}{R}'_{Li}$$

The global load vector is formed by the summation of frame sway effects and the addition of externally applied lateral loads $\underline{F}$, i.e.:

$$\underline{R} = \sum_i \underline{R}_i + \underline{F}$$

The global forces $\underline{F}$ are specified; however, they are also given by:

$$\underline{F} = \Sigma\ A_i^T\ P_{Li}$$

Expanding the tri-matrix product:

$$\underline{K}_i = \underline{A}_i^T \quad \underline{K}_{Li} \quad \underline{A}_i$$

we get:

$$\begin{bmatrix} \underline{K}_{11} & \underline{K}_{12} & \cdot & \cdot & \cdot & \cdot \\ \underline{K}_{21} & \underline{K}_{22} & \cdot & \cdot & & \cdot \\ \cdot & & \cdot & & & \cdot \\ \cdot & & & \underline{K}_{ij} & & \cdot \\ \cdot & & & & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & K_{NN} \end{bmatrix} = \begin{bmatrix} \underline{a}_1^T & & & & \\ & \underline{a}_2^T & & & \\ & & \cdot & & \\ & & & \underline{a}_i^T & \\ & & & & \cdot \\ & & & & & \underline{a}_N^T \end{bmatrix} \begin{bmatrix} k_{11} & k_{12} & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ & & & \\ & & & \\ & & & \\ \cdot & \cdot & & k_{NN} \end{bmatrix} \begin{bmatrix} \underline{a}_1 & & & & & \\ & \underline{a}_2 & & & & \\ & & \cdot & & & \\ & & & \underline{a}_j & & \\ & & & & \cdot & \\ & & & & & \underline{a}_N \end{bmatrix}$$

It is worth noting that a typical 3 x 3 submatrix $\underline{K}_{ij}$ within $\underline{K}_i$ has the form $\underline{a}_i^T\ \underline{k}_{ij}\ \underline{a}_j$. Obviously this product may be formed independently for each term in and added directly into $\underline{K}$. Hence the global equilibrium equations are formed.

$$\underline{R} = \underline{K}\ \underline{r}$$

It may be noted that the global stiffness $\underline{K}$ is a full matrix, but it is of course relatively small compared to the total number of degrees of freedom associated with all the frames in the structure.

CHAPTER III: STATIC ANALYSIS

The static analysis equations:

$$R = k\,r$$

are solved directly by Gaussian elimination giving a vector of global lateral displacements, $\underline{r}$. Next, for each frame, the lateral displacements, $\underline{r}_{Li}$ are computed using:

$$r_{Li} = A_i\,r$$

To complete the solution for each frame, the following system is considered.

$$\underline{R}'_n = \begin{bmatrix} \underline{K}'_n & C'_n & E'_n \end{bmatrix} \begin{bmatrix} \underline{r}_n \\ \underline{r}_{n+1} \\ \underline{r}_L \end{bmatrix}$$

Note that these are the equations which were reduced, then saved at each level, n, of the frame. That is, $\underline{K}'_n$ was triangularized. At any stage, n, $\underline{r}_{n+1}$ and $\underline{r}_L$ are known and so $\underline{r}_n$ is computed by back substitution. To start this sequence, we simply note that for n = N (the number of stories in the structure) $\underline{r}_{N+1}$ represents the displacements at the foundation which are zero since columns are assumed rigidly connected to the foundation. Thus the frame joint displacements are computed successively story by story and individual member forces may be computed at the same time from the force/deformation transformations previously presented.

A. Vertical Loads Analysis

The vertical loads are applied on each individual frame as beam span loads. Four independent vertical loading conditions are possible. The self weight of the frames can be automatically calculated by the program and added to the load vector of the first load condition. Typically the first load condition is used for the dead load analysis of the structure; the second load condition is used for the live load analysis of the structure. The third and fourth load conditions may be used for skip live loading or left unused.

B. Lateral Load Analysis

The lateral static loads are applied as forces acting at a particular point on each floor level. Two independent lateral loading conditions are possible. The lateral loads may be due to wind or earthquake. The wind loads have to be calculated and input by the user, based upon the wind pressure and the exposed tributary area of the building at each level of the structure. The seismic static equivalent loads may be automatically calculated by the program, based upon the requirement of Reference 14. The modal participation factors calculated by the program are used to determine the predominant directions of the modes and the time periods of the predominant modes are used in calculating the seismic loads in the corresponding directions.

The program has options to calculate the dynamic properties, such as the mass and mass moment of inertia of each floor level based upon simplified user input.

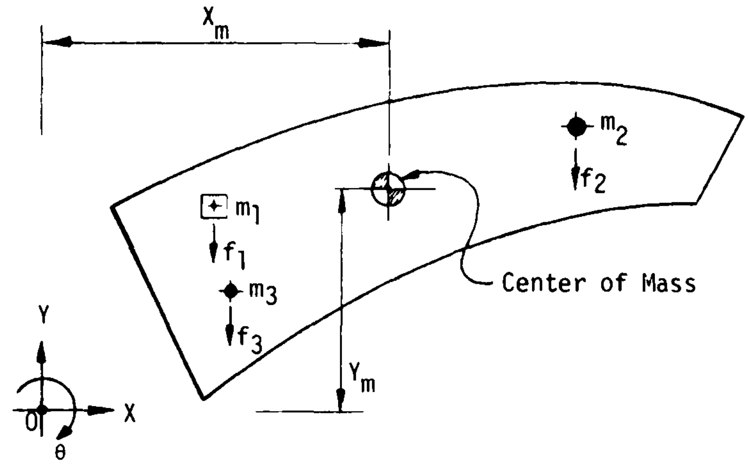
CHAPTER IV: DYNAMIC ANALYSIS

The exact formulation of the dynamic response of a structure involves an infinite number of degrees of freedom. For most structures, however, the response may be adequately captured by a limited number of discrete points (or joints) within the system. In the buildings considered here, the response may be described by the lateral motions of each floor level, as previously described for the formation of the lateral structure stiffness matrix. The center of mass is used as the master constraint location at each level in the generation of the lateral stiffness matrix. The tributary mass of each story level is lumped at the center of mass of the level along with the mass moment of inertia of the floor about a vertical axis through the center of mass to compensate for the rotational aspects of the lumping process. The resulting mass matrix is of diagonal form. With this lumped parameter idealization, equilibrium of the structure is described by a set of ordinary second order differential equations.

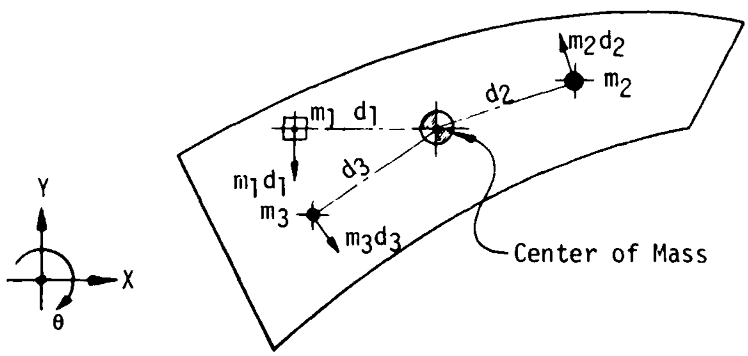
A. Mass Approximation, Mass and Mass Moment of Inertia

In the diaphragm shown in Figure 12, there are various lumped masses (m_1, m_2, m_3 . . . etc.) and other distributed masses associated with the diaphragm level.

When the diaphragm is subjected to a unit translational acceleration in the Y-direction, inertia forces opposing the direction of the acceleration will be generated, i.e. $f_1 = m_1 \times 1$, $f_2 = m_2 \times 1$, $f_3 = m_3 \times 1$ . . . The resultant of all these forces and line of action of the resultant can be determined. The magnitude of the resultant is found to be = f_1 +



(a) DEFINITION OF MASS AND CENTER OF MASS



(b) DEFINITION OF MASS MOMENT OF INERTIA

Figure 12

$f_2 + f_3 \ldots = m_1 + m_2 + m_3 \ldots$ = total mass associated with the diaphragm. The resultant is parallel to the Y-direction and passes through a point at a distance X_m from 0.

Similarly, a unit translation in the X-direction will give a resultant of the same magnitude but parallel to the X-direction and passing through a point a distance Y_m from 0.

The coordinates X_m, Y_m define the location of a point known as the center of mass.

Redefining the term, Mass: "The mass of a diaphragm may be defined as the force generated when the center of mass of the diaphragm undergoes a unit translational acceleration. This force acts at the center of mass, resulting in no associated moment."

$$\text{Mass} = \sum_{i=1}^{n} m_i$$

Similarly, defining the term, Mass Moment of Inertia (or Rotational Mass): "The mass moment of inertia of a diaphragm may be defined as the moment generated when the center of mass of the diaphragm undergoes a unit rotational acceleration about a vertical axis. No resultant translational force is associated with the couple."

The radial distances from the center of mass of the lumped masses m_1, m_2, $m_3 \ldots$ are d_1, d_2, $d_3 \ldots$ respectively, as shown in Figure 12.

Due to a unit rotational acceleration of the center of mass about a vertical axis, m_1, m_2, $m_3 \ldots$ will have translational accelerations of $d_1 x1$, $d_2 x1$, $d_3 x1 \ldots$ Thereby giving corresponding inertia forces of

m_1d_1, m_2d_2, m_3d_3 . . . . The moments of these forces about a vertical axis through the center of mass are $m_1d_1^2$, $m_2d_2^2$, $m_3d_3^2$ . . . .

$$\therefore \text{MMI} = m_1d_1^2 + m_2d_2^2 + m_3d_3^2 \ldots.$$

$$= \sum_{i=1}^{n} m_i d_i^2$$

= Polar Moment of Inertia of all Masses, about a vertical axis through the center of mass

B. <u>Dynamic Equilibrium Equations</u>

The equilibrium equations for a structure, including dynamic effects, may be written in the following form:

$$\underline{M}\,\ddot{\underline{r}}_a + \underline{C}\,\dot{\underline{r}} + \underline{K}\,\underline{r} = P\,(t) \qquad \ldots\ldots.\ (a)$$

where: $\underline{M}$ = mass matrix

$\underline{C}$ = damping matrix

$\underline{K}$ = stiffness matrix

$\underline{P}\,(t)$ = applied load vector, which may be time dependent

$\underline{r}$ = displacement vector of deformation relative to support motion

$\ddot{\underline{r}}_a$ = absolute acceleration vector

$\underline{r}$ and $\underline{r}_a$ are related in the following fashion:

$$\underline{r}_a = \underline{v}_g + \underline{r}$$

where $\underline{v}_g$ is the vector of pseudo-static displacements due to support movement. Also:

$$\ddot{\underline{r}}_a = \ddot{\underline{v}}_g + \ddot{\underline{r}}$$

These vectors have the following form for a typical floor, of a building shown in Figure 13.

$$\begin{Bmatrix} r_{xa} \\ r_{ya} \\ r_{\theta a} \end{Bmatrix}_n = \begin{Bmatrix} v_{gx} \\ v_{gy} \\ v_{g\theta} \end{Bmatrix} + \begin{Bmatrix} r_{xn} \\ r_{yn} \\ r_{\theta n} \end{Bmatrix} = \begin{Bmatrix} \sin\beta \\ \cos\beta \\ 0 \end{Bmatrix} v_g + \begin{Bmatrix} r_{xn} \\ r_{yn} \\ r_{\theta n} \end{Bmatrix}$$

and:

$$\begin{Bmatrix} \ddot{r}_{xa} \\ \ddot{r}_{ya} \\ \ddot{r}_{\theta a} \end{Bmatrix}_n = \begin{Bmatrix} \sin\beta \\ \cos\beta \\ 0 \end{Bmatrix} \ddot{v}_g + \begin{Bmatrix} \ddot{r}_{xn} \\ \ddot{r}_{yn} \\ \ddot{r}_{\theta n} \end{Bmatrix}$$

i.e.:

$$\underline{r}_{na} = \underline{b}\, \ddot{v}_g + \underline{r}_n$$

Or, for all floors:

$$\underline{r}_a = \underline{B}\, V_g + \underline{r}$$

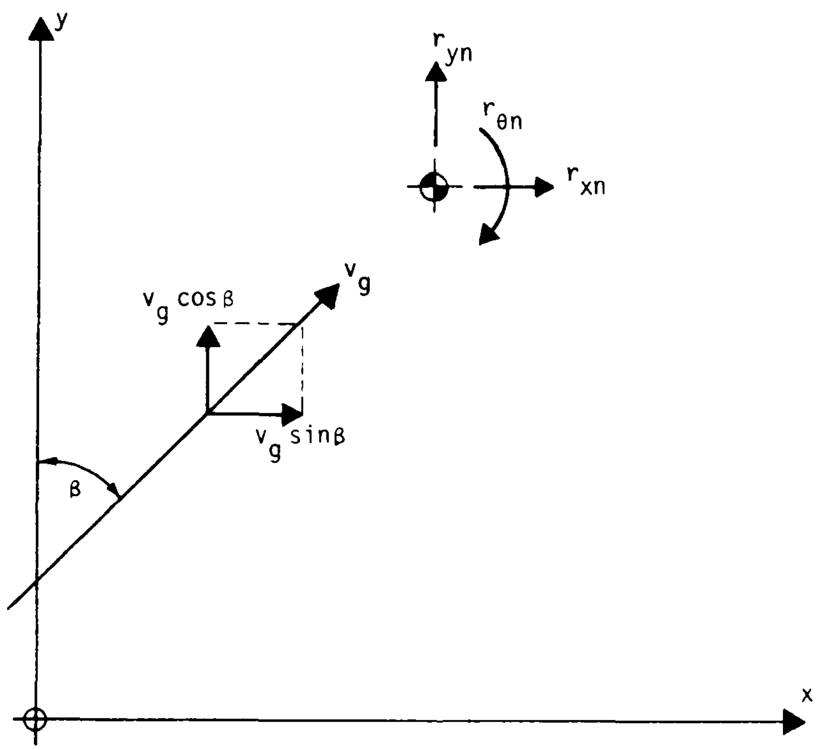


Figure 13. Ground and structural displacements

where:

$$\underline{B} = \begin{Bmatrix} \underline{b}_1 \\ \underline{b}_2 \\ \underline{b}_3 \\ \cdot \\ \cdot \\ \underline{b}_N \end{Bmatrix} ; \quad \underline{b}_1 = \underline{b}_2 \text{ etc.}$$

In the case of seismic analysis, there are no externally applied loads; i.e., P (t) = 0. Then equation (a) may be written as:

$$\underline{M} (\ddot{\underline{r}} + \underline{B} \ddot{v}_g) + \underline{C} \dot{\underline{r}} + \underline{K} \underline{r} = \underline{0}$$

or:

$$\underline{M} \ddot{\underline{r}} + \underline{C} \dot{\underline{r}} + \underline{K} \underline{r} = - \underline{M} \underline{B} \ddot{\underline{v}}_g \qquad \ldots\ldots (b)$$

This coupled set of equations may be solved simultaneously with an appropriate numerical technique. Another approach, which will be used here, is to find a transformation which uncouples the equations so that they may be solved independently. This transformation, of course is via the eigen-vectors or mode shapes of the system.

C. Mode Shapes and Frequencies

The vibration mode shapes represent the solution of the undamped free vibration problem given by·

$$\underline{M} \ddot{\underline{r}} + \underline{K} \underline{r} = \underline{0}$$

The eigen-value problem to be solved is written as:

$$\underset{\sim}{K}\ \underset{\sim}{\phi} = \underset{\sim}{\omega}^2\ \underset{\sim}{M}\ \underset{\sim}{\phi}$$

where: $\underset{\sim}{\phi}$ = *mode shapes*

$\underset{\sim}{\omega}$ = frequencies

The mode shapes are normalized such that:

$$\underset{\sim}{\phi}^T\ \underset{\sim}{M}\ \underset{\sim}{\phi} = \underset{\sim}{I}$$

then also:

$$\underset{\sim}{\phi}^T\ \underset{\sim}{K}\ \underset{\sim}{\phi} = \underset{\sim}{\omega}^2$$

Also, it is assumed that the damping matrix $\underset{\sim}{C}$ is of a form that is uncoupled by the mode shapes; specifically it is assumed that:

$$\underset{\sim}{\phi}^T\ \underset{\sim}{C}\ \underset{\sim}{\phi} = [\ 2\lambda_m\ \ \omega_m\ ]$$

so that λ_m represents the damping of the m th mode.

The actual displacements, $\underset{\sim}{r}$, are now expressed as a linear combination of the mode shapes.

$$\underset{\sim}{r} = [\ \underset{\sim}{\phi}_1\ \ \underset{\sim}{\phi}_2\ \ \underset{\sim}{\phi}_3 \cdot \cdot\ \underset{\sim}{\phi}_N\ ] \begin{bmatrix} Z_1(t) \\ Z_2(t) \\ \cdot \\ \cdot \\ Z_N(t) \end{bmatrix} \quad . . . (c)$$

i.e.: $\underset{\sim}{r} = \underset{\sim}{\phi}\ \underset{\sim}{Z}$

also $\dot{\underset{\sim}{r}} = \underset{\sim}{\phi}\ \dot{\underset{\sim}{Z}}$

and: $\ddot{\underline{r}} = \underline{\phi}\,\ddot{\underline{Z}}$

where $Z_m(t)$ represents the response of the m th mode.

D. Time History Analysis

Using equations (b) , equation (c) may be rewritten as:

$$\underline{M}\,\underline{\phi}\,\ddot{\underline{Z}} + \underline{C}\,\underline{\phi}\,\dot{\underline{Z}} + \underline{K}\,\underline{\phi}\,\underline{Z} = -\,\underline{M}\,\underline{B}\,\ddot{v}_g$$

Premultiplication by $\underline{\phi}^T$ yields the uncoupled set of second order equations:

$$\underline{M}^*\,\ddot{\underline{Z}} + \underline{C}^*\,\dot{\underline{Z}} + \underline{K}^*\,\underline{Z} = \underline{P}^*\,\ddot{v}_g$$

where:

$$\underline{M}^* = \underline{\phi}^T\,\underline{M}\,\underline{\phi} = \underline{I}$$

$$\underline{C}^* = \underline{\phi}^T\,\underline{C}\,\underline{\phi} = [\ 2\lambda_m\,\omega_m\ ]$$

$$\underline{K}^* = \underline{\phi}^T\,\underline{K}\,\underline{\phi} = [\ \omega_m^2\ ]$$

$$\underline{P}^*\,\ddot{v}_g = \underline{\phi}^T\,\underline{M}\,\underline{B}\,\ddot{v}_g$$

to find the form of $\underline{P}^*$, consider:

$$\underline{M}\,\underline{B} = \begin{bmatrix} m_1 & & & & & & & & \\ & m_1 & & & & & & & \\ & & J_1 & & & & & & \\ & & & m_2 & & & & & \\ & & & & m_2 & & & & \\ & & & & & J_2 & & & \\ & & & & & & \cdot & & \\ & & & & & & & \cdot & \\ & & & & & & & & J_N \end{bmatrix} \begin{Bmatrix} \sin\beta \\ \cos\beta \\ 0 \\ \sin\beta \\ \cos\beta \\ 0 \\ \cdot \\ \cdot \\ 0 \end{Bmatrix}$$

where: m_1 = mass of story 1

J_1 = rotational mass moment of inertia of story 1

i.e.:

$$\underset{\sim}{M}\,\underset{\sim}{B} = \begin{bmatrix} m_1 \sin\beta \\ m_1 \cos\beta \\ 0 \\ m_2 \sin\beta \\ m_2 \cos\beta \\ 0 \\ \cdot \\ \cdot \\ \cdot \end{bmatrix}$$

So, a typical term of $\underset{\sim}{P}^*$ has the form:

$$P_m^* = \underset{\sim}{\phi}_m^T \; M \; B$$

$$= \langle \phi_{1x} \;\; \phi_{1y} \;\; \phi_{1\theta} \;\; \phi_{2x} \;\; \phi_{2y} \;\; \phi_{2\theta} \;\; \cdot \cdot \cdot \rangle \begin{Bmatrix} m_1 \sin\beta \\ m_1 \cos\beta \\ 0 \\ m_2 \sin\beta \\ m_2 \cos\beta \\ 0 \\ \cdot \\ \cdot \\ \cdot \end{Bmatrix}$$

$$P_m^* = \sum_{n=1}^{N} m_n \; \{ \sin\beta \; \phi_{nx} + \cos\beta \; \phi_{ny} \}$$

Now a typical equation governing the response in the m th mode has the form:

$$\ddot{Z}_m + 2\lambda_m \omega_m \dot{Z}_m + \omega_m^2 Z_m = P_m^* \ddot{v}_g \qquad \ldots\ldots\ldots (d)$$

For any earthquake, the ground acceleration, $\ddot{v}_g$ is specified as a set of discrete values and linear interpolation is used for intermediate values. On any linear portion then:

$$\ddot{v}_g = A + Bt$$

where A and B are computed from the end values as shown in Figure 14.

On any linear segment t_1, t_2 then:

$$\ddot{Z}_m + 2\lambda_m \omega_m \dot{Z}_m + \omega_m^2 Z_m = P_m^* (A + Bt)$$

The solution to this equation is summarized in Figure 14.

At rest initial conditions are used for the first linear portion. The values of displacement and velocity at the end of any linear portion form the initial conditions for the following linear segment and so on. Repetition gives the complete solution over the required time span. With solutions for each mode, equation (c) is used to give a set of structure displacements $\underline{r}$ at each output time step.

The backsubstitution procedure used for the time history analysis is exactly the same as that described for the static analysis in Chapter III. Backsubstitution for each time step is equivalent to one static load backsubstitution. The frame displacements and member forces are determined at each time step and the maxima of these parameters over the time span are output as dynamic load condition 3.

For the equation: $\ddot{Z}_m + 2\lambda_m\omega_m\dot{Z}_m + \omega^2 Z_m = P_m^* (A + Bt)$

On any linear segment such as t_1, t_2, the solution is given by:

$$Z_m(t) = P_m^* e^{-\lambda_m\omega_m t}\left\{\left[Z_m(t_1) - \frac{A}{\omega_m^2} + \frac{2\lambda_m B}{\omega_m^3}\right] \cos \omega_{Dm} t\right.$$

$$\left. + \frac{1}{\omega_{Dm}}\left[\dot{Z}_m(t_1) + \lambda_m\omega_m Z_m(t_1) - \frac{\lambda_m A}{\omega_m} + \frac{B(2\lambda_m^2 - 1)}{\omega_m^2}\right] \sin \omega_{Dm} t\right\}$$

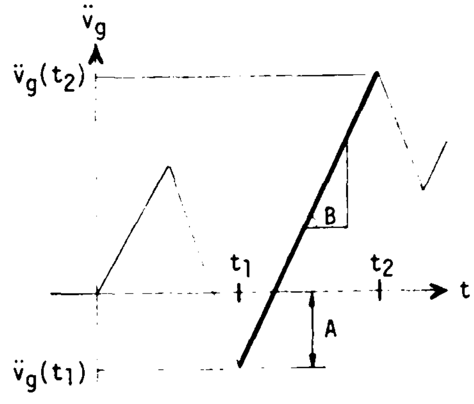
$$+ P_m^* \left[\frac{A}{\omega_m^2} - \frac{2\lambda_m B}{\omega_m^3} + \frac{Bt}{\omega_m^2}\right]$$

and:

$$\dot{Z}_m(t) = P_m^* e^{-\lambda_m\omega_m t}\left\{\left[\dot{Z}_m(t_1) - \frac{B}{\omega_m^2}\right] \cos \omega_{Dm} t\right.$$

$$\left. + \left[A - \omega_m^2 Z_m(t_1) - \lambda_m\omega_m \left(Z_m(t_1) + \frac{B}{\omega_m^2}\right)\right] \frac{\sin \omega_{Dm} t}{\omega_{Dm}}\right\}$$

$$+ P_m^* \frac{B}{\omega_m^2}$$



ACCELEROGRAM

where:

$A = \ddot{v}_g(t_1)$

$B = \dfrac{\ddot{v}_g(t_2) - \ddot{v}_g(t_1)}{t_2 - t_1}$

$\omega_{Dm} = \omega_m (1 - \lambda_m^2)^{1/2}$

$Z_m(t_1)$, $\dot{Z}_m(t_1)$ are the initial conditions for the segment t_1, t_2

Figure 14. Closed form time integration scheme of CTABS80

E. Response Spectrum Analysis

Unless actual histories of displacements and forces are required for a specific earthquake a more realistic and economical approach for dynamic analysis is via the response spectrum method. For a particular ground motion history $\ddot{v}_g(t)$, the spectrum curve is defined as follows:

The response of a single mass system with damping λ, and circular frequency ω, subjected to a ground motion history $\ddot{v}_g(t)$ is governed by the equation:

$$\ddot{u}(t) + 2\lambda\omega\,\dot{u}(t) + \omega^2 u(t) = \ddot{v}_g(t)$$

Let u_{max} be the maximum absolute value that $u(t)$ attains. A plot of this maximum displacement versus the frequency $\underline{\omega}$ for each λ is by definition the displacement response spectrum (Sd) for the earthquake $\ddot{v}_g(t)$. A plot of $u_{max}\omega$ is the pseudo-velocity spectrum (PSv) and a plot of $u_{max}\omega^2$ is the pseudo-acceleration spectrum (PSa). These pseudo-velocity and acceleration spectra are of the same physical interest but are not an essential part of a response spectrum analysis.

Recalling equation (d), if the dynamic loading on the structure is specified in terms of the pseudo-acceleration spectrum, then the maximum response for the m th mode is given by:

$$Z_{m_{max}} = P^*_m \frac{PSa\ (\omega_m, \lambda_m)}{\omega_m^2}$$

Therefore, the maximum contribution of mode m to the total three dimensional response of the structure is:

$$\underline{r}_m = Z_{m_{max}} \underline{\Phi}_m$$

For all modes Sd is, by definition, positive. The maximum modal displacement $\underline{r}_m$ is proportional to the mode shape $\underline{\Phi}$; and the sign of the proportionality constant is given by the sign of the modal participation factor, P^*_m. Therefore, each maximum modal displa ient has a unique sign. Also, the maximum internal modal forces, which are consistently evaluated from the maximum modal displacements, have unique signs.

A complete analysis is performed down to the member force level with the maximum modal displacements of the structure for each individual mode using the backsubstitution procedure described in Chapter II.

The maxima in each mode will generally occur at different times. The combination of the modal components of the displacements and member forces to give resultant values for design purposes is performed at the design parameter level by the following methods.

1. The Square-Root-of-the-Sum-of-the-Squares (SRSS)[13] method
2. The Absolute Sum (ABS)[13] method
3. The Complete Quadratic Combination (CQC)[10] method

The SRSS method and the ABS method entirely neglect the signs of the modal contributions. The SRSS method in general gives good approxi-

mations of the dynamic response in structures with well separated frequencies. The ABS method is basically for interest to give an upper bound on the maximum values.

In structures with closely spaced modes or multiple frequencies, the fact that the SRSS method neglects the signs of the modal components may cause the design parameters to be dramatically overestimated in some elements while being significantly underestimated in other elements. The CQC method overcomes this difficultv and it is recommended as the best of the three methods for obtaining the most realistic results.

F. Dynamic Options

The dynamic options currently available in CTABS80 are:

1. Calculation of mode shapes and periods (frequencies)
2. Response spectrum analysis for any acceleration spectrum supplied by the user using the:
    a. SRSS modal combination as Dynamic load condition I
    b. Sum of absolute value modal combinations as Dynamic load condition 2
    c. Complete Quadratic combinations as Dynamic load condition 3
3. Time history analysis maxima for any ground motion supplied by the user as Dynamic load condition 3

Either dynamic analysis condition may be combined with any static load condition.

1. SRSS Combination

$$\underset{(1\times1)}{F} = \sqrt{\underset{(1\times n)}{\underline{f}^T} \; \underset{(n\times n)}{\underline{I}} \; \underset{(n\times1)}{\underline{f}}}$$

Where $\underline{I}$ is an identity matrix

2. ABS Combination

$$\underset{(1\times1)}{F} = \underset{(1\times n)}{f^T} \; \underset{(n\times1)}{\text{sign } \underline{f}}$$

Where sign $\underline{f}$ is a unit matrix containing the signs of the corresponding elements of matrix $\underline{f}$

3. CQC Combination

$$\underset{(1\times1)}{F} = \sqrt{\underset{(1\times n)}{\underline{f}^T} \; \underset{(n\times n)}{\underline{C}} \; \underset{(n\times1)}{\underline{f}}}$$

Where $\underline{C}$ is the matrix of modal cross-correlation coefficients given by:

$$C_{ij} = \frac{8\lambda^2(1+r)r^{3/2}}{(1-r^2)^2+4\lambda^2r(1+r)^2}$$

where $r = \omega_i/\omega_j$, the ratio of the circular frequencies of the coupling modes and λ is the damping associated with the response spectrum curve being used.

NOTES/

f = vector of modal components

F = combined resultant

n = number of modes

Figure 15. Summary of modal combination techniques used in CTABS80

CHAPTER V: GENERAL OBSERVATIONS

A. Program Application

The effective application of a computer program for the analysis of practical situations involves a considerable amount of experience. The most difficult phase of the analysis is in assembling an appropriate model which captures the major characteristics of the structural behavior of the building. No computer program can replace the engineering judgement of an experienced engineer. It is well said that an incapable engineer cannot do with a ton of computer output what a good engineer can do on the back of an envelope. Correct output interpretation is just as important as the preparation of a good structural model. Verification of unexpected results needs a good understanding of the basic assumptions and the mechanics of the program. Static equilibrium checks are necessary not only to check the computer output but to understand the basic structural behavior of the building.

B. Static Seismic Analysis of Buildings

At the present time, the seismic design of most buildings in California and other earthquake regions of the United States is based upon the Uniform Building Code. The UBC method allows the seismic loads to be approximated by an equivalent set of lateral static loads. The magnitude of the loads is based upon the seismic zone, the structural system, and the fundamental period of the structure. Corrections to compensate for local soil conditions and the physical importance of the structure are also defined.

An approximate formula, specified in the UBC, may be used to estimate the fundamental period. The period associated with the predominant structural mode obtained via the TABS program is more accurate and appropriate. The suggested UBC distribution of the lateral loads over the height of the building is triangular with some correction to allow for higher mode effects. Behavior of structures that have dynamically decoupled regions due to stiffness and/or mass discontinuities, causing significantly non-triangular inertia load patterns are not adequately covered by the code. By examining the structural modes produced by a TABS analysis such structural complexities can be isolated.

The determination of the minimum horizontal torsional design moments, as specified by the UBC for the design of structures having rigid diaphragms, requires the location of a center of rigidity of the structure at each level. The definition of torsional moments on such a basis for multi-story structures is technically vague. It is only meaningful in single story structures where there are no stories above or below to affect the rotation of the level under consideration.

The UBC lateral loads are only a small fraction of the loads developed during a significant earthquake, and must therefore be considered as minimum requirements. As a result of the above mentioned inadequacies the need for a more comprehensive code earthquake analysis methodology is apparent to most structural engineers[8].

C. Computer Methods Versus Hand Methods

High speed digital computers and the development of computer programs such as TABS have given engineers the capability to consider aspects of

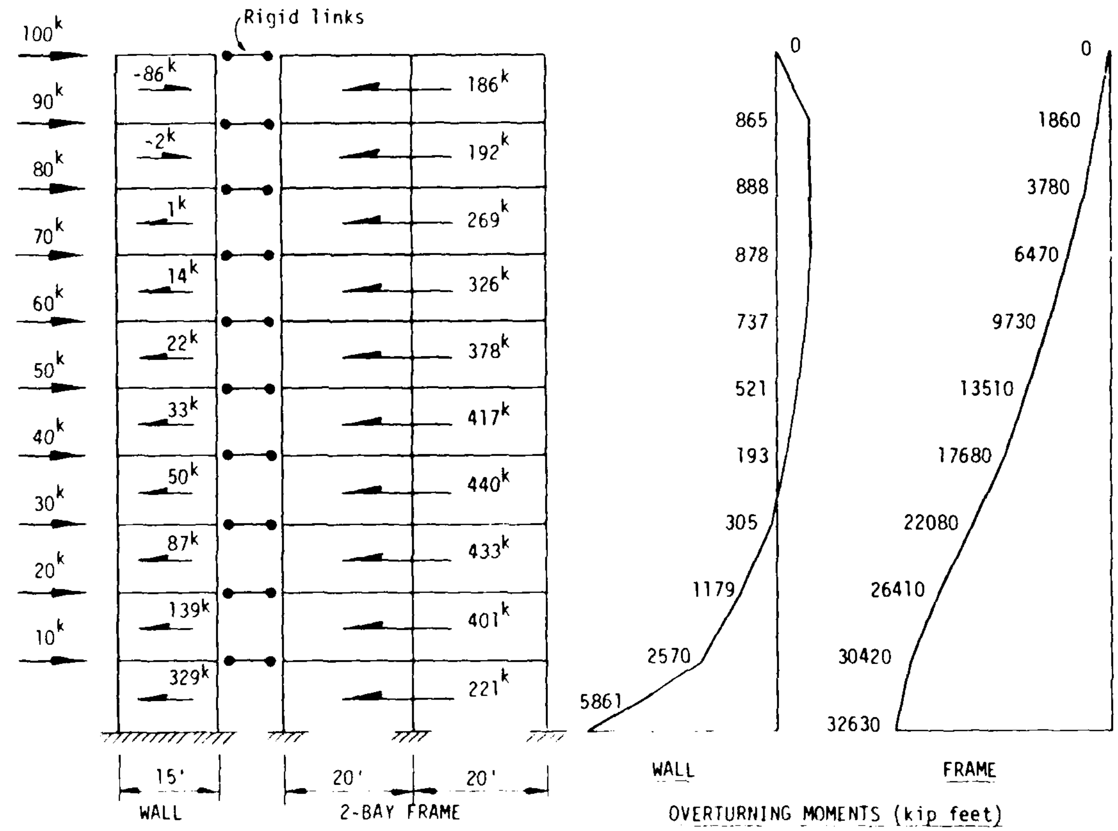
structural behavior that conventional hand analysis techniques have traditionally neglected. Hand analysis techniques used by practicing engineers for the lateral analysis of multistory structures have been shown to violate joint statics and compatibility.

The following examples demonstrate the degree of error that could be present in a conventional hand analysis, by comparison with a TABS analysis; i.e., one that completely satisfies statics, compatibility, and boundary. Examples presented are of simple symmetrical multistory buildings with symmetrical loading. The stories, therefore, translate under lateral load without rotating, thereby keeping the problems clear and demonstrative. The proportions of the structures and magnitudes of the forces have been chosen to generate problems of a nature that a conventional structural engineer is commonly faced with in practice.

(i). Example 1

This is a classic example of shear wall-frame interaction. A 10 story shear wall is connected in parallel with a ten story frame at each story level through a rigid link. The axial deformations in the beams are neglected, thus simulating a rigid diaphragm. Therefore, the lateral displacements of the respective stories of the frame and shear wall are equal. See Figure 16.

Consider the top story. Based on a conventional hand analysis, one would, in general, tend to ignore the stiffness of the frame and conclude that the shear wall takes close to 100% of the applied 100 K, and that the frame being relatively flexible gets a negligible amount of the shear.



NOTE/

Walls are 12" thick
Columns are 24" x 24" typ.
Beams are 12" x 24" typ.
Story height = 10' typ.

Figure 16

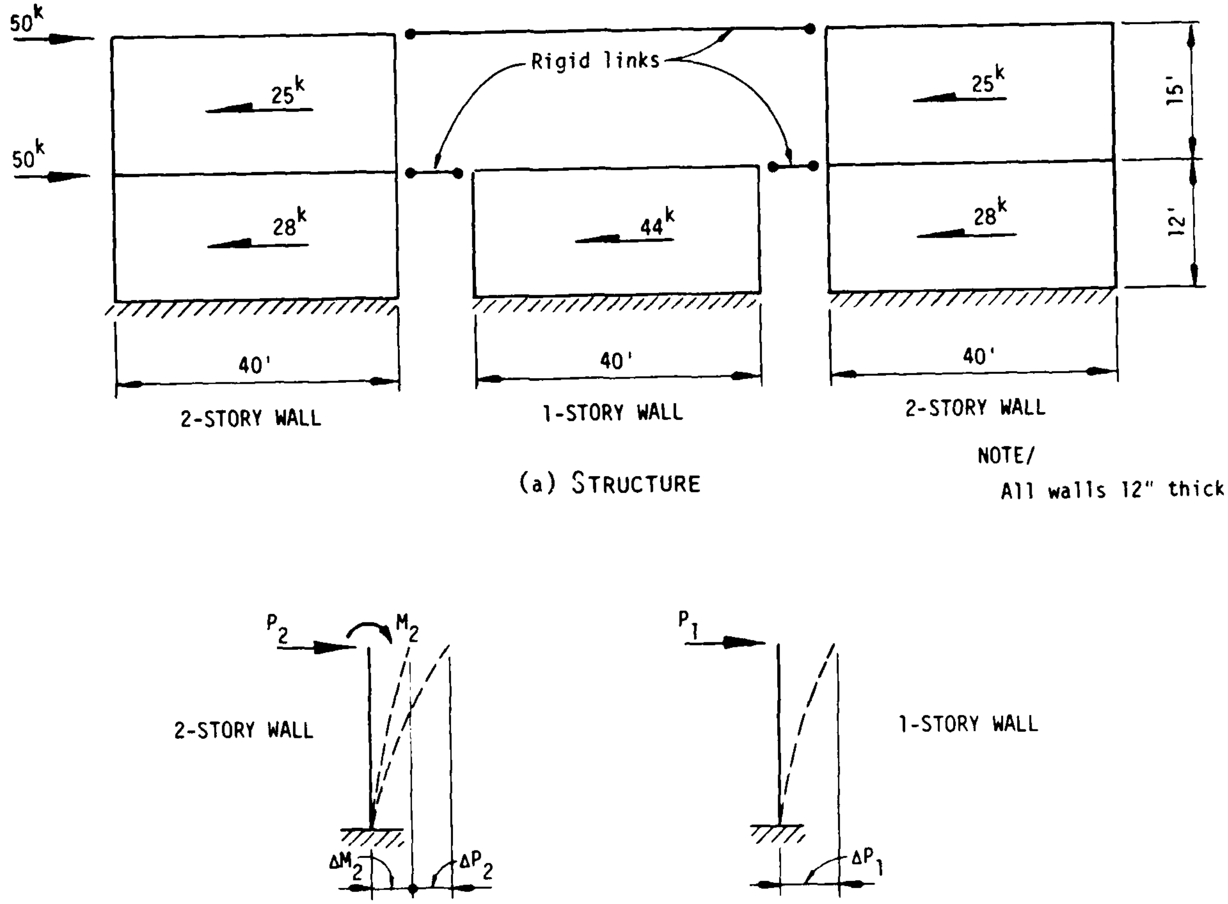
A "correct" analysis, however, reveals that the frame has a total shear of 186 K at that level. Thus, the frame, besides carrying the total 100 K applied load, is laterally supporting the shear wall which puts an additional 86 K on the frame. The shear wall is in effect acting as a propped cantilever supported by the frame at the upper story levels.

The phenomenon may be explained as follows: If the frame and shear wall are loaded independently with the load, and the lateral displacements of the respective floors compared, it will be observed that the shear wall has larger lateral displacements in the upper levels, whereas the frame has larger displacements in the lower stories and vice versa. Compatibility of joint rotations will have a significant effect on these displacement patterns. When loaded together, the constraint of equal story displacements is enforced, thus resulting in this unique shear distribution. Notice that in the lower stories the shear gradually shifts to the shear wall.

This example demonstrates the importance of the interaction of all the elements on one another and that the hand analysis method of analyzing an n-story structure as n 1-story structures stacked one over the other with no interaction of one on the other can lead to highly unreliable results.

(ii). Example 2

This problem consists of two 2-story walls and one 1-story wall. See Figure 17a. Again, as in Example 1, the walls are connected by rigid links at the floor levels to simulate a rigid diaphragm and to enforce



(a) STRUCTURE

(b) FREE BODY DIAGRAM OF LOWER LEVEL

Figure 17

equal lateral story displacements in all the walls. The walls are loaded with a total story load of 50 K at each level. As expected in a conventional hand analysis, the 2-story walls equally share the 50 K lateral load at the upper level.

Now let us consider the shear distribution in the bottom story. The base shear is 100 K. In the light of the fact that there are three resisting elements at this level, all 40' long and 1' thick, a conventional hand analysis would conclude that the base shear will be carried equally by all three elements; that is 33 K each.

A "correct" analysis, however, indicates that the 1-story wall takes over 50% more shear than each 2-story wall. In Figure 17b are presented the free body diagrams of the lower levels of the 1-story wall and the 2-story walls. Consider the lateral story displacements of the 1st level in each wall. In the 1-story wall the lateral displacement, ΔP_1, is due to P_1, the shear force in the wall. In the 2-story wall the lateral displacement is due to two factors. Firstly, ΔP_2, that is due to P_2, the shear force in the wall and, secondly, ΔM_2, due to M_2, the moment at the top of the wall due to the fact that the wall is 2 stories high. Now for the lateral displacements to be equal:

$$\Delta P_1 = \Delta P_2 + \Delta M_2$$

Therefore $\quad \Delta P_1 > \Delta P_2$

so that $\quad P_1 > P_2$

The discrepancy between the conventional hand analysis method and the "correct" method here, again, is due to the fact that the effect of the upper story on the lower story is accounted for incompletely.

(iii). Example 3

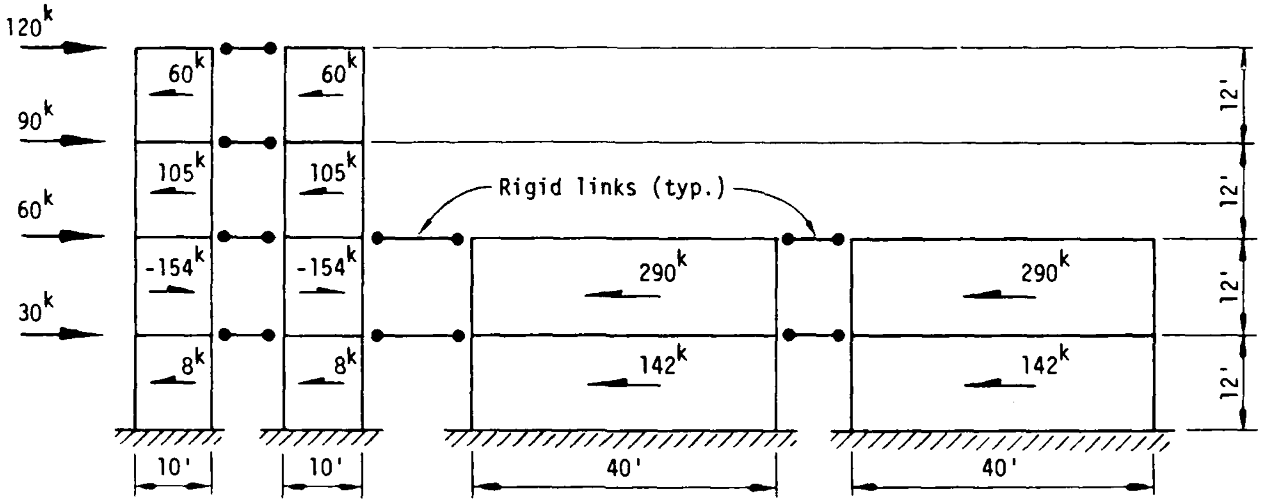
This structure consists of 4 walls. Two 4-story walls and two 2-story walls. See Figure 18a. Again, the walls are connected by rigid links at the floor levels to simulate a rigid diaphragm.

In the "correct" analysis the shear forces in the 10 foot walls in the 3rd and 4th levels are as would be expected in a conventional hand analysis. Note the shear distribution at the 2nd level. At this level there is a considerable increase in the story stiffness due to the two 40-foot walls. This restricts the lateral diaphragm movement to the extent that the 10' walls are in effect laterally supported by the diaphragm at this level, and, therefore, behave like over-hanging cantilevers as shown in Figure 18b. This explains why these walls have a negative shear of 154 K each at this level.

The conventional hand analysis method for such problems completely disregards the possibility of negative shear forces occurring in the walls, thereby always assuming that the walls support the diaphragm laterally, and not recognizing that at times the diaphragm may actually be the support for certain walls at certain levels.

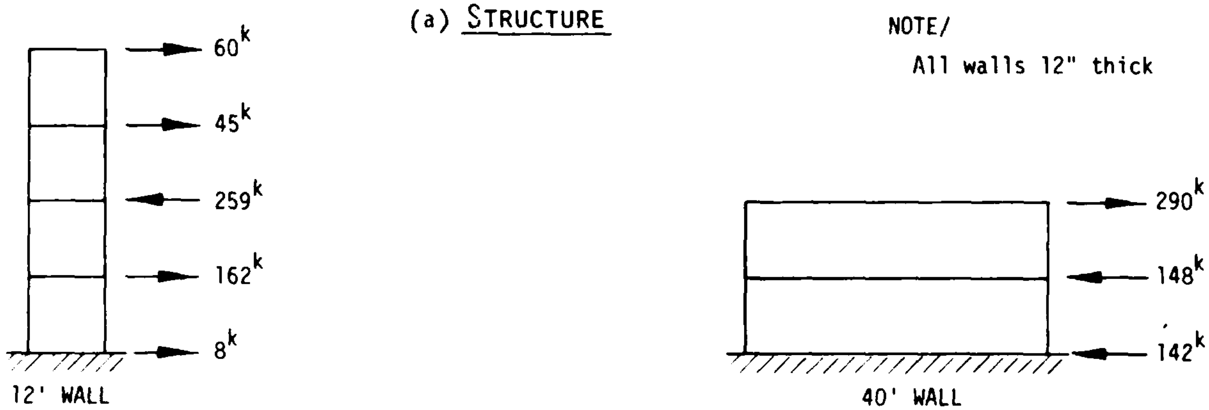
(iv). Example 4

This example demonstrates the effect of axial deformations on the distribution of shear to a series of walls, Consider the structure shown in Figures 19a and 19b. The structure consists of two solid walls, and one wall terminating on two columns. In case A, the columns are 12" square, see Figure 19a. Again, the walls are connected by rigid



(a) STRUCTURE

NOTE/
All walls 12" thick



(b) FREE BODY DIAGRAMS (SHEAR LOADS)

Figure 18

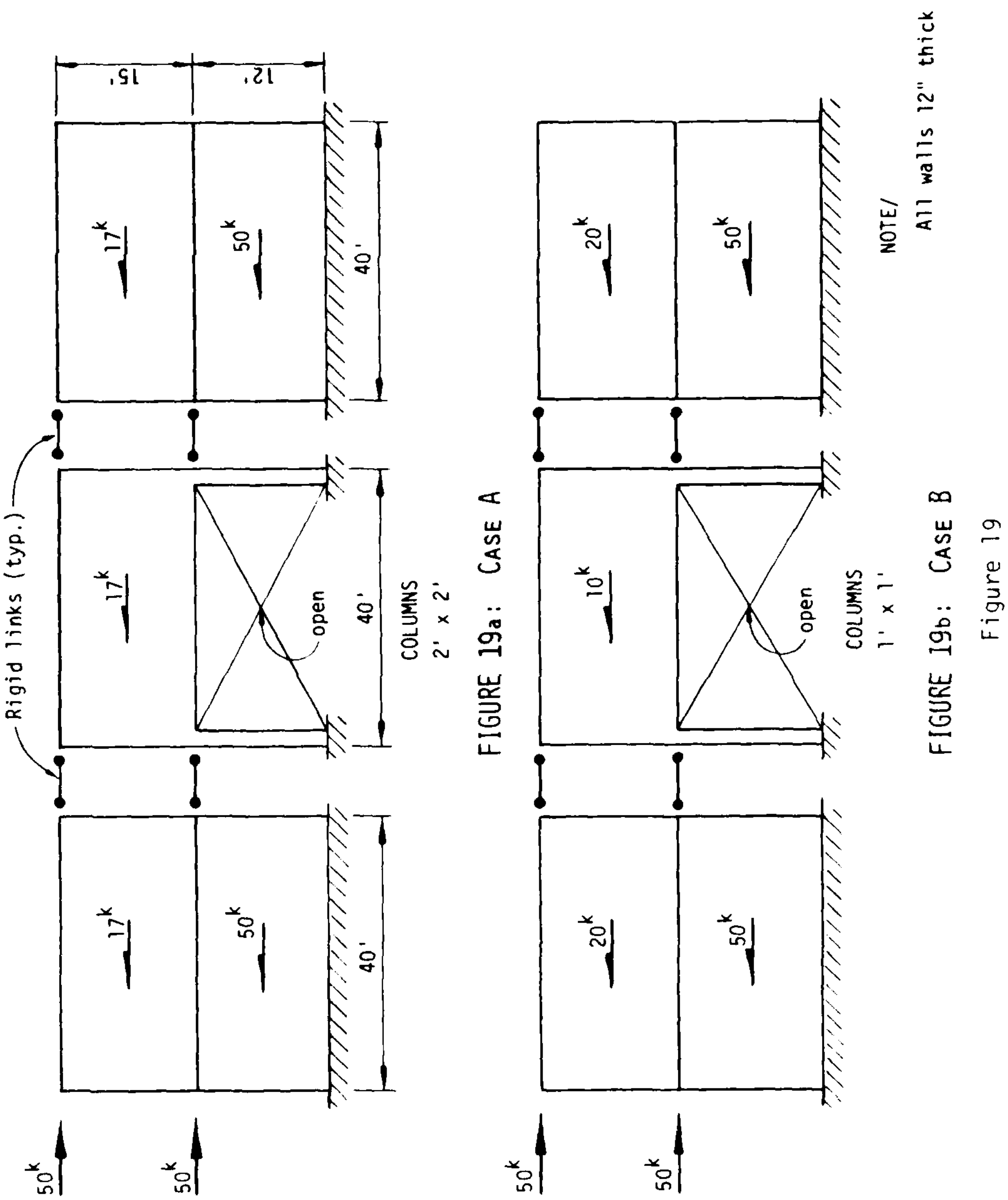


FIGURE 19a: CASE A

FIGURE 19b: CASE B

Figure 19

links at the story levels.

Consider the shear distribution in the upper level. In case A, a "correct" analysis indicates that the 50 K story shear is distributed approximately equally among the 3 walls. A conventional hand analysis would lead us to the same conclusion. In case B, however, the shear taken by the wall terminating on columns is 50% of that taken by the solid walls. The reason being that the smaller axial area of the columns gives larger axial deformations which, in turn, reduce the lateral rigidity of the wall in the story above.

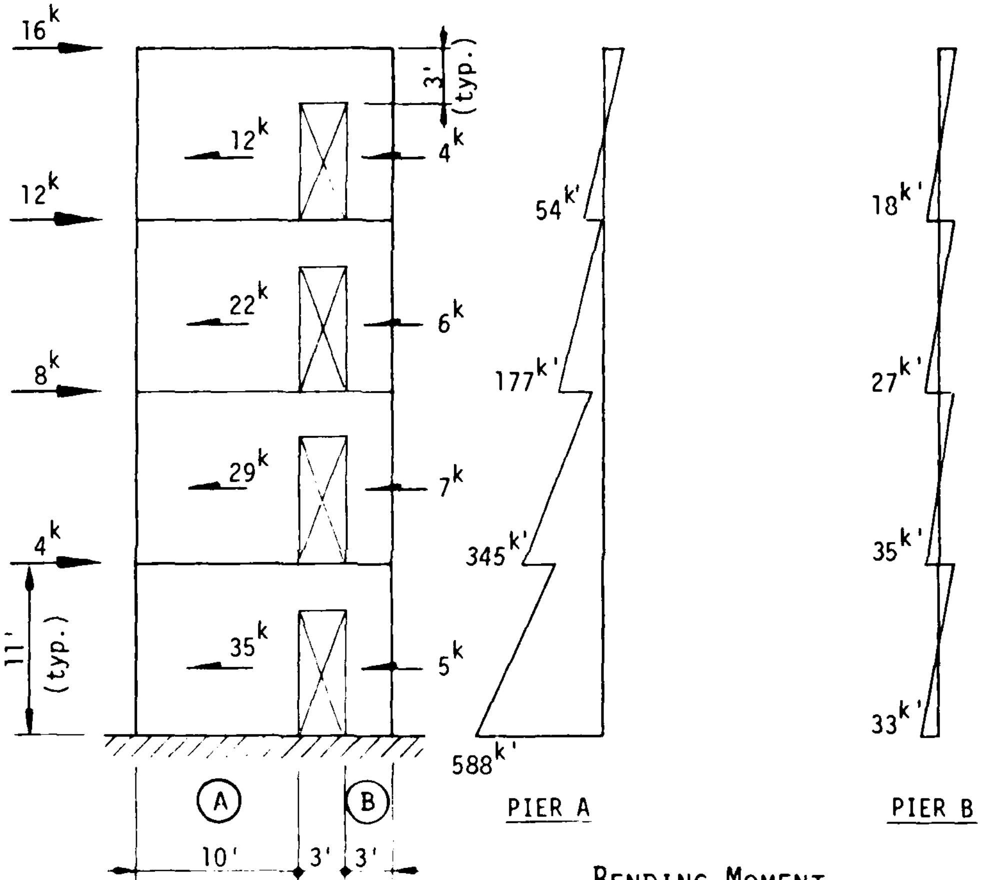
A conventional hand analysis neglects the effects of axial deformations, and, therefore, would give an equal shear distribution in all three walls regardless of the size of the columns.

(v). Example 5

This example clearly demonstrates the analytical discrepancy in analyzing an n-story structure as n 1-story structures. The structure is a shear wall with the same story height, wall dimensions and openings at every level. See Figure 20.

A conventional hand analysis (analysis of the structure as four 1-story structures) would indicate that the location of the point of contra-flexure of the piers in all the stories would be at a constant distance vertically from the corresponding diaphragm. Also, the ratio of the shear force in pier A to that in pier B in all stories will be constant.

A "correct" solution of the structure, however, shows that there is no point of contra-flexure in pier A in the first three stories. Also, the



NOTE/
Wall thickness = 12"

BENDING MOMENT
DIAGRAMS

Figure 20

percentage of the story shear carried by pier B is not constant in all stories but decreases as we move into the lower stories.

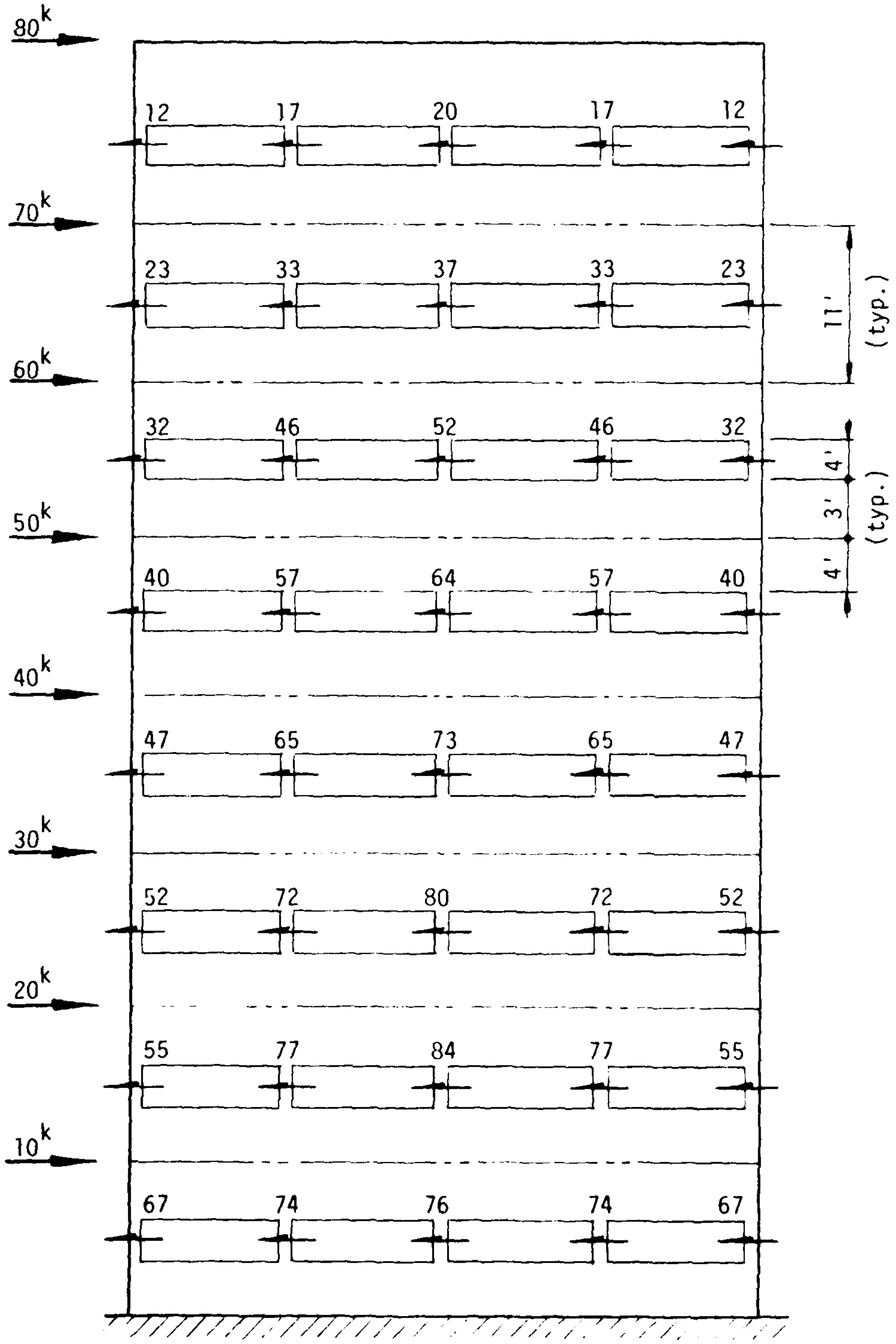
Joint rotation incompatibility from story to story is the main cause for this discrepancy between the hand analysis and the "correct" analysis.

(vi). Example 6

This is another example demonstrating the effect of axial deformations on the shear distribution. The structure consists of slender piers framing into relatively stiff spandrels and is 8 stories tall. See Figure 21. For all practical purposes, the piers may be considered fixed in rotation at both ends. Since all the piers are of the same size, a conventional hand analysis would indicate that the story shear is distributed equally among the 5 piers.

A "correct" analysis, as we can see, indicates that the piers closer to the center take a higher percentage of the story shear. In the top story for instance, the center pier takes over 65% higher shear than the end pier. This discrepancy is due to the fact that the "correct" analysis considers axial deformations in the piers, whereas the hand analysis does not.

This behavior may be explained by the following analogy. Consider the lateral displacements in a vertical cantilever with a rectangular cross section and lateral loading. If the shear deformations are negligible compared to the bending deformations in the cantilever, the distribution of the shear stress across the section is parabolic with the maximum at the center. However, if the deformation pattern is one of pure shear,



80k
12 17 20 17 12
70k
23 33 37 33 23
11' (typ.)
60k
32 46 52 46 32
4'
3' (typ.)
4'
50k
40 57 64 57 40
40k
47 65 73 65 47
30k
52 72 80 72 52
20k
55 77 84 77 55
10k
67 74 76 74 67

Figure 21

such that the bending deformations are negligible, the shear stress distribution is constant over the section and not parabolic.

The axial deformations of the piers in this example correspond to the bending deformations in the analogy above. If the piers were axially stiff to the extent that there were no axial deformations, the shear distributions in the piers would be equal. However, as the piers deform axially, the piers closer to the middle have heavier shear. This also explains why the shear distribution is more uniform as we move into the lower stories.

D. Dynamic Seismic Analysis of Buildings

The deficiencies of the present seismic design procedures are clearly summarized in Reference 8. It is apparent that the present code is a very approximate method based on the first mode only. The foundation factors discussed later are not considered. Another factor which is important in an elastic analysis is the damping. Spectra for damping of 2 and 10% are shown in Figure 22. It is clear that the Uniform Building Code seismic loads are very small compared to the forces produced in recorded earthquakes. It has been estimated that earthquakes of the Parkfield magnitude can be expected about once per year at some point in California, and earthquakes of the El Centro magnitude may be expected every five or six years.

The selection of a design spectrum for the response spectrum analysis of a particular building will depend on the geographical area, the local soil condition, the type of construction material and the intended use of the

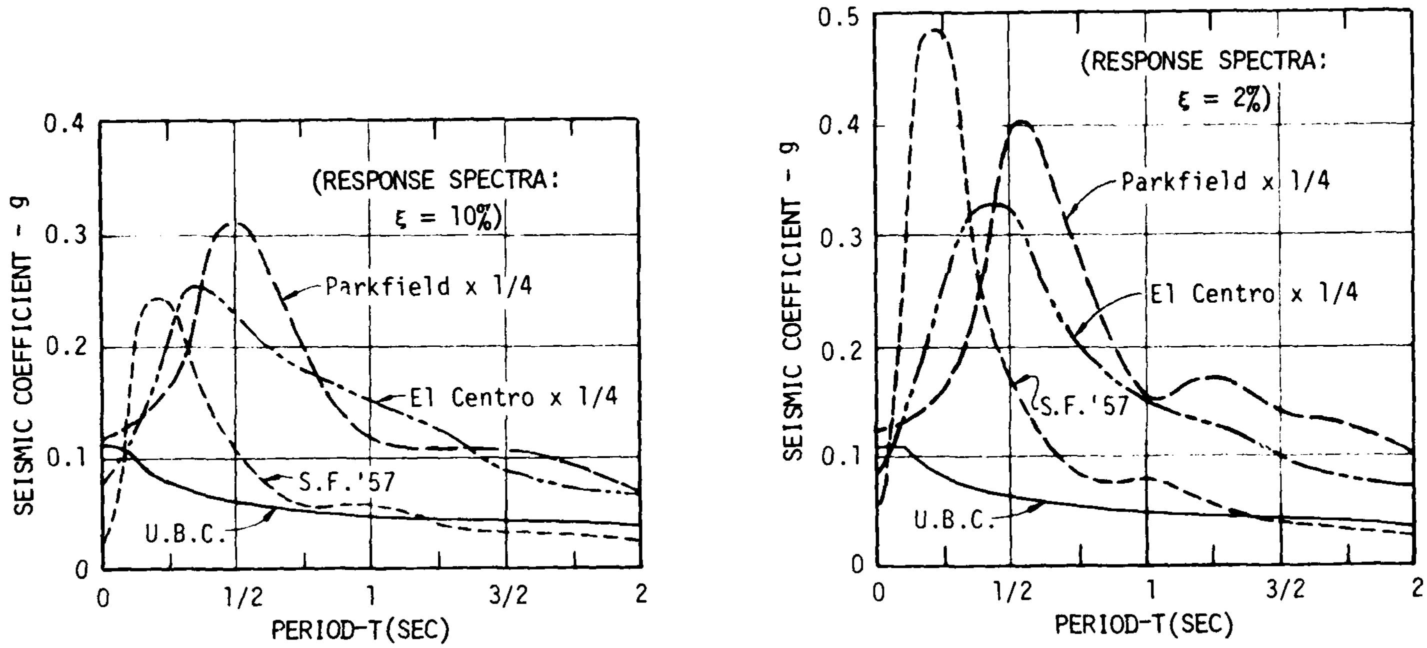


Figure 22. Response spectra

building. Many Soils Engineering firms now specialize in the dynamics of soil systems to evaluate specific sites and recommend shapes and intensities for dynamic response spectra. Most lending agencies are requiring dynamic analysis of structures as part of their financing terms for buildings in major metropolitan areas of California.

For certain types of earthquakes it has been observed that the vertical accelerations are comparable in magnitude to the lateral accelerations. However, all building have been designed elastically for a minimum of 1 g in the vertical direction; therefore, these additional vertical forces very often do not cause direct damage to the structure. Of course, they should be considered in the design of members in addition to the lateral earthquake loads. For most structures the stiffness in the vertical direction is very large; hence, the vertical periods will be very small. Therefore, a dynamic analysis in the vertical direction may not be required. A direct increase in dead load stresses may be a good method to approximate the effects of vertical earthquake loads.

E. Foundation - Building Interaction

Foundation modeling has always been an area of particular concern. The vertical and rotational stiffnesses under each column can be easily input by providing an extra "dummy" story. However, the assigning of accurate stiffness values for these soil springs can be difficult.

In recent years considerable research has been conducted in the area of foundation - building interaction. However, very little of this work has been of direct value to the profession involved in the earthquake analysis

of buildings. Several of the suggested approaches have been difficult to apply in case of complex buildings, or they have had serious theoretical restrictions.

Before foundation interaction effects are included in the analysis it is necessary to define the exact location of the earthquake input. If the design criteria states that the input is at the base of the building then it is impossible to say that the building will modify the input, and it is impossible to include interaction effects.

A large amount of research in this area has been associated with machines vibrating on an infinite foundation where the term radiation damping has been used. This work has little value in earthquake engineering since the energy source is not at the base of the building. It is easy to show that the energy stored in the building is very small compared to the energy stored in the immediate foundation area in the case of earthquake input. Also, the machine vibration problem is a steady state phenomenon, whereas earthquakes produce a transient loading.

The continuous foundation contains an infinite number of degrees of freedom. Therefore, any approach which suggest representing the lateral behavior of the foundation with a simple spring, dashpot and mass system is a very gross approximation. In fact, this technique can produce a filtering effect on the earthquake input and cause serious errors. For lateral earthquake input, this type of approximation is only acceptable in the representation of the rotational stiffness at the base of columns and shear walls.

The most significant factor to consider is the modification of the basic

earthquake rock motion by the layers of soil material under the building[7]. For certain earthquakes and locations this may be a factor of 2 or 3 in amplification. Therefore, it is very important that the dynamic behavior of the site is studied independently of the building. The results of such a study will result in a suggested acceleration spectrum to be used in the analysis of the building. Figure 23 indicates the type of results which can be expected from such a site analysis.

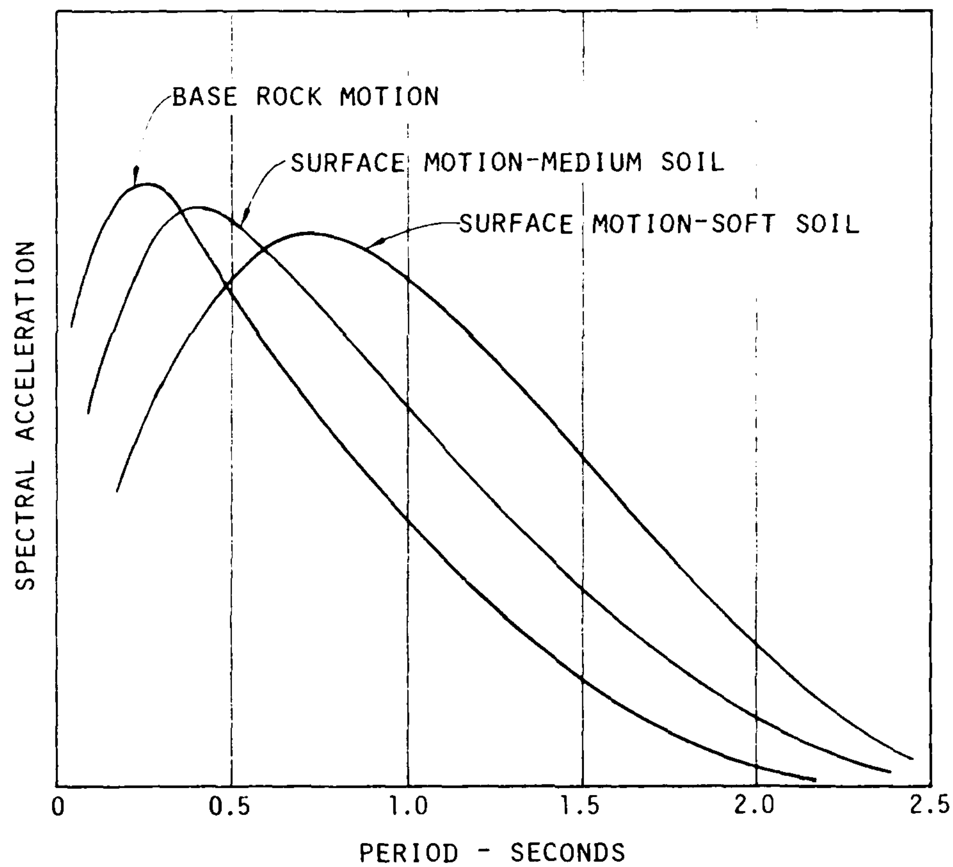


Figure 23. Spectral variation due to soil conditions

CHAPTER VI: INTERNAL ORGANIZATION

An outline of the subroutine structure of CTABS80 is presented in Figure 24. There are seven major calls from CTABS80 associated with the seven major blocks of the program.

1. The first operation is to read the basic control information. The data associated with the complete building (story data and structural lateral loads) is then rolled in via subroutine TABI.

2. The next operation involves reading in the frame data of every different frame in the structure. The frame elevations are plotted, if requested. In non-data check modes the frame stiffnesses are formulated and reduced and the frame lateral stiffness matrices and back-substitution equations are written sequentially on disc. This operation is implemented by the call to subroutine TABF.

3. The call to subroutine TABL reads the frame location data and formulates the complete lateral structural stiffness matrix of the whole building.

4. Subroutine SFRAME causes a plan view of the building to be plotted, showing the frame locations and the directions of their local axes.

5. The call to subroutine TABE gives the modeshapes and freouencies of the structure (TABM) and triggers the automatic UBC lateral seismic load calculation (TUBC). Also the dynamic analysis control information is read in by this call. Structural lateral displacements due to the static loads (TABQ) and response spectrum dynamic loads are obtained at this stage.

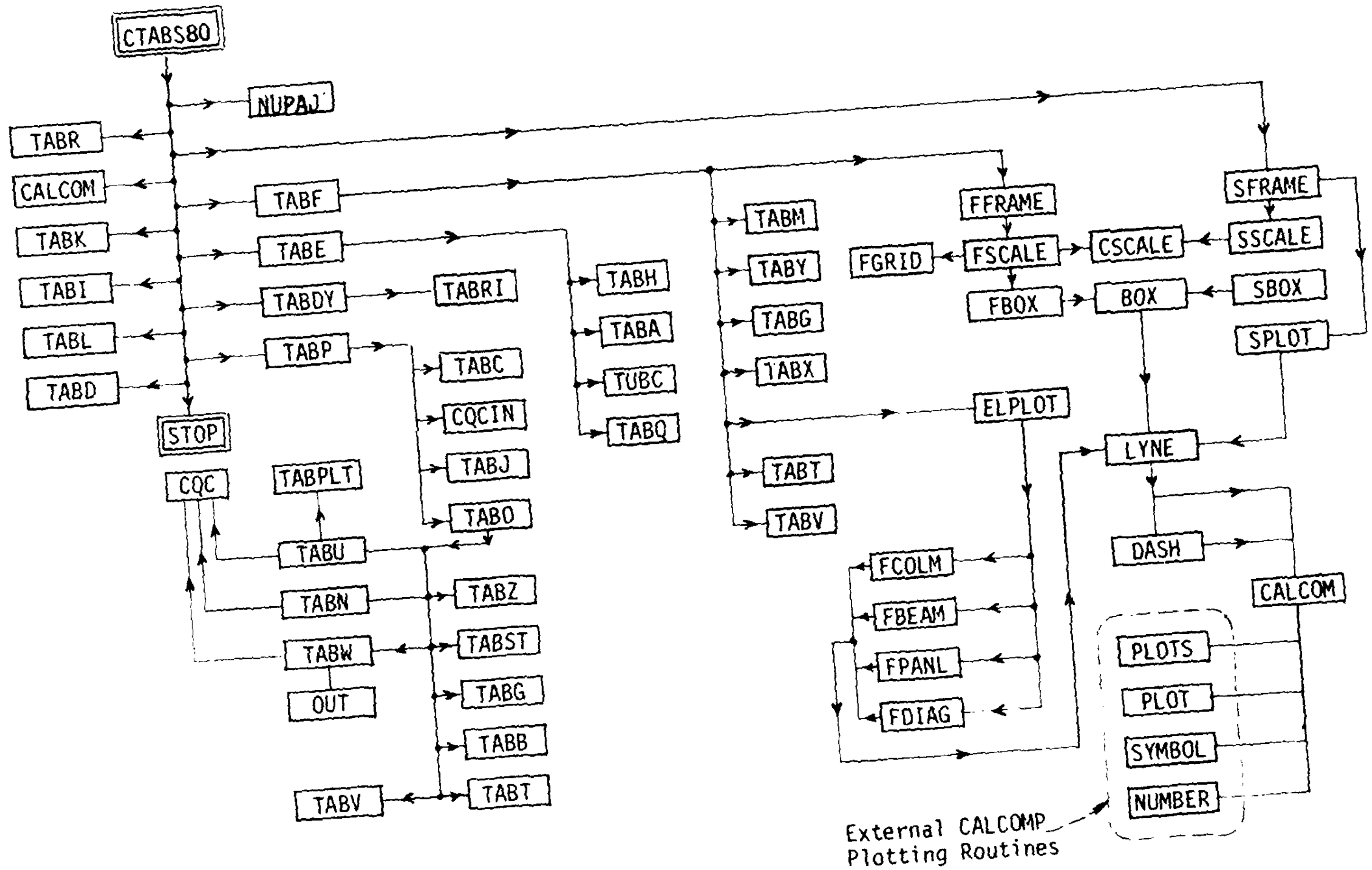


Figure 24. Internal organization of CTABS80

6. Subroutine TABDY reads in the time history earthquake ground motion data and causes the structural lateral displacements to be calculated for each time step.

7. Finally, subroutine TABP is called. This subroutine calls TABC to read the load case definition data for each frame. Then TABU is called to print the frame lateral displacements and TABO is called to calculate the frame joint displacements for each static load condition and each spectral mode or response time increment from the back substitution equations previously saved.

   As the displacements are calculated the member forces are also evaluated and printed according to the load case definition data (TABW).

CHAPTER VII: CONCLUSIONS

A general computer program for the elastic three-dimensional static and dynamic analysis of frame and shear wall buildings has been presented. For buildings which can be approximated by independent frames and shear walls the program is very economical and easy to use as compared to a general purpose three-dimensional structural analysis program.

Many new options have been implemented in this release to make the program a more practical and useful engineering tool.

The program is based on linear theory. Non-linear behavior such as P-Δ effects and material plasticity are not captured by the program.

If non-linear effects are to be considered a step-by-step response analysis is required; however, this involves a significant increase in computational effort and will be justified for only a limited number of buildings. In addition, the non-linear material properties both for most structural and non-structural members have not been established accurately from experimental work.

REFERENCES

1. Wilson, E.L., Habibullah, A.
"Three Dimensional Static and Dynamic Analysis of Multistory Frame and Shear Wall Buildings," NISEE, University of California, Berkeley, California, February 1977.

2. Wilson, E.L., Dovey, H.H.
"Three Dimensional Analysis of Building Systems - TABS," EERC, Report No. 72-8, University of California, Berkeley, California, December 1972.

3. Wilson, E.L., Habibullah, A.
"A Program for Three-Dimensional Static and Dynamic Analysis of Multistory Buildings," Structural Mechanics Software Series, Vol. II, University Press of Virginia, 1978.

4. Peterson, F.E., Blissmer, F.B., Kirby, R.W.
"LATERAL - A Computer Program for the Three Dimensional Analysis of Multistory Frame and Shear Wall Buildings," Engineering Analysis Corporation, Redondo Beach, California

5. Clough, R.W., King, I.P., Wilson, E.L.
"Structural Analysis of Multistory Buildings," ASCE Journal of the Structural Division, August 1963.

6. Clough, R.W., Bunuska, K.L., Wilson, E.L.
"Inelastic Earthquake Response of Tall Buildings," 3rd World Conference on Earthquake Engineering, New Zealand, January 1965.

7. Seed, H.B., Idriss, I.M.
"Influence of Local Soil Conditions on Building Damage Potential During Earthquake," Earthquake Engineering Research Center Report No. EERC 69-15, University of California, Berkeley, California, December 1969.

8. Clough, R.W.
"Deficiencies in Current Seismic Design Procedures," "Civil Engineering Frontiers in Environmental Technology," A Program

of Public Lectures to Commemorate the Dedication of Raymond E. Davis Hall, Department of Civil Engineering, University of California, Berkeley, California, October 1969.

9. Muto, Kiyoshi
"Seismic Analysis of Reinforced Concrete Buildings," 1st World Conference on Earthquake Engineering, Berkeley, California, 1956.

10. Wilson, E.L., Kiureghian, A.D., Bayo, E.
"A Replacement for the SRSS Method in Seismic Analysis," University of California, Berkeley, California, January 1980.

11. Blissmer, F.B., Kirby, R.W., Peterson, F.E.
"EASE2 - Elastic Analysis for Structural Engineering," Engineering/Analysis Corporation, Redondo Beach, California, 1979.

12. Bathe, K.J., Wilson, E.L., Peterson, F.E.
"SAPIV - A Structural Analysis Program for Static and Dynamic Response of Linear Systems," EERC Report No.73-11, June 1973, Revised April 1974.

13. Clough, R.W.
"Earthquake Analysis by Response Spectrum Superposition," Bulletin of the Seismological Society of America, Vol. 52, No. 3, July 1962.

14. International Conference of Building Officials
"Uniform Building Code," Whittier, California, 1979.

15. Wilson, E.L., Hollings, J.P., Dovey, H.H.
"ETABS, Three Dimensional Analysis of Building Systems (Extended Version)," EERC Report No. 75-13, University of California, Berkeley, California, April 1975.

16. Nathan M. Newmark, Consulting Engineering Services
"A Study of Vertical and Horizontal Earthquake Spectra," U.S. Government Printing Press Office, Washington, D.C., April 1973.

17. Habibullah, A., Peterson, F.E.
" E2SPEC-A Time History Post Processor for EASE2," Engineering/Analysis Corporation, Redondo Beach, California, 1979.

18. Wilson, E.L., Dovey, H.H., Habibullah, A.
"User's Guide: Computer Program for Three-Dimensional Analysis of Building Systems (CTABS80)," Instruction Report K-81-9, 1981, U. S. Army Engineer Waterways Experiment Station, Vicksburg, Mississippi.

APPENDIX A: FORTRAN IV LISTING OF CTABS80

```
      PROGRAM    TABS80 (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,NPFILE,
     *                  TAPE1,TAPE2,TAPE3,TAPE4,TAPE8,TAPE9,TAPE15,
     *                  TAPE20=NPFILE)

      A GENERAL PROGRAM FOR THE STATIC AND DYNAMIC ANALYSIS OF FRAME
      AND SHEAR WALL THREE-DIMENSIONAL BUILDINGS--- E WILSON AND H DOVEY
      8 MARCH 1972---UCB.

--------------------------------------------------------------------


      T A B S   8 0     - - -   THE 1980 MODEL OF TABS


      DEVELOPED BY

      ASHRAF HABIBULLAH
      COMPUTERS AND STRUCTURES INC
      4009 WEBSTER STREET OAKLAND CALIFORNIA 94609
      TELEPHONE (415) 655-9151


      RELEASED JUNE 1980

      REVISED MARCH 1981


      DEVELOPMENT SPONSORED BY

      DEPARTMENT OF THE ARMY
      U. S. ARMY WATERWAYS EXPERIMENT STATION
      VICKSBURG MISSISSIPPI 39181

      DR. N. RADHAKRISHNAN
      DANIEL W. REYNOLDS


--------------------------------------------------------------------

      COMMON /GEN1 / NST,NDF,NTF,NLD,NAT,NFO,NSD,NOPT,NRGD,NDSP,NUBC,
     *               NPLT,NSS,IJ,IS,NAL
      COMMON /GEN2 / RLAB(3),ANI,ANP,RIGID
      COMMON /FTAPE/ KTMP,KINP,KOUT,KSTR,NTAPE,KTAPE
      COMMON /IHEDR/ IHED(28),IDATE(2),NPAGE(2),MAXLIN,LINE
      COMMON /IERR / IERROR
      COMMON /KOPT / KINP,NOUT
      COMMON /UBC  / J,IS,UBC1,GRAV,PERIOD(2),UBCK(2),NTOP(2),NBOT(2)
     *       /PLOTS/ X0,Y0,XDPX,XDMN,XDIM,YDIM,S7,B1,B2,D,RAD,FHED(14)
      COMMON /DYN  / NTIPE,DT,NPC,DAMP,NMD,IHTYP,HDT
      COMMON /JUNK / JNK(250)
C
      DIMENSION IBUF(1500)
      DIMENSION T(6)
C
C     FOLLOWING TWO CARDS CONTROL PROGRAM CAPACITY . . . .
C
      COMMON  A (8000)
      DATA MTOT /8000/
C
      DATA RLAB /4H  X ,4H  Y ,4HROTN /
      DATA KTMP/5/,KINP/8/,KOUT/6/,KSTR/9/,NTAPE/4/,MTAPE/20/,KTAPE/15/
      DATA IERROR/0/
C
      DATA MAXLIN/56/
      DATA NPAGE/0,0/
      DATA LINE/39/
C
      DATA XDPN/10./
      DATA XDMX/16./
      DATA YDIP/10./
      DATA S7/.1/
      DATA B1/.75/
      DATA B2/.5/
C
      B=1500.
      X0=0.
      Y0=0.
      CALL CALCOM (B,B,B,IBUF,B,B,MTAPE,1)
      CALL CALCOM (12.,.25,B,B,B,B,-3,2)
C
      RAD=4.*ATAN(1.0)/180.
C
C     RECOVERING RUN DATE FROM SYSTEM
C
      CALL DATE (IDATE(1),IDATE(2))
C
      REWIND KSTR
C
C     REMOVING COMMENT CARDS FROM INPUT STREAM  ($-SIGN IN COLUMN 1)
C
      CALL TARK (KTMP,KINP)
C
C     READ AND PRINT JOB CONTROL INFORMATION
C
  100 READ  (KINP,1000) IHED,NST,NDF,NTF,NLD,NAT,NFO,NSD,NOPT,NRGD,NDSP,
     *                  NUBC,NPAT,NPLT,NAL,NKEY
C
      IF (NST.GT.0) GO TO 150
      CALL CALCOM (0.,0.,B,B,B,B,999,2)
      IF (IERROR.GT.0) CALL TARP (0,0)
```

```
      STOP 77777
  150 CONTINUE
C
      CALL SECOND (TIII)
C
C     SETTING RIGID ZONE REDUCTION FACTOR
C
      RIGID=.25
      IF (NRGD.NE.0) RIGID=0.
C
      NINP=1
      NOUT=1
      IF (NOPT.EQ.3) NINP=0
      IF (NOPT.EQ.2) NINP=0
      IF (NOPT.EQ.2) NOUT=0
C
      I3=3
      IS=0
      IF(NSD.EQ.0) GO TO 50
      I3=1
      IS=NSD-1
   50 NSS=NST*I3
      IF(NFQ.GT.NSS) NFQ=NSS
      IF(NAT.EQ.0) NFQ=0
C
      CALL NUPAJ (1,27,ITEST)
      WRITE (KOUT,2000) NST,NDF,NTF,NLD,NAT,NFQ,NSD,NOPT,NRGD,NDSP,
     *                  NURC,NPAT,NPLT
C
      IF (NAL.EQ.1) NAL=3
      IF (NAL.EQ.0) NAL=1
C
      REWIND KTAPE
      WRITE (KTAPE) IHED,IDATE
      WRITE (KTAPE) NST,NDF,NTF,NFFY
C
C     STRESS TRANSFORMATION DATA
C
      READ  (KINP,1002) ANI,ANP
      IF (ANI.LE..0) ANI=1.0
      IF (ANP.LE..0) ANP=1.0
      IF (NINP.EQ.1)
     *WRITE (KOUT,1004) ANI,ANP
C
      REWIND 1
      REWIND 2
      REWIND 3
C
C     READ AND PRINT OF STORY INFORMATION
C
      N0=1
      N1=N0+NST*NDF
      N2=N1+15*NST
      N3=N2+2*NST
      N4=N3+2*NST
      N5=N4+NPAT
      N6=N5+NPAT
      N7=N6+NPAT
      N8=N7+NPAT
      N9=N8+NST
      IF (N9.GT.MTOT) CALL TARP ((N9-MTOT),1)
      CALL TABI (A(N1),A(N2),A(N3),A(N4),A(N5),A(N6),A(N7),A(N8),
     *           NST,NPAT)
C
C     READ AND PRINT OF FRAME PROPERTIES AND FORM LATERAL STIFFNESSES
C
      DO 200 I=1,NDF
C
      READ  (KINP,1001) FHED,M,NC,NS,NCP,NBP,NFEF,MCONL,NPAN,NDIG,IPLT,
     *                  NSCC
C
      IF (NPLT.EQ.0) IPLT=0
      IF (NPLT.EQ.3) IPLT=0
      IF (NSCC.NE.0) NSCC=1
      IF (NINP.EQ.0) GO TO 250
      CALL SECOND (TT)
      LINE=99
      CALL NUPAJ (1,14,ITEST)
      WRITE (KOUT,2001) FHED,M,NC,NS,NCP,NBP,NFEF,MCONL,NPAN,NDIG,IPLT,
     *                  NSCC
C
  250 NB=NC-1
      NCP=NCP+1
      NBP=NBP+1
      NN=4*NC*NS+1
      N2=N1+19*NST
      N3=N2+NB
      N4=N3+8*NCP
      N5=N4+8*NBP
      N6=N5+7*NFEF
      N7=N6+NB*NS
      N8=N7+NS*NB*4
      N9=N8+NS*NC
      N10=N9+2*NPAN
      N11=N10+2*NPAN
      N12=N11+3*NDIG
      N13=N12+NDIG
      N14=N13+NFEF
      N15=N14+NFEF*MCONL*2
      N16=N15+NB
```

```
      N17=N16+NN*NN
      N18=N17+4*NN
      N19=N18+NC
      N20=N19+NS
      IF(N20.GT.MTOT) CALL TABR((N20-MTOT),1)
C
      CALL TABF
     *    (A(N0),A(N1),A(N2),A(N3),A(N4),A(N5),A(N6),A(N7),A(N8),A(N9),
     *          A(N10),A(N11),A(N12),A(N13),A(N14),A(N15),A(N16),
     *          A(N17),A(N18),A(N19),
     *          NST,NC,NB,NS,NCP,N9P,NFEF,MCONL,NPAN,NDIG,NN,I,NOPT,
     *          IPLT,NAL,NSCC)
C
      IF (NINP.EQ.0) GO TO 200
      CALL NUPAJ (1,5,ITEST)
      CALL SECOND (TTT)
      TTT=TTT-TT
      WRITE (KOUT,2003) I,TTT
C
  200 CONTINUE
C
      NLOC=N0+NST*(NDF+2)-1
      WRITE (KTAPE) (A(I),I=N0,NLOC)
C
      CALL SECOND (T(2))
C
C     ASSEMBLE STRUCTURAL STIFFNESS AND LATERAL LOAD VECTORS
C
      N3=N2+NSS*NSS
      N4=N3+6*NSS
      N5=N4+((NST+1)*NST)/2
      N6=N5+4*NST
      N7=N6+NST*NTF
      N8=N7+5*NTF
      N9=N8+4*NTF
      IF(N9.GT.MTOT) CALL TABR((N9-MTOT),1)
      CALL TABL
     *    (A(N0),A(N1),A(N2),A(N3),A(N4),A(N5),A(N6),A(N7),NSS,NST,NTF,
     1  A(N8))
      CALL SECOND (T(3))
C
C     PLOTTING PLAN VIEW / FRAME LOCATIONS
C
      IF (NPLT.EQ.0) GO TO 550
      IF (NPLT.EQ.2) GO TO 550
      CALL SFRAME
  550 CONTINUE
C
C     UBC LATERAL LOAD DATA
C
```

```
      IF (NUBC.EQ.0) GO TO 600
      READ  (KINP,1003) Z,TS,UBCI,GRAV
      READ  (KINP,1005) (NTOP(I),NBOT(I),PERIOD(I),UBCK(I),I=1,2)
      IF (Z.LE..0) Z=1.0
      IF (UBCI.LE..0) UBCI=1.0
      IF (GRAV.LE..0) GRAV=32.2
      DO 800 I=1,2
      IF (NTOP(I).EQ.0) NTOP(I)=NST
      IF (NTOP(I).GT.NST) NTOP(I)=NST
      IF (NBOT(I).LE.0) NBOT(I)=0
      IF (NBOT(I).GE.NST) NBOT(I)=0
      IF (UBCK(I).LE..0) UBCK(I)=0.67
  800 CONTINUE
C
C     LATERAL STATICS AND DYNAMICS
C
  600 N3=N2+NSS*MAX0(NFQ+6,NSS)
      N4=N3+NSS
      N5=N4+NSS
      N6=N5+NSS
      N7=N6+NSS*NSS
      N8=N7+400
      IF(N8.GT.MTOT) CALL TABR((N8-MTOT),1)
      CALL TABE (A(N1),A(N2),A(N3),A(N4),A(N5),A(N6),A(N7),NSS,NST)
C
C     TIME HISTORY RESPONSE
C
      IF (NAT.NE.4) GO TO 350
      MLD=6+NTIME
      N3=N2+MLD*NSS
      N4=N3+2*NPC
      N5=N4+NTIME
      N6=N5+NSS
      IF (N6.GT.MTOT) CALL TABR ((N6-MTOT),1)
      CALL TABDY (A(N2),A(N3),A(N4),A(N5),NSS)
  350 CALL SECOND (T(4))
C
C     COMPUTE FRAME DISPLACEMENTS
C
      IF(NAT.LE.2) NFQ=0
      IF (NAT.EQ.1) GO TO 300
      MLD=6+NFQ
      IF (NAT.EQ.4) MLD=6+NTIME
      N3=N2+MLD*NSS
      N4=N3+NST*NTF
      N5=N4+NST*MLD
      N6=N5+5*NTF
      N7=N6+NTF*4
      IF(N7.GT.MTOT) CALL TABR((N7-MTOT),1)
      CALL TABD
```

```
     *     (A(N0),A(N1),A(N2),A(N3),A(N4),A(N5),NSS,NTF,MLD,NST,A(N6))
 300 CALL SECOND (T(5))

     CALCULATE AND PRINT DISPLACEMENTS AND STRESSES

     IF (NAT.EQ.1) GO TO 400
     CALL TABP (MTOT)
 400 CALL SECOND (T(6))
     TT=T(6)-T(1)
     DO 500 I=1,5
 500 T(I)=T(I+1)-T(I)
     T(6)=TT

     LINE=99
     CALL NUPAJ (2,11,ITEST)
     WRITE (KOUT,2002) T
     WRITE (KSTR,2002) T
     LINE=99

     GO TO 100


1000 FORMAT (14A5/14A5/14I5,A5)
1001 FORMAT (14A5/14I5)
1002 FORMAT (2F10.0)
1003 FORMAT (4F10.0)
1004 FORMAT (///42H CONVERSION DATA FOR STRESSES                //
    *          45H LENGTH CONVERSION FACTOR----------------------------F10.3/
    *          45H FORCE  CONVERSION FACTOR----------------------------F10.3)
1005 FORMAT (2I5,2F10.0)
2000 FORMAT (//
    *          45H TOTAL NUMBER OF STORIES IN STRUCTURE--------I10//
    *          45H NUMBER OF DIFFERENT FRAMES IN STRUCTURE-----I10//
    *          45H TOTAL NUMBER OF FRAMES IN STRUCTURE---------I10//
    *          45H TOTAL NUMBER OF STRUCTURAL LOAD CASES-------I10//
    *          45H TYPE OF ANALYSIS----------------------------I10//
    *          45H NUMBER OF MODES CONSIDERED------------------I10//
    *          45H LATERAL STORY TRANSLATION CODE--------------I10//
    *          45H EXECUTION MODE------------------------------I10//
    *          45H FRAME JOINT RIGID ZONE MODIFICATION CODE----I10//
    *          45H FRAME JOINT DISPLACEMENT PRINT FLAG---------I10//
    *          45H UBC LATERAL SEISMIC FORCE CODE--------------I10//
    *          45H NUMBER OF STORY MASS PATTERNS---------------I10//
    *          45H MASTER PEN PLOT FLAG------------------------I10//
    *                                                         //)
2001 FORMAT (//1H ,14A5//
    *          45H FRAME IDENTIFICATION NUMBER-----------------I10/
    *          45H NUMBER OF COLUMN LINES IN FRAME-------------I10/
    *          45H NUMBER OF STORY LEVELS IN FRAME-------------I10/
    *          45H NUMBER OF COLUMN/PANEL/DIAGONAL PROPERTIES--I10/
    *          45H NUMBER OF BEAM PROPERTIES-------------------I10/
    *          45H NUMBER OF BEAM SPAN LOADING PATTERNS--------I10/
    *          45H MAXIMUM POINT LOADS IN ANY SPAN LOADING-----I10/
    *          45H NUMBER OF PANEL ELEMENTS IN FRAME-----------I10/
    *          45H NUMBER OF DIAGONAL ELEMENTS IN FRAME--------I10/
    *          45H FRAME PEN PLOT FLAG-------------------------I10/
    *          45H STORY CONNECTIVITY CODE---------------------I10/
    *                                                          )
2002 FORMAT (//19H TIME LOG (SECONDS) //
    1  41H FORM FRAME STIFFNESSES................. = F7.2/
    2  41H SOLVE STATIC LOAD CASES................ = F7.2/
    3  41H MODE SHAPES AND FREQUENCIES............ = F7.2/
    4  41H COMPUTE FRAME DISPLACEMENTS............ = F7.2/
    5  41H MEMBER FORCES AND STRESSES............. = F7.2//
    6  41H TOTAL TIME............................. = F7.2)
2003 FORMAT (//11H FRAME NO.= I6/11H TIME      = F6.2)
     END


     SUBROUTINE NUPAJ (ITAG,NLINE,ITEST)
C
C    THIS SUCKER IS A PAGE EJECTOR
C
     COMMON /FTAPE/ KTMP,KINP,KOUT,KSTR
     COMMON /IHEDR/ IHED1(14),IHED2(14),IDATE(2),NPAGE(2),MAXLIN,LINE
C
     ITEST=0
     LINE=LINE+NLINE
     IF (LINE.LE.MAXLIN) GO TO 999
C
     ITEST=1
     LINE=NLINE+2
     NPAGE(1)=NPAGE(1)+1
     WRITE (KOUT,1000) IHED1,NPAGE(1),IHED2,IDATE(1)
C
     IF (ITAG.EQ.1) GO TO 999
     NPAGE(2)=NPAGE(2)+1
     WRITE (KSTR,1000) IHED1,NPAGE(2),IHED2,IDATE(1)
C
 999 RETURN
C
1000 FORMAT (1H1,14A5,4HPAGE,I5/
    *          1H ,14A5,A9)
     END
```

```
      SUBROUTINE TABK (KTMP,KINP)

      REMOVING S-COMMENT CARDS FROM INPUT STREAM

      COMMON /JUNK/ IDLR,ICARD(18)

      DATA ISTOP1 /2H  /
      DATA ISTOP2 /2HNO/
      DATA IIEST /2HS /

      REWIND KINP

   10 READ  (KTMP,1000) IDLR,ICARD
      IF (IDLR.EQ.ITEST) GO TO 10
      IF (IDLR.EQ.ISTOP1.AND.ICARD(1).EQ.ISTOP2) GO TO 20
      WRITE (KINP,1000) IDLR,ICARD
      GO TO 10

   20 IDLR=2H
      DO 30 I=1,18
   30 ICARD(I)=4H
      DO 40 I=1,3
   40 WRITE (KINP,1000) IDLR,ICARD

      ENDFILE KINP
      REWIND KINP
      RETURN

 1000 FORMAT (A1,17A4,A6)
      END


      SUBROUTINE TABR (N,ITAG)

      ERROR MESSAGE SUBROUTINE

      COMMON /GEN1 / NST,NOF,NTF,NLO,NAT,NFO,NSD,NOPT
      COMMON /FTAPE/ KTMP,KINP,KOUT,KSTR,NTAPE
      COMMON /IERR / IERROR
      COMMON /IHEDR/ IHED(29),IDATE(2),NPAGE(2),MAXLIN,LINE

      CALL NUPAJ (1,3,ITEST)

      IF (ITAG.EQ.0) GO TO 999
      IF (IERROR.GT.50) GO TO 999

      IERROR=IERROR+1
      NOPT=1
      WRITE (KOUT,1000)
C
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,
     *       23,24,25,26,27,28,29,30,31),ITAG
C
    1 WRITE (KOUT,1001) N
      GO TO 999
C
    2 WRITE (KOUT,1002) N
      GO TO 99
C
    3 WRITE (KOUT,1003)
      GO TO 99
C
    4 WRITE (KOUT,1004)
      GO TO 99
C
    5 WRITE (KOUT,1005)
      GO TO 99
C
    6 WRITE (KOUT,1006) N
      LINE=LINE+1
      GO TO 99
C
    7 WRITE (KOUT,1007) N
      GO TO 99
C
    8 WRITE (KOUT,1008) N
      GO TO 99
C
    9 WRITE (KOUT,1009) N
      LINE=LINE+1
      GO TO 99
C
   10 WRITE (KOUT,1010) N
      GO TO 99
C
   11 WRITE (KOUT,1011) N
      GO TO 99
C
   12 WRITE (KOUT,1012) N
      GO TO 99
C
   13 WRITE (KOUT,1013) N
      GO TO 99
C
   14 WRITE (KOUT,1014) N
      GO TO 99
C
   15 WRITE (KOUT,1015) N
```

```
      GO TO 99
C
   16 WRITE (KOUT,1016) N
      GO TO 99
C
   17 WRITE (KOUT,1017) N
      LINE=LINE+1
      GO TO 99
C
   18 WRITE (KOUT,1018) N
      LINE=LINE+1
      GO TO 99
C
   19 WRIT[illegible],1019) N
      LIN[illegible]
      GO T[illegible]
C
   20 WRITE (KOUT,1020) N
      LINE=LINE+1
      GO TO 99
C
   21 WRITE (KOUT,1021) N
      GO TO 99
C
   22 WRITE (KOUT,1022)
      GO TO 999
C
   23 WRITE (KOUT,1023) N
      LINE=LINE+1
      GO TO 999
C
   24 WRITE (KOUT,1024) N
      GO TO 999
C
   25 WRITE (KOUT,1025) N
      GO TO 999
C
   26 WRITE (KOUT,1026)
      GO TO 999
C
   27 WRITE (KOUT,1027) N
      GO TO 999
C
   28 WRITE (KOUT,1028) N
      GO TO 999
C
   29 WRITE (KOUT,1029) N
      LINE=LINE+1
      GO TO 999
C
   30 WRITE (KOUT,1030)
      LINE=LINE+1
      GO TO 999
C
   31 WRITE (KOUT,1031) N
      LINE=LINE+1
      GO TO 99
C
C
   99 WRITE (KOUT,2000)
      LINE=LINE+1
      RETURN
C
C
  999 WRITE (KOUT,9999)
      STOP 12345
C
C
 1000 FORMAT (45H0* * * E R R O R * * *                         )
 1001 FORMAT (45H INCREASE BLANK COMMON LENGTH (STORAGE) BY   I10)
 1002 FORMAT (45H STORY MASS TYPE IS OUT OF RANGE FOR LEVEL   I10)
 1003 FORMAT (50H COLUMN PROPERTY TYPE IS OUT OF RANGE OR SEQUENCE   )
 1004 FORMAT (50H BEAM PROPERTY TYPE IS OUT OF RANGE OR SEQUENCE     )
 1005 FORMAT (50H BEAM LOADING TYPE IS OUT OF RANGE OR SEQUENCE      )
 1006 FORMAT (45H NUMBER OF CONCENTRATED LOADS IS OUT OF RANGE/
     *           16H IN SPAN LOADING I10)
 1007 FORMAT (45H BEAM PROPERTY REFERENCE OUT OF RANGE IN BAY I10)
 1008 FORMAT (45H BEAM LOADING  REFERENCE OUT OF RANGE IN BAY I10)
 1009 FORMAT (45H CONCENTRATED LOADS ARE NOT INPUT IN SEQUENCE/
     *           45H FROM LEFT TO RIGHT IN BEAM LOADING NUMBER   I10)
 1010 FORMAT (50H COLUMN PROPERTY REFERENCE OUT OF RANGE IN COLUMN I10)
 1011 FORMAT (45H PANEL LEVEL NUMBER OUT OF RANGE IN PANEL     I10)
 1012 FORMAT (45H PANEL BAY NUMBER OUT OF RANGE IN PANEL       I10)
 1013 FORMAT (45H PANEL PROPERTY NUMBER OUT OF RANGE IN PANEL    I10)
 1014 FORMAT (45H DIAGONAL LEVEL NUMBER OUT OF RANGE IN DIAG      I10)
 1015 FORMAT (45H DIAGONAL COLUMN NUMBER OUT OF RANGE IN DIAG     I10)
 1016 FORMAT (50H DIAGONAL PROPERTY NUMBER OUT OF RANGE IN DIAG     I10)
 1017 FORMAT (45H CLEAR COLUMN HEIGHT IS NOT GREATER THAN ZERO/
     *           17H IN COLUMN NUMBER I10)
 1018 FORMAT (45H CLEAR BEAM LENGTH IS NOT GREATER THAN ZERO  /
     *           15H IN BAY NUMBER I10)
 1019 FORMAT (54H STARTING BAY NUMBER IS GREATER THAN ENDING BAY NUMBER/
     *           10H IN PANEL I10)
 1020 FORMAT (45H UPPER COLUMN IS THE SAME AS THE LOWER COLUMN/
     *           19H IN DIAGONAL NUMBER I10)
 1021 FORMAT (45H CONCENTRATED LOAD LIES ON RIGID LINK IN BAY I10)
 1022 FORMAT (45H FRAME TYPE NUMBER IS OUT OF SEQUENCE/RANGE  )
 1023 FORMAT (45H FRAME LOCATION POINTS 1 AND 2 HAVE IDENTICAL/
     *           30H COORDINATES FOR FRAME NUMBER I10)
 1024 FORMAT (45H MASS NOT GREATER THAN ZERO AT LEVEL          I10)
```

```
 1025 FORMAT (45H MMI  NOT GREATER THAN ZERO AT LEVEL             I10)
 1026 FORMAT (50H NUMBER OF SPECTRUM PERIOD CARDS SHOULD BE LT  200)
 1027 FORMAT (45H DAMPING RATIO IS GREATER THAN 1 FOR MODE       I10)
 1028 FORMAT (45H DAMPING RATIO IS LESS THAN ZERO FOR MODE       I10)
 1029 FORMAT (45H INCONSISTENT ACCELERATION DATA AT POINT        I10/
     *        45H TIME (OR PERIOD) MUST INCREASE SEQUENTIALLY )
 1030 FORMAT (45H ACCELERATION HISTORY TIME SPAN IS LESS THAN /
     *        45H REQUESTED STRUCTURAL RESPONSE TIME           )
 1031 FORMAT (47H STRUCTURAL LEVEL NUMBER FOR FRAME CONNECTIVITY
     *        45H OUT OF RANGE OR SEQUENCE AT LEVEL NUMBER     I10)
 2000 FORMAT (1H )
 9999 FORMAT (45H0EXCESSIVE ERRORS OR IMMEDIATELY FATAL ERROR /
     *        45H EXECUTION TERMINATED                         )
      END


      SUBROUTINE TABI (SD,PA,PB,XM,XI,XX,YY,IMST,NST,NPAT)

      READ AND PRINT STORY DATA

      DIMENSION SD(NST,15),PA(NST,2),PB(NST,2),IPST(NST)
      DIMENSION XM(NPAT),XI(NPAT),XX(NPAT),YY(NPAT)

      COMMON /IHEDR/ IHED(28),IDATE(2),NPAGE(2),MAXLIN,LINE
      COMMON /FTAPE/ KTMP,KINP,KOUT,KSTR,NTAPE,KTAPE
      COMMON /KOPT / NINP,NOUT

      IF (NPAT.LE.0) GO TO 900

      DO 775 NT=1,NPAT
      XM(NT)=0.
      XX(NT)=0.
      YY(NT)=0.
      XI(NT)=0.

      READ  (KINP,1001) M,NSEG,SF
      IF (NSEG.LE.0) NSEG=1
      IF (SF.LE..0) SF=1.

      IF (NINP.EQ.0) GO TO 650
      LINE=99
      CALL NUPAJ (1,9,ITEST)
      WRITE (KOUT,8000) NT,NSEG,SF
      WRITE (KOUT,2000)

  650 DO 700 NG=1,NSEG
      READ  (KINP,1002) AM,XC,YC,XD,YD
      IF (NINP.EQ.0) GO TO 675
      CALL NUPAJ (1,1,ITEST)
      IF (ITEST.EQ.0) GO TO 655
      WRITE (KOUT,2000)
      LINE=LINE+4
  655 WRITE (KOUT,2001) NG,AM,XC,YC,XD,YD
  675 AM=AM*SF
      XM(NT)=XM(NT)+AM
      XI(NT)=XI(NT)+AM*((XC*XC+YC*YC)+(XD*XD+YD*YD)/12.)
      XX(NT)=XX(NT)+AM*XC
      YY(NT)=YY(NT)+AM*YC
  700 CONTINUE
C
      XI(NT)=XI(NT)-(XX(NT)*XX(NT)+YY(NT)*YY(NT))/XM(NT)
      XX(NT)=XX(NT)/XM(NT)
      YY(NT)=YY(NT)/XM(NT)
C
      IF (NINP.EQ.0) GO TO 775
      CALL NUPAJ (1,5,ITEST)
  750 WRITE (KOUT,7000) XM(NT),XI(NT),XX(NT),YY(NT)
  775 CONTINUE
C
  900 DO 100 N=1,NST
  100 READ  (KINP,1000)  SD(N,1),IPST(N),SD(N,2),
     *                   (SD(N,I),I=7,8),SD(N,15),
     *                   (SD(N,I),I=3,6),
     *                   (SD(N,I),I=9,10),
     *                   (PA(N,I),I=1,2),
     *                   (SD(N,J),J=12,13),
     *                   (PB(N,I),I=1,2)
C
      IF (NPAT.LE.0) GO TO 250
      DO 800 N=1,NST
      NT=IMST(N)
      IF (NT.GT.NPAT) CALL TABR (NST-N+1,2)
      IF (NT.LE.0) GO TO 800
      SD(N,3)=XM(NT)
      SD(N,4)=XI(NT)
      SD(N,5)=XX(NT)
      SD(N,6)=YY(NT)
  800 CONTINUE
C
  250 DO 200 N=1,NST
      SD(N,11)=SD(N,9)*(PA(N,2)-SD(N,6))-SD(N,10)*(PA(N,1)-SD(N,5))
  200 SD(N,14)=SD(N,12)*(PB(N,2)-SD(N,6))-SD(N,13)*(PB(N,1)-SD(N,5))
C
      IF (NINP.EQ.0) GO TO 999
C
      LINE=99
      DO 300 N=1,NST
      CALL NUPAJ (1,1,ITEST)
```

```
      IF (ITEST.EQ.0) GO TO 300
      WRITE (KOUT,3000)
      LINE=LINE+5
  300 WRITE (KOUT,3001) SD(N,1),(MST(N),SD(N,2),(SD(N,I),I=7,8),SD(N,15)
C
      CALL NUPAJ (1,6,ITEST)
      LINE=LINE-1
      WRITE (KOUT,4000)
      DO 400 N=1,NST
      CALL NUPAJ (1,1,ITEST)
      IF (ITEST.EQ.0) GO TO 400
      WRITE (KOUT,4000)
      LINE=LINE+5
  400 WRITE (KOUT,4001) SD(N,1),(SD(N,I),I=3,6)
C
      CALL NUPAJ (1,6,ITEST)
      LINE=LINE-1
      WRITE (KOUT,5000)
      DO 500 N=1,NST
      CALL NUPAJ (1,1,ITEST)
      IF (ITEST.EQ.0) GO TO 500
      WRITE (KOUT,5000)
      LINE=LINE+5
  500 WRITE (KOUT,5001) SD(N,1),(SD(N,I),I=9,10),(PA(N,I),I=1,2)
C
      CALL NUPAJ (1,6,ITEST)
      LINE=LINE-1
      WRITE (KOUT,6000)
      DO 600 N=1,NST
      CALL NUPAJ (1,1,ITEST)
      IF (ITEST.EQ.0) GO TO 600
      WRITE (KOUT,6000)
      LINE=LINE+5
  600 WRITE (KOUT,6001) SD(N,1),(SD(N,I),I=12,13),(PB(N,I),I=1,2)
C
  999 RETURN
C
 1000 FORMAT (A5,I5,4F10.0/(4F10.0))
 1001 FORMAT (2I5,F10.0)
 1002 FORMAT (5F10.0)
 2000 FORMAT (//
     *          1X7HSEGMENT5X7HSEGMENT4X21HCOORDINATES OF CENTER
     *          9X21HDIMENSIONS OF SEGMENT/
     *          2X6HNUMBER8X4HMASS14X1HX9X1HY19X1HX9X1HY)
 3000 FORMAT (//
     *          45H STRUCTURAL STORY DATA . . .                //
     *          6H LEVEL3X9HMASS TYPE6X6HHEIGHT9X1HK-X9X1HK-Y6X6HK-ROTN)
 4000 FORMAT (//
     *          45H STRUCTURAL MASS DATA . . .                 //
     *          6H LEVEL8X4HMASS9X3HMPI10X2HXM10X2HYM)
 5000 FORMAT (//
     *          45H STRUCTURAL LATERAL LOAD CONDITION A . . .   //
     *          6H LEVEL10X2HFX10X2HFY11X1HX11X1HY)
 6000 FORMAT (//
     *          45H STRUCTURAL LATERAL LOAD CONDITION B . . .   //
     *          6H LEVEL10X2HFX10X2HFY11X1HX11X1HY)
 7000 FORMAT (//
     *          45H CALCULATED STORY MASS PROPERTIES            //
     *          45H STORY MASS----------------------------------F10.2/
     *          45H MASS MOMENT OF INERTIA----------------------F10.1/
     *          45H X-ORDINATE OF CENTER OF MASS----------------F10.2/
     *          45H Y-ORDINATE OF CENTER OF MASS----------------F10.2)
 8000 FORMAT (//
     *          45H STORY MASS TYPE NUMBER----------------------I10/
     *          45H NUMBER OF MASS SEGMENTS---------------------I10/
     *          45H MASS SCALE FACTOR---------------------------E10.3)
 2001 FORMAT (I8,F12.2,4X,F11.2,F10.2,9X,F11.2,F10.2)
 3001 FORMAT (1X,A5,I12,F12.2,3(1PE12.4))
 4001 FORMAT (1X,A5,F12.3,F12.1,2F12.2)
 5001 FORMAT (1X,A5,4F12.2)
 6001 FORMAT (1X,A5,4F12.2)
      END



      SUBROUTINE TARF
     *              (NSC,SD,BW,CP,BP,FEF,LB,LDB,LC,LP,PP,LDIG,PDIG,IFTY,
     *                   CONL,SL,S,R,X,Y,
     *                   NST,NC,NB,NS,NCP,NBP,NFEF,MCONL,NPAN,NDIG,
     *                   NN,IF,NOPT,IPLT,NAL,NSCC)
C
      DIMENSION SD(NST,15),BW(NB),CP(8,NCP),BP(8,NBP),FEF(7,NFEF),
     1 LB(NS,NB),LDB(NS,NB,4),LC(NS,NC),LP(2,NPAN),PP(2,NPAN),
     1 LDIG(3,NDIG),PDIG(NDIG),IFTY(NFEF),CONL(2,MCONL,NFEF),SL(NB),
     1 X(NC),Y(NS),NSC(NST,1)
C
      COMMON /GEN2 / RLAB(3),ANI,ANP,RIGID
      COMMON /FTAPE/ KTMP,KINP,KOUT,KSTR,NTAPE
      COMMON /IHEDR/ IHED(20),IDATE(2),NPAGE(2),MAXLIN,LINE
      COMMON /KOPT / NINP,NOUT
      COMMON /JUNK / NLD,N,ND,MM,MN,LM(6),SS(6,6),P(6,4),T(6)
C
      DIMENSION DD(4)
      DIMENSION JD(6)
      DIMENSION VERT(4)
      DIMENSION MI(4)
C
      INTEGER PP
      INTEGER PDIG
```

```
C
      DATA HI /3H  I,3H II,3HIII,3H IV/
C
      NID=4
C
C     READ AND PRINT OF FRAME/STORY CONNECTIVITY DATA
C
      K=NST-NS
      DO 65 NUMS=1,NST
      N=NUMS
   65 NSC(N,IF)=0
      DO 70 NUMS=1,NS
      N=NUMS
   70 NSC(NST-N+1,IF)=N
      IF (NSCC.LE.0) GO TO 80
      READ  (KINP,9500) (NSC(NUMS+K,IF),NUMS=1,NS)
      IF (NINP.EQ.0) GO TO 80
      CALL NUPAJ (1,(NS/14+3),ITEST)
      WRITE (KOUT,9600) (NSC(NUMS+K,IF),NUMS=1,NS)
C
   80 M=NS-1
      DO 75 NUMS=1,M
      N=NUMS
      IF (NSC(N+K,IF).LE.NSC(N+K+1,IF)) CALL TABR (N,31)
   75 CONTINUE
      DO 76 NUMS=1,NS
      N=NUMS
      IF (NSC(N+K,IF).GT.NST.OR.NSC(N+K,IF).LT.1) CALL TABR (N,31)
   76 CONTINUE
C
      DO 60 NUMS=1,NS
      N=NUMS
   60 NSC(N+K,IF)=NST-NSC(N+K,IF)+1
C
C     READ AND PRINT OF BAY WIDTHS
C
      IF(NB.EQ.0) GO TO 100
      READ  (KINP,1000) (BW(I),I=1,NB)
      IF (NINP.EQ.0) GO TO 90
      CALL NUPAJ (1,(NB/7+3),ITEST)
      WRITE (KOUT,2000) (BW(I),I=1,NB)
C
C     READ AND PRINT OF SILL  HEIGHTS ABOVE GROUND LEVEL
C
   90 READ  (KINP,1000) (SL(I),I=1,NB)
      IF (NINP.EQ.0) GO TO 100
      CALL NUPAJ (1,(NB/7+3),ITEST)
      WRITE (KOUT,1111) (SL(I),I=1,NB)
C
C     READ AND PRINT OF COLUMN PROPERTIES
C
  100 L=NCP-1
      IF (NINP.EQ.0) GO TO 125
      CALL NUPAJ (1,6,ITEST)
      LINE=LINE-1
      WRITE (KOUT,5000)
C
  125 DO 20 I=1,L
      K=1H
      READ  (KINP,1001) M,CP(7,I),(CP(J,I),J=1,6)
      IF (I.GT.1.AND.CP(1,I).EQ..0) CP(1,I)=CP(1,I-1)
      IF (NINP.EQ.0) GO TO 105
      CALL NUPAJ (1,1,ITEST)
      IF (ITEST.EQ.0) GO TO 105
      LINE=LINE+5
      WRITE (KOUT,5000)
  105 IF (M.GE.0) GO TO 25
      K=1H-
      CP(2,I)=CP(5,I)*CP(6,I)
      CP(3,I)=CP(2,I)*CP(5,I)*CP(5,I)/12.0
      CP(4,I)=CP(2,I)*5.0/6.0
   25 IF (NINP.EQ.0) GO TO 20
      M=IABS(M)
      IF (M.NE.I) CALL TABR (M,3)
      WRITE (KOUT,4000) K,M,CP(7,I),(CP(J,I),J=1,6)
   20 CONTINUE
      DO 110 I=1,8
  110 CP(I,NCP)=0.
C
C     READ AND PRINT OF BEAM PROPERTIES
C
      IF (NBP.EQ.1) GO TO 115
      L=NBP-1
      IF (NINP.EQ.0) GO TO 135
      CALL NUPAJ (1,6,ITEST)
      LINE=LINE-1
      WRITE (KOUT,6000)
  135 DO 50 I=1,L
      K=1H
      B=0.
      READ  (KINP,1002) M,BP(8,I),(BP(J,I),J=1,7)
      IF (I.GT.1.AND.BP(1,I).EQ..0) BP(1,I)=BP(1,I-1)
      IF (BP(3,I).EQ..0) BP(3,I)=4.0
      IF (BP(4,I).EQ..0) BP(4,I)=0.5
      IF (NINP.EQ.0) GO TO 116
      CALL NUPAJ (1,1,ITEST)
      IF (ITEST.EQ.0) GO TO 116
      WRITE (KOUT,6000)
      LINE=LINE+5
  116 IF (M.GE.0) GO TO 55
```

```
      K=IM-                                                             TABS80  881
      A=BP(7,I)                                                         TABS80  882
      O=(BP(5,I)+BP(6,I))                                               TABS80  883
      BP(7,I)=B*D                                                       TABS90  884
      BP(2,I)=BP(7,I)*D*U/12.0                                          TABS80  885
      BP(7,I)=BP(7,I)*5.0/6.0                                           TABS80  886
   55 IF (NINP.EQ.0) GO TO 50                                           TABS80  887
      M=IABS(P)                                                         TABS80  888
      IF (M.NE.I) CALL TABR (M,4)                                       TABS80  889
      WRITE (KOUT,7000) K,M,BP(8,I),(BP(J,I),J=1,7),B                   TABS80  890
   50 CONTINUE                                                          TABS80  891
  115 DO 120 I=1,8                                                      TABS80  892
  120 BP(I,NBP)=0.                                                      TABS80  893
                                                                        TABS80  894
      READ AND PRINT OF FIXED END BEAM LOADS                            TABS80  895
                                                                        TABS80  896
      IF (NFEF.EQ.0) GO TO 208                                          TABS80  897
                                                                        TABS80  898
      DO 201 I=1,NFEF                                                   TABS80  899
      READ  (KINP,1003) M,IFTY(I),(FEF(J,I),J=1,4)                      TABS80  900
      IF (M.NE.I) CALL TABR (M,5)                                       TABS80  901
      IF (IFTY(I).GT.MCONL) CALL TABR (M,6)                             TABS80  902
      IF (IFTY(I).LT.0     ) CALL TABR (M,6)                            TABS80  903
      READ  (KINP,1008) (FEF(J,I),J=5,7)                                TABS80  904
      NCON=IFTY(I)                                                      TABS80  905
      IF (NCON.LE.0) GO TO 201                                          TABS80  906
      READ  (KINP,1009) ((CONL(K,NCN,I),K=1,2),NCN=1,NCON)              TABS80  907
  201 CONTINUE                                                          TABS80  908
      IF (NINP.EQ.0) GO TO 208                                          TABS80  909
      CALL NUPAJ (1,6,ITEST)                                            TABS80  910
      LINE=LINE-1                                                       TABS80  911
      WRITE (KOUT,2003)                                                 TABS80  912
      DO 202 I=1,NFEF                                                   TABS80  913
      CALL NUPAJ (1,1,ITEST)                                            TABS80  914
      IF (ITEST.EQ.0) GO TO 202                                         TABS80  915
      WRITE (KOUT,2001)                                                 TABS80  916
      LINE=LINE+5                                                       TABS80  917
  202 WRITE (KOUT,1003) I,IFTY(I),(FEF(J,I),J=1,4)                      TABS80  918
                                                                        TABS80  919
      CALL NUPAJ (1,6,ITEST)                                            TABS80  920
      LINE=LINE-1                                                       TABS90  921
      WRITE (KOUT,2008)                                                 TABS80  922
      DO 203 I=1,NFEF                                                   TABS80  923
      CALL NUPAJ (1,1,ITEST)                                            TABS80  924
      IF (ITEST.EQ.0) GO TO 203                                         TABS80  925
      WRITE (KOUT,2008)                                                 TABS80  926
      LINE=LINE+5                                                       TABS80  927
  203 WRITE (KOUT,3008) I,(FEF(J,I),J=5,7)                              TABS80  928
                                                                        TABS80  929
      IF (MCONL.EQ.0) GO TO 208                                         TABS80  930
      CALL NUPAJ (1,6,ITEST)                                            TABS80  931
      LINE=LINE-1                                                       TABS80  932
      WRITE (KOUT,2009)                                                 TABS80  933
      DO 204 I=1,NFEF                                                   TABS80  934
      NCON=IFTY(I)                                                      TABS80  935
      IF (NCON.LE.0) GO TO 204                                          TABS80  936
      CALL NUPAJ (1,2,ITEST)                                            TABS80  937
      LINE=LINE-1                                                       TABS80  938
      IF (ITEST.EQ.0) GO TO 205                                         TABS80  939
      WRITE (KOUT,2009)                                                 TABS80  940
      LINE=LINE+5                                                       TABS80  941
  205 WRITE (KOUT,3010) I                                               TABS80  942
      DO 206 NCN=1,NCON                                                 TABS80  943
      CALL NUPAJ (1,1,ITEST)                                            TABS80  944
      IF (ITEST.EQ.0) GO TO 206                                         TABS80  945
      WRITE (KOUT,2009)                                                 TABS80  946
      LINE=LINE+4                                                       TABS80  947
  206 WRITE (KOUT,3009) NCN,(CONL(K,NCN,I),K=1,2)                       TABS80  948
  204 CONTINUE                                                          TABS80  949
C                                                                       TABS80  950
C     READ (OR GENERATE) AND PRINT COLUMN LOCATIONS - LC                TABS80  951
C                                                                       TABS80  952
  208 DO 360 M=1,NC                                                     TABS80  953
      K=0                                                               TABS80  954
      DO 350 NUMS=1,NS                                                  TABS80  955
      N=NUMS                                                            TABS80  956
      IF (K.NE.0) GO TO 320                                             TABS80  957
      LN=NS+1-N                                                         TABS80  958
      READ  (KINP,1005) LC(N,M),K                                       TABS80  959
      IF (LC(N,M).LT.0  ) CALL TABR(M,10)                               TABS80  960
      IF (LC(N,M).GE.NCP) CALL TABR(M,10)                               TABS80  961
      IF(LC(N,M).EQ.0) LC(N,M)=NCP                                      TABS80  962
      GO TO 350                                                         TABS80  963
  320 K=K-1                                                             TABS80  964
      LC(N,M)=LC(N-1,M)                                                 TABS80  965
  350 CONTINUE                                                          TABS80  966
  360 CONTINUE                                                          TABS80  967
C                                                                       TABS80  968
      IF (NINP.EQ.0) GO TO 385                                          TABS80  969
      CALL NUPAJ (1,8,ITEST)                                            TABS80  970
      LINE=LINE-1                                                       TABS80  971
      WRITE (KOUT,5004)                                                 TABS80  972
      CALL TABP (NSC,LC,SD,NS,NC,1,1,NCP,NST,IF)                        TABS80  973
C                                                                       TABS80  974
  385 IF (NA.EQ.0) GO TO 300                                            TABS80  975
C                                                                       TABS80  976
C     READ (OR GENERATE) AND PRINT OF BEAM LOCATION CARDS - LB          TABS80  977
C                                                                       TABS80  978
      DO 260 M=1,NA                                                     TABS80  979
      K=0                                                               TABS80  980
```

```
      DO 250 NUMS=1,NS
      N=NUMS
      IF (K.NE.0) GO TO 220
      LN=NS+1-N
      READ (KINP,1004) LB(N,M),K
      IF (LB(N,M).LT.0) CALL TABR (M,7)
      IF (LB(N,M).GE.NBP) CALL TABR (M,7)
      IF (LB(N,M).EQ.0) LB(N,M)=NBP
      GO TO 250
  220 K=K-1
      LB(N,M)=LB(N-1,M)
  250 CONTINUE
  260 CONTINUE

      IF (NINP.EQ.0) GO TO 365
      CALL NUPAJ (1,8,ITEST)
      LINE=LINE-1
      WRITE (KOUT,3004)
      CALL TABR (NSC,LB,SD,NS,NR,1,1,NBP,NST,IF)

      READ AND PRINT OF PANEL CARDS - LP,PP

  365 IF(NPAN.EQ.0) GO TO 311
      I=0
  367 I=I+1
      READ  (KINP,1006) (LP(J,I),J=1,2),(PP(J,I),J=1,2),K
      IF (I.EQ.NPAN) GO TO 368
      IF (K.LE.0) GO TO 367
  366 I=I+1
      K=K-1
      LP(1,I)=LP(1,I-1)-1
      LP(2,I)=LP(2,I-1)
      PP(1,I)=PP(1,I-1)
      PP(2,I)=PP(2,I-1)
      IF (I.EQ.NPAN) GO TO 368
      IF (K.LE.0) GO TO 367
      GO TO 366

  368 IF (NINP.EQ.0) GO TO 311
      CALL NUPAJ (1,6,ITEST)
      LINE=LINE-1
      WRITE (KOUT,2006)
      DO 369 I=1,NPAN
      IF (LP(1,I).LE.0 ) CALL TABR (I,11)
      IF (LP(1,I).GT.NS) CALL TABR (I,11)
      IF (LP(2,I).LT.0 ) CALL TABR (I,12)
      IF (LP(2,I).GT.NB) CALL TABR (I,12)
      IF (PP(1,I).LE.0 ) CALL TABR (I,12)
      IF (PP(1,I).GT.NB) CALL TABR (I,12)
      IF (PP(2,I).LT.0  ) CALL TABR (I,13)
      IF (PP(2,I).GE.NCP) CALL TABR (I,13)
      IF (LP(2,I).GT.PP(1,I)) CALL TABR (1,19)
      CALL NUPAJ (1,1,ITEST)
      IF (ITEST.EQ.0) GO TO 369
      WRITE (KOUT,2006)
      LINE=LINE+5
  369 WRITE (KOUT,3006) I,(LP(J,I),J=1,2),(PP(J,I),J=1,2)
C
C     READ AND PRINT OF DIAGONAL CARDS
C
  311 IF (NDIG.EQ.0) GO TO 380
      I=0
  387 I=I+1
      READ  (KINP,1007) (LDIG(J,I),J=1,3),PDIG(I),K
      IF (I.EQ.NDIG) GO TO 388
      IF (K.LE.0) GO TO 387
  386 I=I+1
      K=K-1
      LDIG(1,I)=LDIG(1,I-1)-1
      LDIG(2,I)=LDIG(2,I-1)
      LDIG(3,I)=LDIG(3,I-1)
      PDIG(I)=PDIG(I-1)
      IF (I.EQ.NDIG) GO TO 388
      IF (K.LE.0) GO TO 387
      GO TO 386
C
  388 IF (NINP.EQ.0) GO TO 380
      CALL NUPAJ (1,6,ITEST)
      LINE=LINE-1
      WRITE (KOUT,2007)
      DO 389 I=1,NDIG
      M=I
      IF (LDIG(1,I).LE.0 ) CALL TABR (M,14)
      IF (LDIG(1,I).GT.NS) CALL TABR (M,14)
      IF (LDIG(2,I).LE.0 ) CALL TABR (M,15)
      IF (LDIG(2,I).GT.NC) CALL TABR (M,15)
      IF (LDIG(3,I).LE.0 ) CALL TABR (M,15)
      IF (LDIG(3,I).GT.NC) CALL TABR (M,15)
      IF (PDIG(I).LT.0  ) CALL TABR (M,16)
      IF (PDIG(I).GE.NCP) CALL TABR (M,16)
      IF (LDIG(2,I).EQ.LDIG(3,I)) CALL TABR (M,20)
      CALL NUPAJ (1,1,ITEST)
      IF (ITEST.EQ.0) GO TO 389
      WRITE (KOUT,2007)
      LINE=LINE+5
  389 WRITE (KOUT,3007) I,(LDIG(J,I),J=1,3),PDIG(I)
C
C     READ (OR GENERATE) AND PRINT OF BEAM LOADING DATA - LDB
C
  380 DO 375 NUMS=1,NS
```

```
      N=NUPS
      DO 375 M=1,NB
      DO 375 L=1,NLD
  375 LDR(N,M,L)=0

      IF (NEFF.EQ.0) GO TO 300

      DO 265 M=1,NB
      K=0
      DO 255 NUPS=1,NS
      N=NUPS
      IF (K.NE.0) GO TO 225
      LN=NS+1-N
      READ (NINP,1004) (LDR(N,M,L),L=1,NLD),K
      DO 215 L=1,NLD
      IF (LDR(N,M,L).GT.NEFF) CALL TABR (M,8)
      IF (LDR(N,M,L).LT.0   ) CALL TABR (M,8)
  215 CONTINUE
      GO TO 255
  225 K=K-1
      DO 230 L=1,NLD
  230 LDR(N,M,L)=LDR(N-1,M,L)
  255 CONTINUE
  265 CONTINUE

      IF (NINP.EQ.0) GO TO 300
      DO 270 L=1,NLD
      CALL NUPAJ (L,R,ITEST)
      LINE=LINE-1
      WRITE (NOUT,6004) H(L)
  270 CALL TABM (NSC,LDB,SD,NS,NB,3,L,100000,NST,IF)

  300 IF (IPLT.EQ.0) GO TO 945

      INITIALIZING FRAME ELEVATION PLOT

      X(1)=0.
      IF (NC.EQ.1) GO TO 915
      DO 916 M=2,NC
  916 X(M)=X(M-1)+W(M-1)

  915 COM=0.
      K=NST-NS
      DO 937 LV=1,NS
      N=NS-LV+1
      SXL=0.
      I=NSC(N+K,IF)
      L=NST
      IF (N.NE.NS) L=NSC(N+K+1,IF)-1
      DO 938 J=I,L
  938 SXL=SXL+SD(J,2)
      Y(N)=COM+SXL
  937 COM=Y(N)
C
      DO 935 I=1,4
  935 DD(I)=0.
C
      DO 920 NUMS=1,NS
      N=NUMS
      ML=LC(N,1)
      MR=LC(N,NC)
      IF (CP(5,ML)/2..GT.DD(1)) DD(1)=CP(5,ML)/2.
      IF (CP(5,MR)/2..GT.DD(2)) DD(2)=CP(5,MR)/2.
  920 CONTINUE
C
      IF (NB.EQ.0) GO TO 926
      DO 925 M=1,NB
      MT=LB(1,M)
      IF (BP(6,MT).GT.DD(3)) DD(3)=BP(6,MT)
  925 CONTINUE
C
      DO 930 L=1,NPAN
      MC=PP(2,L)
      IF (MC.EQ.0) GO TO 930
      M=LP(2,L)
      J=PP(1,L)+1
      D=0.5*(X(J)-X(M)-CP(5,MC))
      D=-D
      IF (D.LT.,0) D=0.
      IF (M.EQ.1.AND.D.GT.DD(1)) DD(1)=D
      IF (J.EQ.NC.AND.D.GT.DD(2)) DD(2)=D
  930 CONTINUE
C
  926 CALL FFRAME (NSC,SD,X,Y,DD,NST,NS,NC,IF,1)
C
  945 CONTINUE
C
C     STORY BY STORY FORMATION AND REDUCTION OF STIFFNESS MATRIX
C
      K=NST-NS
      MM=2*NC
      MN=4*NC
      NN=MN*NS+1
      CALL TABX(1,5,R,NN)
C
      IF (NINP.EQ.0) GO TO 399
      CALL NUPAJ (1,R,ITEST)
      LINE=LINE-1
      WRITE (NOUT,8000)
C
```

A13

```
  399 DO 700 NUMS=1,NS                                              TABS80 1181
      N=NUMS                                                        TABS80 1182
C                                                                   TABS80 1183
C     CALCULATING REAL STORY HEIGHT OF FRAME                        TABS80 1184
C                                                                   TABS80 1185
      SXL=0.                                                        TABS80 1186
      I=NSC(N+K,IF)                                                 TABS80 1187
      L=NST                                                         TABS80 1188
      IF (N.NE.NS) L=NSC(N+K+1,IF)-1                                TABS80 1189
      DO 710 J=I,L                                                  TABS80 1190
  710 SXL=SXL+SD(J,2)                                               TABS80 1191
C                                                                   TABS80 1192
      CALL TARX(2,S,R,NN)                                           TABS80 1193
C                                                                   TABS80 1194
C      1. FORM COLUMN MATRICES                                      TABS80 1195
C                                                                   TABS80 1196
      DO 395 I=1,6                                                  TABS80 1197
      DO 390 L=1,NLD                                                TABS80 1198
  390 P(I,L)=0.0                                                    TABS80 1199
      DO 395 J=1,6                                                  TABS80 1200
  395 SS(I,J)=0.0                                                   TABS80 1201
      DO 10 I=1,NLD                                                 TABS80 1202
   10 VERT(I)=0.0                                                   TABS80 1203
C                                                                   TABS80 1204
      ND=6                                                          TABS80 1205
      DO 500 M=1,NC                                                 TABS80 1206
      MC=LC(N,M)                                                    TABS80 1207
      IF(MC.EQ.NCP) GO TO 500                                       TABS80 1208
C                                                                   TABS80 1209
      P(6,1)=CP(7,MC)*CP(2,MC)*SD(N+K,2)                            TABS80 1210
      VERT(1)=VERT(1)+P(6,1)                                        TABS80 1211
C                                                                   TABS80 1212
      DO 907 I=1,4                                                  TABS80 1213
  907 DD(I)=0.                                                      TABS80 1214
      A=0.                                                          TABS80 1215
      B=0.                                                          TABS80 1216
      XL=SXL                                                        TABS80 1217
      IF (NB.EQ.0) GO TO 420                                        TABS80 1218
      IF (N.NE.NS) GO TO 900                                        TABS80 1219
C                                                                   TABS80 1220
C     N.EQ.NS                                                       TABS80 1221
C                                                                   TABS80 1222
      IF (M.EQ.1) GO TO 905                                         TABS80 1223
      IF (M.EQ.NC) GO TO 906                                        TABS80 1224
      ML=LB(N,M-1)                                                  TABS80 1225
      MR=LB(N,M)                                                    TABS80 1226
      A=BP(5,ML)                                                    TABS80 1227
      IF (A.LT.BP(5,MR)) A=BP(5,MR)                                 TABS80 1228
      DD(1)=BP(5,ML)                                                TABS80 1229
      DD(2)=BP(5,MR)                                                TABS80 1230
      B=SL(M-1)                                                     TABS80 1231
      IF (B.LT.SL(M)) B=SL(M)                                       TABS80 1232
      DD(3)=SL(M-1)                                                 TABS80 1233
      DD(4)=SL(M)                                                   TABS80 1234
      GO TO 420                                                     TABS80 1235
  905 MR=LB(N,M)                                                    TABS80 1236
      A=BP(5,MR)                                                    TABS80 1237
      B=SL(M)                                                       TABS80 1238
      DD(2)=A                                                       TABS80 1239
      DD(4)=B                                                       TABS80 1240
      GO TO 420                                                     TABS80 1241
  906 ML=LB(N,M-1)                                                  TABS80 1242
      A=BP(5,ML)                                                    TABS80 1243
      B=SL(M-1)                                                     TABS80 1244
      DD(1)=A                                                       TABS80 1245
      DD(3)=B                                                       TABS80 1246
      GO TO 420                                                     TABS80 1247
C                                                                   TABS80 1248
C     N.NE.NS                                                       TABS80 1249
C                                                                   TABS80 1250
  900 IF (M.EQ.1) GO TO 901                                         TABS80 1251
      IF (M.EQ.NC) GO TO 910                                        TABS80 1252
      MRR=LB(N+1,M)                                                 TABS80 1253
      MR=LB(N,M)                                                    TABS80 1254
      MLL=LB(N+1,M-1)                                               TABS80 1255
      ML=LB(N,M-1)                                                  TABS80 1256
      DD(1)=BP(5,ML)                                                TABS80 1257
      DD(2)=BP(5,MR)                                                TABS80 125
      DD(3)=BP(6,MLL)                                               TABS80 1259
      DD(4)=BP(6,MRR)                                               TABS80 1260
      A=BP(5,ML)                                                    TABS80 1261
      B=BP(6,MLL)                                                   TABS80 1262
      IF (A.LT.BP(5,MR)) A=BP(5,MR)                                 TABS80 1263
      IF (B.LT.BP(6,MRR)) B=BP(6,MRR)                               TABS80 1264
      GO TO 420                                                     TABS80 1265
  901 MR=LB(N,M)                                                    TABS80 1266
      MRR=LB(N+1,M)                                                 TABS80 1267
      A=BP(5,MR)                                                    TABS80 1268
      B=BP(6,MRR)                                                   TABS80 1269
      DD(2)=A                                                       TABS80 1270
      DD(4)=B                                                       TABS80 1271
      IF (N.EQ.1) DD(1)=-BP(6,MR)                                   TABS80 1272
      GO TO 420                                                     TABS80 1273
  910 ML=LB(N,M-1)                                                  TABS80 1274
      MLL=LB(N+1,M-1)                                               TABS80 1275
      A=BP(5,ML)                                                    TABS80 1276
      B=BP(6,MLL)                                                   TABS80 1277
      DD(1)=A                                                       TABS80 1278
      DD(3)=B                                                       TABS80 1279
      IF (N.EQ.1) DD(2)=-BP(6,ML)                                   TABS80 1280
```

```
C
  420 D=CP(5,MC)*RIGID
      IF ((XL-A-B).LE..0) CALL TABR (M,17)
C
      IF (IPLT.EQ.0) GO TO 950
      Y1=0.
      IF (N.NE.NS) Y1=Y(N+1)
      JD(5)=MC
      IF (JD(5).EQ.NCP) JD(5)=0
      CALL ELPLOT (X(M),B,Y1,Y(N),DD,CP(5,MC),CP(5,MC),JD,IPLT,1)
  950 CONTINUE
C
      IF (NOPT.EQ.1) GO TO 500
      A=A-D
      B=B-D
      IF (B.LT.0.0) B=0.0
      IF (A.LT.0.0) A=0.0
      XL=XL-A-B
      BB=0.0
      IF ( CP(4,MC) ) 430,430,425
  425 BB=14.4*CP(3,MC)/(XL**2*CP(4,MC))
  430 COMM=2.*CP(1,MC)*CP(3,MC)/(XL*(1.+2.*BB))
      SA=COMM*(2.+BB)
      SB=COMM*(1.-BB)
      SC=CP(2,MC)*CP(1,MC)/XL
      CALL TABY(SA,SB,SC,XL,A,B)
      LM(3)=2*M
      LM(1)=LM(3)+MM
      LM(5)=LM(1)-1
      LM(6)=LM(3)-1
      LM(4)=MM+N
      LM(2)=LM(4)+1
      CALL TABX(5,S,P,MM)
  500 CONTINUE
      P(6,1)=0.0
C
       2. FORM BEAM MATRICES
C
      IF (NB.EQ.0) GO TO 615
      ND=4
      DO 600 M=1,NB
      MB=LB(N,M)
      IF(MB.EQ.NBP) GO TO 600
      DO 610 I=1,4
  610 DD(I)=0.
      ML=LC(N,M)
      MR=LC(N,M+1)
      A=CP(5,MR)/2.
      B=CP(5,ML)/2.
      IF (IPLT.EQ.0) GO TO 960
      IF (M.EQ.1) DD(3)=-B
      IF (M.EQ.NB) DD(1)=-A
      DD(4)=B
      DD(2)=A
      IF (N.EQ.1) GO TO 605
      MLL=LC(N-1,M)
      MRR=LC(N-1,M+1)
      DD(3)=CP(5,MLL)/2.
      DD(1)=CP(5,MRR)/2.
  605 CONTINUE
      DO 535 L=1,NLD
  535 JD(L)=LDB(N,M,L)
      JD(5)=NFFF
      JD(6)=MB
      IF (JD(6).EQ.NBP) JD(6)=0
      CALL ELPLOT (X(M),X(M+1),Y(N),Y(N),DD,BP(5,MB),BP(6,MB),JD,IPLT,2)
  960 CONTINUE
C
      WA=A
      WB=B
      IF (M.EQ.NB) WA=2.0*WA
      IF (M.EQ.1)  WB=2.0*WB
      D=(BP(5,MB)+BP(6,MB))*RIGID
      IF ((BW(M)-A-B).LE..0) CALL TABR (M,18)
      AAA=A
      BBB=B
      A=A-D
      B=B-D
      IF (A.LT.0.0) A=0.0
      IF (B.LT.0.0) B=0.0
      XL=BW(M)-A-B
      IF (NOPT.EQ.1) GO TO 429
      BB=0.0
      IF (BP(3,MB).NE.4.0.OR.BP(4,MB).NE.0.5) GO TO 431
      IF (BP(7,MB)) 426,426,427
  427 BB=14.4*BP(2,MB)/(XL**2*BP(7,MB))
  426 COMM=2.*BP(1,MB)*BP(2,MB)/(XL*(1+2.*BB))
      SA=COMM*(2.+BB)
      SB=COMM*(1.-BB)
      GO TO 428
  431 EIOL=BP(1,MB)*BP(2,MB)/XL
      SA=BP(3,MB)*EIOL
      SB=BP(4,MB)*SA
  428 CALL TABY(SA,SB,0.0,XL,A,B)
      LM(1)=2*M
      LM(2)=LM(1)-1
      LM(3)=LM(1)+2
      LM(4)=LM(3)-1
  429 A=AAA
```

```
      B=BBB
      XL=HW(M)-AAA-BBB

      DO 550 L=1,NLD
      DO 520 I=1,4
  520 P(I,L)=0.
      J=LDB(N,M,L)
      IF (J) 550,550,510

  510 WW=FEF(5,J)
      IF (L.EQ.NAL) WW=WW+BP(7,MB)+BP(8,MB)
      V=WW*XL/2.
      VM=V*XL/6.
      P(1,L)=FEF(1,J)+VM
      P(2,L)=FEF(2,J)+V
      P(3,L)=-FEF(3,J)-VM
      P(4,L)=FEF(4,J)+V

      NCON=IFTY(J)
      IF (NCON.EQ.0) GO TO 513
      TEMP=0.
      DO 512 NCN=1,NCON
      P1=CONL(2,NCN,J)
      DIST=CONL(1,NCN,J)
      IF (DIST.LT..0) DIST=HW(M)*ABS(DIST)
      IF (DIST.LT.TEMP) CALL TABR(J,9)
      TEMP=DIST
      AA=DIST-B
      IF (AA.LE..0) CALL TABR (M,21)
      BB=XL-AA
      COM=P1*AA*BB/XL/XL
      P(1,L)=P(1,L)+COM*BB
      P(3,L)=P(3,L)-COM*AA
      COM=P1/XL**3
      P(2,L)=P(2,L)+COM*BB*BB*(3.*AA+BB)
  512 P(4,L)=P(4,L)+COM*AA*AA*(AA+3.*BB)

  513 P(1,L)=P(1,L)+P(2,L)*B
      P(3,L)=P(3,L)-P(4,L)*A
      P(2,L)=P(2,L)+WB*FEF(5,J)+FEF(6,J)
      P(4,L)=P(4,L)+WA*FEF(5,J)+FEF(7,J)

      MODIFYING FEFS FOR POSSIBLE PINS ENDS

      CALL TABG (BP(2,MB),BP(3,MB),BP(4,MB),P(1,L),P(3,L),P(2,L),P(4,L),
     *           XL,J)
  540 VERT(L)=VERT(L)+P(2,L)+P(4,L)

  550 CONTINUE
C
C     CREATING SPECIAL AUTOMATIC LOAD VECTORS FOR PROGRAM STACI
C
      IF (NAL.EQ.1) GO TO 650
      DO 630 L=3,NLD
      I=L-2
      J=LDB(N,M,L)
      IF (J.EQ.0) GO TO 630
C
      BB=WB*FEF(5,J)+FEF(6,J)
      AA=WA*FEF(5,J)+FEF(7,J)
C
      P(2,L)=P(2,L)-BB
      P(4,L)=P(4,L)-AA
      P(2,1)=P(2,1)+BB
      P(4,1)=P(4,1)+AA
C
      VERT(L)=VERT(L)-AA-BB
      VERT(1)=VERT(1)+AA+BB
C
  630 CONTINUE
  650 CONTINUE
C
      IF (NOPT.EQ.1) GO TO 600
      CALL TABX(5,S,R,NN)
  600 CONTINUE
C
      DO 720 I=1,6
      DO 720 J=1,NLD
  720 P(I,J)=0.
      DO 741 I=1,4
  741 DD(I)=0.
C
C       3. FORM PANEL STIFFNESS
C
      IF(NPAN.EQ.0) GO TO 611
      ND=6
      DO 800 L=1,NPAN
      NP=NS-LP(1,L)+1
      IF(NP.NE.N) GO TO 800
C
      M=LP(2,L)
      J=PP(1,L)
      MC=PP(2,L)
      IF (MC.EQ.0) GO TO 800
C
      XL=SXL
      P(5,1)=XL*CP(2,MC)*CP(7,MC)/2.0
      P(6,1)=P(5,1)
      VERT(1)=VERT(1)+P(5,1)+P(6,1)
C
```

```
      IF (IPLT.EQ.0) GO TO 980
      Y1=0.
      IF (N.NE.NS) Y1=Y(N+1)
      JD(4)=L
      JD(5)=MC
      CALL ELPLOT (X(M),X(J+1),Y1,Y(N),DD,CP(5,MC),CP(5,MC),JD,IPLT,3)
  980 CONTINUE

      IF (NOPT.EQ.1) GO TO 600

      D=0.0
      DO 740 I=M,J
  740 D=D+BW(I)

      BB=XL**2*CP(4,MC)*CP(1,MC)/2.4
      IF (BB) 730,730,725
  725 BA=6.0*CP(1,MC)*CP(3,MC)/BB
  730 COMM=2.0*CP(1,MC)*CP(3,MC)/(XL*(1.+2.*BB))
      SA=COMM*(2.+BA)
      SB=COMM*(1.-BB)
      SC=CP(1,MC)*CP(2,MC)/XL

      CALL TABT (1,SA,SB,SC,XL,D,SS,CA)

      LM(2)=MN+N
      LM(1)=LM(2)+1
      LM(5)=2*M-1
      LM(6)=LM(5)+2*(J-M+1)
      LM(3)=LM(5)+MM
      LM(4)=LM(6)+MM

  780 CALL TABX (5,S,R,NN)

  900 CONTINUE

      P(5,1)=0.0
      P(6,1)=0.0
  611 CONTINUE

      4.  FORM DIAGONAL STIFFNESSES

      IF (NDIG.EQ.0) GO TO 615
      MD=6
      DO 560 M=1,NDIG
      NDI=NS-LDIG(1,M)+1
      IF (N.NE.NDI) GO TO 560

      DY=SXL
      IB=LDIG(2,M)
      IT=LDIG(3,M)
      MC=MDIG(M)
      IF (MC.EQ.0) GO TO 560
C
      DX=0.
      IF (IT.EQ.1) GO TO 521
      DO 510 I=2,IT
  510 DX=DX+BW(I-1)
  521 IF (IB.EQ.1) GO TO 541
      DO 531 I=2,IB
  531 DX=DX-BW(I-1)
  541 XL=SQRT(DX**2+DY**2)
C
      P(6,NAL)=XL*CP(2,MC)*CP(7,MC)
      VERT(NAL)=VERT(NAL)+P(6,NAL)
C

      IF (IPLT.EQ.0) GO TO 990
      Y1=0.
      IF (N.NE.NS) Y1=Y(N+1)
      JD(4)=M
      JD(5)=MC
      CALL ELPLOT (X(IB),X(IT),Y1,Y(N),DD,CP(5,MC),CP(5,MC),JD,IPLT,4)
  990 CONTINUE
C
      IF (NOPT.EQ.1) GO TO 560
C
      BB=0.0
      IF ( CP(4,MC) ) 580,580,585
  585 BA=14.4*CP(3,MC)/(XL**2*CP(4,MC))
  580 COMM=2.*CP(1,MC)*CP(3,MC)/(XL*(1.+2.*BA))
      SA=COMM*(2.+BB)
      SB=COMM*(1.-BB)
      SC=CP(2,MC)*CP(1,MC)/XL
      CALL TABY (1,SA,SB,SC,XL,DX,DY,SS,CA)
C
      LM(1)=2*IB+MM
      LM(2)=MN+N+1
      LM(3)=2*IT
      LM(4)=LM(2)-1
      LM(5)=LM(1)-1
      LM(6)=LM(3)-1
C
      CALL TABX(5,S,R,NN)

  560 CONTINUE
C
      P(6,NAL)=0.
C
C      REDUCE STIFFNESS MATRIX FOR LEVEL N
C
```

```
615 IF (NOPT.EQ.1) GO TO 570
    CALL TABX(3,S,R,NN)

570 IF (NINP.EQ.0) GO TO 700
    NL=NSC(N+K,IF)
    CALL NUPAJ (L,L,ITEST)
    IF (ITEST.EQ.0) GO TO 620
    WRITE (KOUT,8000)
    LINE=LINE+6
620 WRITE (KOUT,9000) SD(NL,I),VERT

700 CONTINUE

    WRITE LATERAL STIFFNESS ON TAPE

810 N=NST+1-NS

    IF (IPLT.EQ.0) GO TO 940

    PLOTTING SILL HEIGHTS

    IF (NB.EQ.0) GO TO 555
    DD(2)=0.
    DD(4)=0.
    DO 551 M=1,NB
    ML=LC(NS,M)
    MR=LC(NS,M+1)
    DD(1)=CP(5,MR)/2.
    DD(3)=CP(5,ML)/2.
    CALL ELPLOT (X(M),X(M+1),0.,0.,DD,SL(M),SL(M),JD,IPLT,5)
551 CONTINUE

555 CALL FFRAME (NSC,SD,X,Y,DD,NST,NS,NC,IF,2)

940 CONTINUE

    IF (NOPT.EQ.1) GO TO 999
    CALL TABX(4,S,R,NN)

    WRITE (3) Kk,CP,BP,FEF,LB,LDB,LC,LP,IFEF,PP,LDIG,PDIG,IFTY,CONL,SL
    WRITE (3) IF,NC,NS,NCP,NBP,NFEF,NPAN,NDIG,MCONL

999 RETURN

1000 FORMAT (7F10.0)
2000 FORMAT (11HOBAY WIDTHS/(7F11.2))
3000 FORMAT (/)
1111 FORMAT (/12H SILL DEPTHS/(7F11.2))
1001 FORMAT (I5,F5.0,6F10.0)
1002 FORMAT (I5,F5.0,2F10.0,2F5.0,3F10.0)
```

```
5000 FORMAT (//
   *          45H COLUMN SECTION PROPERTY DATA              //
   *          4X2HID9X1HU9X1HE9X1HA9X1HI8X2HAV6X1HW6X1HT)
4000 FORMAT (1X,A1,I4,F10.3,F10.1,2E10.3,F10.2,F7.2,F7.2)
6000 FORMAT (//
   *          45H BEAM SECTION PROPERTY DATA                //
   *          4X2HID9X1HU9X1HE9X1HI6X1HK6X1HC5X2HDB5X2HDA7X2HAV6X1HT)
7000 FORMAT (1X,A1,I4,F10.3,F10.1,E10.3,4F7.2,F9.2,F7.2)
1003 FORMAT (2I5,4F10.0)
1008 FORMAT (3F10.0)
1009 FORMAT (6F10.0)
2003 FORMAT (//
   *          45H BEAM LOADING DATA . . . FIXED END FORCES    //
   *          4X2HID6X4HNCON12X2HML8X2HVL8X2HMR8X2HVR)
2008 FORMAT (//
   *          53H BEAM LOADING DATA . . . UNIFORM AND COLUMN POINT LOS//
   *          4X2HID12X2HWW8X2HFL8X2HFR)
2009 FORMAT (//
   *          45H BEAM LOADING DATA . . . CONCENTRATED LOADS  //
   *          4X2HID9X1HN10X4HDIST9X1HP)
3003 FORMAT (I6,4X,I6,4X,4F10.2)
3008 FORMAT (I6,4X,F10.3,2F10.2)
3009 FORMAT (6X,4X,I6,4X,2F10.2)
3010 FORMAT (I6)
1004 FORMAT (9I5)
1005 FORMAT (4I5)
1006 FORMAT (5I5)
2006 FORMAT (//
   *          45H INPUT AND/OR GENERATED SHEAR PANEL DATA      //
   *          5X5HPANEL7X5HLEVEL7X5HFIRST8X4HLAST6X6HCOLUMN/
   *          8X2HID6X6HAT TOP9X3HBAY9X3HBAY5X7HPROP ID)
3006 FORMAT (I10,4I12)
1007 FORMAT (5I5)
2007 FORMAT (//
   *          45H INPUT AND/OR GENERATED DIAGONAL DATA         //
   *          2X8HDIAGONAL7X5HLEVEL6X6H LOWER7X5HUPPER6X6HCOLUMN/
   *          8X2HID6X6HAT TOP6X6HCOLUMN6X6HCOLUMN5X7HPROP ID)
3007 FORMAT (I10,4I12)
3004 FORMAT (///
   *          45H INPUT/GENERATED BEAM LOCATIONS                )
4004 FORMAT (///
   *          47H INPUT/GENERATED BEAM LOADS FOR LOAD CONDITION A3)
5004 FORMAT (///
   *          45H INPUT/GENERATED COLUMN LOCATIONS              )
8000 FORMAT (//
   *          45H TOTAL VERTICAL LOAD APPLIED ON FRAME LEVEL-B
   *          45HY-LEVEL                                       //
   *          5X5HLEVEL45H/--------VERTICAL LOAD COND--------/           /
   *          8X2HID8X1HI7X2HII6X3HIII7X2HIV/)
9000 FORMAT (5X,A5,4F9.2)
```

```
 9500 FORMAT (14I5)                                                     TABS80 1681
 9600 FORMAT (45H0STRUCTURE LEVELS CONNECTING TO FRAME        /(1X14I5)) TABS80 1682
      END                                                               TABS80 1683


      SUBROUTINE TABM (NSC,LBC,SD,NS,NR,NL,L,NPROP,NST,IF)              TABS80 1684
C                                                                       TABS80 1685
C     PRINT MEMBER LOCATIONS AS GENERATED                               TABS80 1686
C                                                                       TABS80 1687
      DIMENSION LBC(NS,NR,NL),SD(NST,15),NSC(NST,1)                     TABS80 1688
C                                                                       TABS80 1689
      COMMON /FTAPE/ KTMP,KINP,KOUT,KSTR,NTAPE                          TABS80 1690
      COMMON /IHEDR/ IHED(28),IDATE(2),NPAGE(2),MAXLIN,LINE             TABS80 1691
      COMMON /JUNK/ NLO,LOC(15)                                         TABS80 1692
C                                                                       TABS80 1693
      M=NST-NS                                                          TABS80 1694
C                                                                       TABS80 1695
      DO 300 J=1,NR,15                                                  TABS80 1696
C                                                                       TABS80 1697
      JJ=J+14                                                           TABS80 1698
      IF (JJ.GT.NR) JJ=NR                                               TABS80 1699
      WRITE (KOUT,2000) (K,K=J,JJ)                                      TABS80 1700
      WRITE (KOUT,2002)                                                 TABS80 1701
C                                                                       TABS80 1702
      DO 200 I=1,NS                                                     TABS80 1703
C                                                                       TABS80 1704
      DO 100 K=J,JJ                                                     TABS80 1705
      K1=K-J+1                                                          TABS80 1706
      LOC(K1)=LBC(I,K,L)                                                TABS80 1707
      IF (LOC(K1).EQ.NPROP) LOC(K1)=0                                   TABS80 1708
  100 CONTINUE                                                          TABS80 1709
C                                                                       TABS80 1710
      KR=JJ-J+1                                                         TABS80 1711
      ML=NSC(I+M,IF)                                                    TABS80 1712
      CALL NUPAJ (1,1,ITEST)                                            TABS80 1713
      IF (ITEST.EQ.0) GO TO 200                                         TABS80 1714
      WRITE (KOUT,2000) (K,K=J,JJ)                                      TABS80 1715
      WRITE (KOUT,2002)                                                 TABS80 1716
      LINE=LINE+1                                                       TABS80 1717
  200 WRITE (KOUT,2001) SD(ML,1),(LOC(K),K=1,KR)                        TABS80 1718
C                                                                       TABS80 1719
  300 CONTINUE                                                          TABS80 1720
C                                                                       TABS80 1721
      RETURN                                                            TABS80 1722
C                                                                       TABS80 1723
 2000 FORMAT (6H0LEVEL,2X,15I4)                                         TABS80 1724
 2001 FORMAT (1X,A5,2X,15I4)                                            TABS80 1725
 2002 FORMAT (1H )                                                      TABS80 1726
```

```
      END                                                               TABS80 1727

      SUBROUTINE TABY(SA,SB,SC,XL,A,B)                                  TABS80 1728
      COMMON /JUNK/ NLD,N,ND,MM,PN,LM(6),SS(6,6),P(6,4),T(2,4)          TABS80 1729
C                                                                       TABS80 1730
      T(2,1)=B/XL                                                       TABS80 1731
      T(1,1)=1.0+T(2,1)                                                 TABS80 1732
      T(1,2)=1.0/XL                                                     TABS80 1733
      T(2,2)=T(1,2)                                                     TABS80 1734
      T(1,3)=A/XL                                                       TABS80 1735
      T(2,3)=1.0+T(1,3)                                                 TABS80 1736
      T(1,4)=-T(1,2)                                                    TABS80 1737
      T(2,4)=T(1,4)                                                     TABS80 1738
C                                                                       TABS80 1739
      DO 100 I=1,4                                                      TABS80 1740
      D1=SA*T(1,I)+SB*T(2,I)                                            TABS80 1741
      D2=SB*T(1,I)+SA*T(2,I)                                            TABS80 1742
      DO 100 J=1,4                                                      TABS80 1743
  100 SS(J,I)=D1*T(1,J)+D2*T(2,J)                                       TABS80 1744
C                                                                       TABS80 1745
      IF(SC) 120,120,110                                                TABS80 1746
  110 SS(5,5)=SC                                                        TABS80 1747
      SS(5,6)=-SC                                                       TABS80 1748
      SS(6,5)=-SC                                                       TABS80 1749
      SS(6,6)=SC                                                        TABS80 1750
C                                                                       TABS80 1751
  120 RETURN                                                            TABS80 1752
C                                                                       TABS80 1753
      END                                                               TABS80 1754


      SUBROUTINE TART (IO,SA,SB,SC,XL,D,SS,CA)                          TABS80 1755
C                                                                       TABS80 1756
C     COLUMN PANEL STIFFNESS AND FORCE MATRICES                         TABS80 1757
C                                                                       TABS80 1758
      DIMENSION SS(6,6),CA(10,6),T(3,6)                                 TABS80 1759
C                                                                       TABS80 1760
      DO 10 I=1,3                                                       TABS80 1761
      DO 10 J=1,6                                                       TABS80 1762
   10 T(I,J)=0.0                                                        TABS80 1763
C                                                                       TABS80 1764
      T(1,1)=1.0/XL                                                     TABS80 1765
      T(2,1)=T(1,1)                                                     TABS80 1766
      T(1,2)=-T(1,1)                                                    TABS80 1767
      T(2,2)=-T(1,1)                                                    TABS80 1768
```

```
C                                                         TABS80 1769
      T(1,4)=1.0/D                                        TABS80 1770
      T(2,6)= T(1,4)                                      TABS80 1771
      T(1,3)=-T(1,4)                                      TABS80 1772
      T(2,5)=-T(1,4)                                      TABS80 1773
C                                                         TABS80 1774
      T(3,3)= 0.5                                         TABS80 1775
      T(3,4)= 0.5                                         TABS80 1776
      T(3,5)=-0.5                                         TABS80 1777
      T(3,6)=-0.5                                         TABS80 1778
C                                                         TABS80 1779
      IF (IO.EQ.2) GO TO 200                              TABS80 1780
C                                                         TABS80 1781
C     COLUMN PANEL STIFFNESS                              TABS80 1782
C                                                         TABS80 1783
  100 DO 120 J=1,6                                        TABS80 1784
      U=SA*T(1,J)+SB*T(2,J)                               TABS80 1785
      V=SB*T(1,J)+SA*T(2,J)                               TABS80 1786
      W=SC*T(3,J)                                         TABS80 1787
      DO 120 I=1,6                                        TABS80 1788
  120 SS(I,J)=U*T(1,I)+V*T(2,I)+W*T(3,I)                  TABS80 1789
      RETURN                                              TABS80 1790
C                                                         TABS80 1791
C     COLUMN PANEL FORCE DISPLACEMENT TRANSFORMATION      TABS80 1792
C                                                         TABS80 1793
  200 DO 220 I=1,6                                        TABS80 1794
      CA(1,I)=SA*T(1,I)+SB*T(2,I)                         TABS80 1795
      CA(2,I)=SB*T(1,I)+SA*T(2,I)                         TABS80 1796
      CA(3,I)=SC*T(3,I)                                   TABS80 1797
  220 CA(4,I)=-(CA(1,I)+CA(2,I))/XL                       TABS80 1798
      RETURN                                              TABS80 1799
C                                                         TABS80 1800
      END                                                 TABS80 1801


      SUBROUTINE TABV (IO,SA,SB,SC,D,L,H,SS,CA)           TABS80 1802
C                                                         TABS80 1803
C     DIAGONAL STIFFNESS AND FORCE MATRICES               TABS80 1804
C                                                         TABS80 1805
      DIMENSION SS(6,6),CA(10,6),T(3,6)                   TABS80 1806
      REAL L                                              TABS80 1807
C                                                         TABS80 1808
      DO 10 I=1,3                                         TABS80 1809
      DO 10 J=1,6                                         TABS80 1810
   10 T(I,J)=0.0                                          TABS80 1811
C                                                         TABS80 1812
      DD=D*D                                              TABS80 1813
C                                                         TABS80 1814
      T(1,1)=1.                                           TABS80 1815
      T(1,2)=H/DD                                         TABS80 1816
      T(1,4)=-T(1,2)                                      TABS80 1817
      T(1,5)=L/DD                                         TABS80 1818
      T(1,6)=-T(1,5)                                      TABS80 1819
C                                                         TABS80 1820
      T(2,2)=T(1,2)                                       TABS80 1821
      T(2,3)=T(1,1)                                       TABS80 1822
      T(2,4)=-T(1,2)                                      TABS80 1823
      T(2,5)=T(1,5)                                       TABS80 1824
      T(2,6)=-T(1,5)                                      TABS80 1825
C                                                         TABS80 1826
      T(3,2)=-L/D                                         TABS80 1827
      T(3,4)=-T(3,2)                                      TABS80 1828
      T(3,5)=H/D                                          TABS80 1829
      T(3,6)=-T(3,5)                                      TABS80 1830
C                                                         TABS80 1831
      IF (IO.EQ.2) GO TO 200                              TABS80 1832
C                                                         TABS80 1833
C     DIAGONAL STIFFNESS                                  TABS80 1834
C                                                         TABS80 1835
  100 DO 120 J=1,6                                        TABS80 1836
      U=SA*T(1,J)+SB*T(2,J)                               TABS80 1837
      V=SB*T(1,J)+SA*T(2,J)                               TABS80 1838
      W=SC*T(3,J)                                         TABS80 1839
      DO 120 I=1,6                                        TABS80 1840
  120 SS(I,J)=U*T(1,I)+V*T(2,I)+W*T(3,I)                  TABS80 1841
      RETURN                                              TABS80 1842
C                                                         TABS80 1843
C     DIAGONAL FORCE DISPLACEMENT TRANSFORMATION          TABS80 1844
C                                                         TABS80 1845
  200 DO 220 I=1,6                                        TABS80 1846
      CA(1,I)=SA*T(1,I)+SB*T(2,I)                         TABS80 1847
      CA(2,I)=SB*T(1,I)+SA*T(2,I)                         TABS80 1848
      CA(3,I)=SC*T(3,I)                                   TABS80 1849
  220 CA(4,I)=-(CA(1,I)+CA(2,I))/D                        TABS80 1850
      RETURN                                              TABS80 1851
C                                                         TABS80 1852
      END                                                 TABS80 1853


      SUBROUTINE TABX(IO,S,R,NN)                          TABS80 1854
      DIMENSION S(NN,NN),R(NN,1)                          TABS80 1855
      COMMON /JUNK/ NLD,N,ND,MM,MN,LM(6),SS(6,6),P(6,4)   TABS80 1856
C                                                         TABS80 1857
      GO TO (1,2,3,4,5),IO                                TABS80 1858
C                                                         TABS80 1859
C     IO=1 ZERO MATRICES                                  TABS80 1860
```

AD-A107 635 COMPUTERS/STRUCTURES INTERNATIONAL OAKLAND CA F/G 13/13
THEORETICAL BASIS FOR CTABS80: A COMPUTER PROGRAM FOR THREE-DIM--ETC(U)
SEP 81 E L WILSON, H H DOVEY, A HABIBULLAH
UNCLASSIFIED WES-TR-K-81-2 NL

2 OF 2
AD A
107635

END
DATE
FILMED
11 -82
DTIC

```
C
    1 DO 200 I=1,NN
      DO 190 L=1,NLD
  190 R(I,L)=0.0
      DO 200 J=1,NN
  200 S(I,J)=0.0
      GO TO 800
C
C     IO=2 SHIFT STORAGE
C
    2 DO 220 I=1,MM
      II=MM+I
      DO 210 J=1,MM
      JJ=MM+J
      S(I,J)=S(II,JJ)
      S(II,JJ)=0.0
  210 S(I,JJ)=0.0
      DO 215 L=1,NLD
      R(I,L)=R(II,L)
  215 R(II,L)=0.0
      JL=MN+1
      DO 220 J=JL,NN
      S(I,J)=S(II,J)
  220 S(II,J)=0.0
      GO TO 800
C
C     IO=3 REDUCE MATRIX AND WRITE B.S. EQUATIONS ON TAPE 3
C
    3 NNN=MN+N+1
      DO 345 I=1,MM
      IF (S(I,I)) 345,345,305
C
  305 DO 320 L=1,NLD
  320 R(I,L)=R(I,L)/S(I,I)
C
      JJ=I+1
      DO 310 J=JJ,NNN
      S(J,I)=S(I,J)
  310 S(I,J)=S(I,J)/S(I,I)
C
      DO 340 K=JJ,NNN
      DO 330 J=K,NNN
  330 S(K,J)=S(K,J)-S(K,I)*S(I,J)
      DO 340 L=1,NLD
  340 R(K,L)=R(K,L)-S(K,I)*R(I,L)
  345 CONTINUE
C
      WRITE (3) (((S(I,J),J=I,NN),(R(I,L),L=1,NLD)),I=1,MM)
      GO TO 800
C
C     IO=4 WRITE LATERAL STIFFNESS MATRIX ON TAPE 2
C
    4 IL=MN+1
      IH=NN-1
      NS=NN-MN-1
      LRD=((NS+1)*NS)/2
      WRITE (2) LRD,N ,((S(I,J),J=I,IH),I=IL,IH),((R(I,L),I=IL,IH),
     1           L=1,NLD)
      GO TO 800
C
C     IO=5 ADD ELEMENT STIFFNESS TO TOTAL STIFFNESS
C
    5 DO 550 I=1,ND
      II=LM(I)
      DO 510 L=1,NLD
  510 R(II,L)=R(II,L)+P(I,L)
      DO 550 J=1,ND
      JJ=LM(J)
  550 S(II,JJ)=S(II,JJ)+SS(I,J)
      GO TO 800
C
  800 RETURN
C
      END




      SUBROUTINE TABQ (A,N,B,NL)
C
C     SOLUTION OF SYMMETRICAL LINEAR EQUATIONS- E L WILSON
C
      DIMENSION A(N,N),B(N,NL)
C
      M=0
C
C     REDUCTION OF M TH EQUATION
C
   50 M=M+1
      MM=M+1
      DO 60 L=1,NL
   60 B(M,L)=B(M,L)/A(M,M)
      IF (M-N) 70,130,70
   70 DO 80 J=MM,N
   80 A(M,J)=A(M,J)/A(M,M)
C
C     SUBSTITUTION INTO REMAINING EQUATIONS
C
      DO 120 I=MM,N
      IF (A(I,M)) 90,120,90
```

```
   90 DO 100 J=I,N
      A(I,J)=A(I,J)-A(I,M)*A(M,J)
  100 A(J,I)=A(I,J)
      DO 110 L=1,NL
  110 B(I,L)=B(I,L)-A(I,M)*B(M,L)
  120 CONTINUE
      GO TO 50
C
C     BACK SUBSTITUTION
C
  130 M=M-1
      IF (M.EQ.0) GO TO 150
      MM=M+1
      DO 140 L=1,NL
      DO 140 J=MM,N
  140 B(M,L)=B(M,L)-A(M,J)*B(J,L)
      GO TO 130
C
  150 RETURN
      END


      SUBROUTINE TABL (NSC,A,SS,RR,S,R,D,AA,MSS,MST,MTF,MED)
C
C     ASSEMBLING LATERAL STIFFNESS
C
      DIMENSION SS(MSS,MSS),RR(MSS,6),A(MST,15),S(1),R(MST,4)
      DIMENSION D(MST,MTF),AA(MTF,5),MED(MTF,4),NSC(MST,1)
C
      COMMON /GEN1 / NST,NDF,NTF,NLD,NAT,NFQ,NSD,NOPT,NRGD,NDSP,NUBC,
     *               NEGT,NSS,I3,I5
      COMMON /GEN2 / BLAB(3),ANI,ANP,RIGID
      COMMON /IHEDR/ IHED(20),IDATE(2),NPAGE(2),MAXLIN,LINE
      COMMON /FTAPE/ KTMP,KINP,KOUT,KSTR,NTAPE
      COMMON /KOPT / NINP,NOUT
      COMMON/JUNK/AI(3),AJ(3)
C
      IF (NOPT.EQ.1) GO TO 105
C
      REWIND 7
      DO 50 I=1,NSS
      DO 40 J=1,4
   40 RR(I,J)=0.
      DO 50 J=1,NSS
   50 SS(I,J)=0.
      DO 100 I=1,NST
      II=I3*(I-1)
      SS(II+1,II+1)=A(I,7+I5)
      IF (NSD.NE.0) GO TO 60
      SS(II+2,II+2)=A(I,8)
      SS(II+3,II+3)=A(I,15)
   60 DO 100 J=1,I3
      DO 100 L=1,2
      LL=5+3*L+J+I5
  100 RR(II+J,L+4)=A(I,LL)
C
  105 IF (NINP.EQ.0) GO TO 110
      LINE=99
      CALL NUPAJ (1,7,ITEST)
      WRITE (KOUT,3000)
  110 IFP=0
C
      REWIND 1
C
      DO 500 K=1,NTF
      READ  (KINP,1000) IF,IFC,X1,Y1,X2,Y2,(MED(K,J),J=1,4)
      IF (NINP.EQ.0) GO TO 170
      CALL NUPAJ (1,1,ITEST)
      IF (ITEST.EQ.0) GO TO 160
      WRITE (KOUT,3000)
      LINE=LINE+7
  160 WRITE (KOUT,2000) K,IF,IFC,X1,Y1,X2,Y2,(MED(K,J),J=1,4)
  170 IF(IF.EQ.IFP) GO TO 150
      IF (IF.LT.IFP) CALL TABR (K,22)
      IF (IF.LE.0.OR.IF.GT.NDF) CALL TABR (K,22)
C
      IF (NOPT.NE.1)
     *READ (2) LRD,NT,(S(I),I=1,LRD),((R(I,L),I=NT,NST),L=1,4)
      IFP=IF
  150 YY=Y1-Y2
      XL=SQRT((X2-X1)*(X2-X1)+YY*YY)
      IF (XL.LT.1.E-10) CALL TABR (K,23)
      IF (NOPT.EQ.1) GO TO 500
      DO 200 N=NT,NST
      NL=NSC(N,IF)
      AR=X1*(Y2-A(NL,6))+X2*(A(NL,6)-Y1)+A(NL,5)*YY
  200 D(N,K)=AR/XL
C
      AI(1)=(X2-X1)/XL
      AI(2)=(Y2-Y1)/XL
      AJ(2)=AI(2)
      AJ(1)=AI(1)
C
      KK=0
      DO 400 N=NT,NST
      AI(3)=-D(N,K)
      NL=NSC(N,IF)
      NN=I3*(NL-1)
```

```
      DO 300 L=1,4                                                      TABS80 2053
      DO 300 I=1,I3                                                     TABS80 2054
  300 RR(NN+I,L)=RR(NN+I,L)+R(N,L)*AI(I+IS)                             TABS80 2055
      DO 400 M=N,NST                                                    TABS80 2056
      ML=NSC(M,IF)                                                      TABS80 2057
      MM=I3*(ML-1)                                                      TABS80 2058
      AJ(3)=-D(M,K)                                                     TABS80 2059
      KK=KK+1                                                           TABS80 2060
      DO 400 I=1,I3                                                     TABS80 2061
      DO 400 J=1,I3                                                     TABS80 2062
  400 SS(NN+I,MM+J)=SS(NN+I,MM+J)+S(KK)*AI(I+IS)*AJ(J+IS)               TABS80 2063
C                                                                       TABS80 2064
      AA(K,1)=IF                                                        TABS80 2065
      AA(K,2)=IFC                                                       TABS80 2066
      AA(K,3)=AI(1)                                                     TABS80 2067
      AA(K,4)=AI(2)                                                     TABS80 2068
      AA(K,5)=NT                                                        TABS80 2069
  500 WRITE (1) IF,IFC,X1,Y1,X2,Y2,(HED(K,J),J=1,4)                     TABS80 2070
C                                                                       TABS80 2071
      IF (NOPT.EQ.1) GO TO 999                                          TABS80 2072
C                                                                       TABS80 2073
      DO 600 I=1,NSS                                                    TABS80 2074
      DO 600 J=1,NSS                                                    TABS80 2075
  600 SS(J,I)=SS(I,J)                                                   TABS80 2076
      REWIND 2                                                          TABS80 2077
      WRITE (2) SS                                                      TABS80 2078
      IF(IABS(NAT).EQ.1) RETURN                                         TABS80 2079
C                                                                       TABS80 2080
      WRITE (2) RR                                                      TABS80 2081
      WRITE (2) AA,D,HED                                                TABS80 2082
C                                                                       TABS80 2083
  999 RETURN                                                            TABS80 2084
C                                                                       TABS80 2085
 1000 FORMAT (2I5,4F10.0,4A5)                                           TABS80 2086
 2000 FORMAT (3I6,4F10.2,2X,4A5)                                        TABS80 2087
 3000 FORMAT (//                                                        TABS80 2088
     *        45H FRAME LOCATION DATA                        //         TABS80 2089
     *        1X5HFRAME1X5HFRAME1X5HPRINT/                              TABS80 2090
     *        4X2HNO2X4HTYPE2X4HCODE8X2HX18X2HY18X2HX28X2HY22X          TABS80 2091
     *        20H/--FRAME LOCATION--//)                                 TABS80 2092
      END                                                               TABS80 2093


      SUBROUTINE TABD (NSC,SD,R,D,RF,A,MSS,MTF,MLD,MST,HED)             TABS80 2094
C                                                                       TABS80 2095
C     CALCULATING FRAME LATERAL DISPLACEMENTS                           TABS80 2096
C                                                                       TABS80 2097
      DIMENSION SD(MST,8),R(MSS,MLD),D(MST,MTF),RF(MST,MLD),A(MTF,5)    TABS80 2098
     1  ,HED(MTF,4),NSC(MST,1)                                          TABS80 2099
C                                                                       TABS80 2100
      COMMON /GEN1 / NST,NDF,NTF,NLD,NAT,NFQ,NSD,NOPT,NRGD,NDSP,NURC,   TABS80 2101
     *                NEGT,NSS,I3,IS                                    TABS80 2102
      COMMON /GEN2 / RLAB(3),ANI,ANP,RIGID                              TABS80 2103
      COMMON/JUNK/ AI(3)                                                TABS80 2104
C                                                                       TABS80 2105
      IF (NOPT.EQ.1) GO TO 999                                          TABS80 2106
C                                                                       TABS80 2107
      READ (2) A,D,HED                                                  TABS80 2108
      REWIND 2                                                          TABS80 2109
C                                                                       TABS80 2110
      DO 500 K=1,NTF                                                    TABS80 2111
      I=NTF+1-K                                                         TABS80 2112
      IF=A(I,1)                                                         TABS80 2113
      IFC=A(I,2)                                                        TABS80 2114
      AI(1)=A(I,3)                                                      TABS80 2115
      AI(2)=A(I,4)                                                      TABS80 2116
      NT=A(I,5)                                                         TABS80 2117
C                                                                       TABS80 2118
      DO 300 L=1,NLD                                                    TABS80 2119
      DO 300 N=NT,NST                                                   TABS80 2120
      NL=NSC(N,IF)                                                      TABS80 2121
      NN=I3*(NL-1)                                                      TABS80 2122
      AI(3)=-D(N,I)                                                     TABS80 2123
      DD=0.                                                             TABS80 2124
C                                                                       TABS80 2125
      DO 200 J=1,I3                                                     TABS80 2126
  200 DD=DD+R(NN+J,L)*AI(J+IS)                                          TABS80 2127
C                                                                       TABS80 2128
  300 RF(N,L)=DD                                                        TABS80 2129
C                                                                       TABS80 2130
      NS=NST-NT+1                                                       TABS80 2131
      WRITE (2) IF,IFC,NS,((RF(N,L),N=NT,NST),L=1,NLD),(HED(I,J),J=1,4) TABS80 2132
  500 CONTINUE                                                          TABS80 2133
C                                                                       TABS80 2134
  999 RETURN                                                            TABS80 2135
      END                                                               TABS80 2136


      SUBROUTINE TAB[illegible] (A,F,XM,W,IQ,S,PA,MSS,MST)              TABS80 2137
C                                                                       TABS80 2138
C     THIS IS WHERE ALL THE STRUCTURAL DYNAMICS TAKES PLACE             TABS80 2139
C                                                                       TABS80 2140
      DIMENSION F(MSS,MSS),S(MSS,MSS),XM(1),W(1),IQ(1),A(MST,4),PA(2,1) TABS80 2141
C                                                                       TABS80 2142
      COMMON /GEN1 / NST,NDF,NTF,NLD,NAT,NFQ,NSD,NOPT,NRGD,NDSP,NURC,   TABS80 2143
     *                NEGT,NSS,I3,IS                                    TABS80 2144
```

```
      COMMON /GEN2 / RLAB(3),ANI,ANP,RIGID
      COMMON /FTAPE/ KINP,KINP,KOUT,KSTR,NTAPE
      COMMON /IHEDR/ IHED(28),IDATE(2),NPAGE(2),MAXLIN,LINE
      COMMON /KOPT / NINP,NOUT
      COMMON /UBC  / Z,TS,UBCI,GRAV,PERIOD(2),UBCK(2),NTOP(2),NBOT(2)
      COMMON /DYN  / NTIPE,DT,NPC,DAMP,NMD,IHTYP,HDT
      COMMON /JUNK/ SC(2),SFMT(10),PF(3),ITP(3),TP(3),SHED(10)
C
      NMD=0
C
      REWIND 2
      IF (NOPT.NE.1) READ (2) S
C
      IF (NUBC.EQ.0) GO TO 920
      DO 980 K=1,13
      ITP(K)=0
  980 TP(K)=0.
C
  920 IF (NAT.EQ.0) GO TO 700
      IF (NOPT.EQ.1) GO TO 556
C
      TUPI=8.0*ATAN(1.0)
C
      DO 100 I=1,NST
      IF (A(I,3).LE..0) CALL TABR (NST-I+1,24)
      II=I3*(I-1)+1
      XM(II   )=A(I,3)
      IF (I3.EQ.1) GO TO 100
      XM(II+1)=A(I,3)
      XM(II+2)=A(I,4)
      IF (A(I,4).LE..0) CALL TABR (NST-I+1,25)
  100 CONTINUE
C
      DO 150 I=1,NSS
  150 XM(I)=1.0/SQRT(XM(I))
      DO 200 I=1,NSS
      DO 200 J=1,NSS
  200 S(I,J)=S(I,J)*XM(I)*XM(J)
C
      CALL TABH (S,NSS,O,F,NR,W,10)
C
      DO 250 I=1,NSS
      W(I)=S(I,I)
      DO 250 J=1,NSS
  250 F(I,J)=F(I,J)*XM(I)
      DO 300 I=1,NFO
      WM=W(I)
      DO 270 J=I,NSS
      IF(W(J).GT.WM) GO TO 270
      WM=W(J)
      K=J
  270 CONTINUE
C
      W(K)=W(I)
      W(I)=TUPI/SQRT(WM)
      DO 300 J=1,NSS
      WM=F(J,K)
      F(J,K)=F(J,I)
  300 F(J,I)=WM
C
      IF (NOUT.EQ.0) GO TO 810
      LINE=99
      CALL NUPAJ (1,4,ITEST)
      WRITE (KOUT,4001)
      DO 530 I=1,NFO
      CALL NUPAJ (1,1,ITEST)
      IF (ITEST.EQ.0) GO TO 530
      WRITE (KOUT,4001)
      LINE=LINE+4
  530 WRITE (KOUT,2001) I,W(I)
C
      DO 600 I=1,NFO,4
      IH=I+3
      IF(IH.GT.NFO) IH=NFO
      CALL NUPAJ (1,5+1*I3,ITEST)
      LINE=LINE-1-I3
      WRITE (KOUT,2003) (L,L=I,IH)
      DO 600 N=1,NST
      LN=NST+1-N
      NN=I3*(N-1)
      WRITE (KOUT,2002)
      CALL NUPAJ (1,I+I3,ITEST)
      IF (ITEST.EQ.0) GO TO 605
      LINE=LINE+5
      WRITE (KOUT,2003) (L,L=I,IH)
  605 DO 600 J=1,I3
  600 WRITE (KOUT,2004)   A(N,1),RLAB(J+I5),( F(NN+J,K),K=I,IH)
C
C     REGENERATING THE MASS MATRIX
C
  810 DO 101 I=1,NST
      II=I3*(I-1)+1
      XM(II   )=A(I,3)
      IF (I3.EQ.1) GO TO 101
      XM(II+1)=A(I,3)
      XM(II+2)=A(I,4)
  101 CONTINUE
C
C     MODAL PARTICIPATION FACTORS
C
```

```
      IF (NOUT.EQ.0) GO TO 815
      CALL NUPAJ (1,2*I3+1,ITEST)
      LINE=LINE-I3-1
      WRITE (KOUT,3001)
      LINE=LINE-1
  815 DO 850 I=1,NFQ
      DO 851 K=1,I3
  851 PF(K)=0.0
      DO 750 J=1,NST
      JJ=I3*(J-1)
      DO 751 K=1,I3
  751 PF(K)=PF(K)+F(JJ+K,I)*XM(JJ+K)
  750 CONTINUE
C
      IF (NUBC.EQ.0) GO TO 550
      DO 540 K=1,I3
      WM=PF(K)*PF(K)
      IF (WM.LT.TP(K)) GO TO 540
      TP(K)=WP
      ITP(K)=I
  540 CONTINUE
C
  550 IF (NOUT.EQ.0) GO TO 850
      CALL NUPAJ (1,1+I3,ITEST)
      IF (ITEST.EQ.0) GO TO 852
      WRITE (KOUT,3001)
      LINE=LINE+7
  852 WRITE (KOUT,3002) I,(RLAB(K+IS),PF(K),K=1,I3)
      WRITE (KOUT,2002)
  850 CONTINUE
C
      IF (NUBC.EQ.0) GO TO 560
      JJ=I3
      IF (JJ.EQ.3) JJ=2
      DO 555 K=1,JJ
      IF (PERIOD(K+IS).GT..0) GO TO 555
      I=ITP(K)
      PERIOD (K+IS)=W(I)
  555 CONTINUE
C
  556 IF (NUBC.EQ.0) GO TO 560
      IF (NINP.EQ.0) GO TO 560
      LINE=99
      CALL NUPAJ (1,1,ITEST)
      WRITE (KOUT,3005) 2,TS,URCI,GRAV,
     *                  (NTOP(I),NROT(I),PERIOD(I),URCK(I),I=1,2)
C
  560 IF(NAT.EQ.1) RETURN
C
C     TIME HISTORY ANALYSIS
C
      IF (NAT.NE.4) GO TO 440
      READ  (KINP,9001) SHED,SFMT,NPC,NTIME,SF,FI,DT,IHTYP,HDT
C
      IF (IHTYP.EQ.1H ) IHTYP=1HU
      DO 909 K=1,10
      IF (SFMT(K).NE.4H    ) GO TO 906
  909 CONTINUE
      SFMT(1)=4H(2F1
      SFMT(2)=4H0.0)
      IF (IHTYP.EQ.1HU) GO TO 906
      SFMT(1)=4H(8F9
      SFMT(2)=4H.0)
  906 CONTINUE
C
      IF (NINP.EQ.0) GO TO 720
      LINE=99
      CALL NUPAJ (1,16,ITEST)
      WRITE (KOUT,9002) SHED,SFMT,NPC,NTIME,SF,FI,DT,IHTYP,HDT
      GO TO 720
C
C     RESPONSE SPECTRUM ANALYSIS
C
  440 IF(NAT.NE.3) GO TO 700
  701 READ  (KINP,1000) NPC,NMD,SF,FI,DAMP,SHED,
     *                  (PA(1,I),PA(2,I),I=1,NPC)
      IF (DAPP.GT.1.) DAPP=.02
      IF (DAPP.LE..0) DAPP=.02
      IF (NMD.GT.NFQ) NMD=NFQ
      IF (NMD.LT.0) NMD=0
      IF (NINP.EQ.0) GO TO 720
      LINE=99
      CALL NUPAJ (1,9,ITEST)
      WRITE (KOUT,2000) SHED,NPC,NMD,SF,FI,DAPP
      IF (NPC.GT.200) CALL TARR (NPC,26)
      WRITE (KOUT,3003)
      LINE=LINE+3
      DO 710 I=1,NPC
      CALL NUPAJ (1,1,ITEST)
      IF (ITEST.EQ.0) GO TO 710
      WRITE (KOUT,3003)
      LINE=LINE+3
  710 WRITE (KOUT,3004) PA(1,I),PA(2,I)
C
  720 IF (NOPT.EQ.1) GO TO 704
      FI=FI*ATAN(1.0)/45.
      SC(1)=SIN(FI)
      SC(2)=COS(FI)
C
      IF (NAT.NE.3) GO TO 730
```

```
      IF (NINP.EQ.0) GO TO 730
      CALL NUPAJ (1,7,ITEST)
      LINE=LINE-1
      WRITE (KOUT,2009)
  730 DO 500 I=1,NFQ
      RLM=0.
C
      DO 450 J=1,NST
      JJ=13*(J-1)+1
      FR=F(JJ,I)*SC(1+IS)
      IF(NSD.EQ.0) FR=FR+F(JJ+1,I)*SC(2)
  450 RLM=RLM+FR*A(J,3)
C
      WW=(TUPI/W(I))**2
      YM=RLM*SF
      YMA=0.
      IF (NAT.NE.3) GO TO 740
      WM=TABA(NPC,SF,W(I),PA)
      YMA=RLM*WM
      YM=YMA/WW
C
      IF (NINP.EQ.0) GO TO 740
      CALL NUPAJ (1,1,ITEST)
      IF (ITEST.EQ.0) GO TO 495
      LINE=LINE+6
      WRITE (KOUT,2009)
  495 WRITE (KOUT,2008) I,WM
C
  740 DO 500 J=1,NSS
      S(J,I)=F(J,I)*YMA*XM(J)
  500 F(J,I)=F(J,I)*YM
      IF (NAT.NE.3) GO TO 704
C
C     WRITE PERIODS ON TAPE 1 FOR CQC COMBINATION
C
      REWIND 1
      WRITE (1) (W(I),I=1,NFQ)
C
C     MAXIMUM MODAL INERTIA FORCES/MOMENTS GENERATED
C
      IF (NOUT.EQ.0) GO TO 702
      LINE=99
      DO 800 I=1,NFQ,4
      IM=I+3
      IF (IM.GT.NFQ) IM=NFQ
      CALL NUPAJ (1,7,ITEST)
      LINE=LINE-1
      WRITE (KOUT,2005) (L,L=I,IM)
      DO 800 N=1,NST
      LN=NST+1-N
      NN=13*(N-1)
      WRITE (6,2002)
      CALL NUPAJ (1,1+13,ITEST)
      IF (ITEST.EQ.0) GO TO 805
      LINE=LINE+6
      WRITE (KOUT,2005) (L,L=I,IM)
  805 DO 800 J=1,13
  800 WRITE (6,2006) A(N,1),RLAB(J+IS),(S(NN+J,K),K=I,IM)
C
C     MAXIMUM MODAL CUMULATIVE INERTIA FORCES/MOMENTS GENERATED
C
      IF (NST.EQ.1) GO TO 702
      DO 501 I=1,NFQ
      DO 501 J=2,NST
      II=13*(J-1)
      DO 501 K=1,13
  501 S(II+K,I)=S(II+K,I)+S(II+K-13,I)
C
      JJ=13
      IF (JJ.EQ.3) JJ=2
      LINE=99
      DO 900 I=1,NFQ,4
      IM=I+3
      IF (IM.GT.NFQ) IM=NFQ
      CALL NUPAJ (1,6,ITAG)
      LINE=LINE-1
      WRITE (KOUT,2007) (L,L=I,IM)
      DO 900 N=1,NST
      LN=NST+1-N
      NN=13*(N-1)
      WRITE (6,2002)
      CALL NUPAJ (1,1+JJ,ITEST)
      IF (ITEST.EQ.0) GO TO 905
      LINE=LINE+5
      WRITE (KOUT,2007) (L,L=I,IM)
  905 DO 900 J=1,JJ
  900 WRITE (6,2006)    A(N,1),RLAB(J+IS),(S(NN+J,K),K=I,IM)
C
  702 CONTINUE
C
C     SAVE MODAL DATA ON TAPE 1 FOR TIME HISTORY RESPONSE
C
  704 IF (NAT.NE.4) GO TO 705
C
      REWIND 1
      LINE=99
C
      DO 707 K=1,NFQ
      READ (KINP,9004) I,DAMP
      IF (DAPP.GT.1.) CALL TABR (K,27)
```

```
      IF (DAMP.LT.0.) CALL TABR (K,28)
C
      IF (NINP.EQ.0) GO TO 709
      CALL NUPAJ (1,1,ITEST)
      IF (ITEST.EQ.0) GO TO 708
      WRITE (KOUT,9003)
      LINE=LINE+3
  708 WRITE (KOUT,9005) K,DAMP
C
  709 IF (NOPT.EQ.1) GO TO 707
      WRITE (1) W(K),DAMP,(F(I,K),I=1,NSS)
  707 CONTINUE
C
  705 IF (NOPT.EQ.1) GO TO 999
      DO 650 K=1,NFQ
      I=NFQ+1-K
      DO 650 J=1,NSS
  650 F(J,I+6)=F(J,I)
C
  700 IF (NUPT.EQ.1) GO TO 999
      IF (NAT.EQ.0) GO TO 910
C
C     REINSTATE STIFFNESS MATRIX
C
      REWIND 2
      READ (2) S
C
  910 READ (2) ((F(I,J),I=1,NSS),J=1,6)
      IF (NUBC.EQ.0) GO TO 950
      CALL TUBC (F(1,5),A,XM,NSS,NST)
C
C     SOLVE FOR STATIC LOAD CONDITIONS
C
  950 CALL TABO (S,NSS,F,6)
C
C     PRINT STRUCTURE DISPLACEMENTS
C
      LINE=99
      CALL NUPAJ (1,7,ITEST)
      WRITE (KOUT,5001)
      DO 960 I=1,NST
      LN=NST+I-1
      WRITE (KOUT,5003)
      CALL NUPAJ (1,1+13,ITEST)
      IF (ITEST.EQ.0) GO TO 970
      WRITE (KOUT,5001)
      LINE=LINE+7
  970 DO 960 J=1,13
      II=13*(I-1)+J
  960 WRITE (6,5002) A(I,1),RLAB(J+IS),(F(II,L),L=1,6)
```

```
C
  999 RETURN
C
 1000 FORMAT (2I5,3F10.0,10A3/(2F10.0))
 2000 FORMAT (//  22H ACCELERATION SPECTRUM ,5X,10A3//
     1 25H NO OF PERIOD CARDS       = I10/
     1 25H NO OF SEPARATE MODES     = I10/
     2 25H SCALE FACTOR             = F10.3/
     3 25H ANGLE OF EQ INCIDENCE    = F10.3/
     3 25H STRUCTURAL DAMPING       = F10.3)
 2001 FORMAT (I7,F15.4)
 2002 FORMAT (1H )
 2003 FORMAT (// 12H MODE SHAPES/ /18H        LEVEL  DIRN ,8I13)
 2004 FORMAT (5X,2X,A5,2X,A4,2X,8F13.6)
 2005 FORMAT (//
     *          45H MAXIMUM MODAL INERTIA LOADS/TORSIONS         /
     *          45H GENERATED IN EACH LEVEL (AT CENTER OF MASS) /
     *                                   /18H        LEVEL  DIRN ,8I13)
 2006 FORMAT (5X,2X,A5,2X,A4,2X,8F13.2)
 2007 FORMAT (//
     *          45H MAXIMUM MODAL STORY SHEARS AT EACH LEVEL     /
     *                                   /18H        LEVEL  DIRN ,8I13)
 2008 FORMAT (I7,F15.3)
 2009 FORMAT (//45H GENERATED MODAL SPECTRAL ACCELERATION VALUES//
     *          3X4HMODE7X8HSPECTRAL/1X6HNUMBER3X12HACCELERATION/)
 3001 FORMAT (//45H MODAL PARTICIPATION FACTORS                   //
     *          3X4HMODE2X8HP-FACTOR7X8HP-FACTOR/
     *          1X6HNUMBER1X9HDIRECTION10X5HVALUE/)
 3002 FORMAT (I7,6X,A4,F15.5,/(13X,A4,F15.5))
 3003 FORMAT (//
     4 26H  PERIOD       ACCELERATION )
 3004 FORMAT (F10.3,5X,F10.3)
 3005 FORMAT (//
     *          45H STATIC SEISMIC LOAD CALCULATION DATA . . .  /
     *          45H UBC 1976 (SEAOC CODE)                      //
     *          45H UBC ZONE FACTOR (Z)-------------------------F10.2/
     *          45H PREDOMINANT SOIL PERIOD (TS)----------------F10.2/
     *          45H UBC IMPORTANCE FACTOR (I)-------------------F10.2/
     *          45H GRAVITATIONAL ACCELERATION------------------F10.2///
     *          45H LOAD CONDITION A (X-DIRECTION) . . .        //
     *          45H TOP    LEVEL OF TRIANGULAR DISTRIBUTION-----I10/
     *          45H BOTTOM LEVEL OF TRIANGULAR DISTRIBUTION-----I10/
     *          45H PERIOD OF PREDOMINANT X STRUCTURAL MODE-----F10.3/
     *          45H UBC STRUCTURAL SYSTEM FACTOR----------------F10.3///
     *          45H LOAD CONDITION B (Y-DIRECTION) . . .        //
     *          45H TOP    LEVEL OF TRIANGULAR DISTRIBUTION-----I10/
     *          45H BOTTOM LEVEL OF TRIANGULAR DISTRIBUTION-----I10/
     *          45H PERIOD OF PREDOMINANT Y STRUCTURAL MODE-----F10.3/
     *          45H UBC STRUCTURAL SYSTEM FACTOR----------------F10.3)
 4001 FORMAT (//
```

```
     *         3X4HMODE11X4HTIME/1X6HNUMBER9X6HPERIOD)
5001 FORMAT (//
     *         45H STATIC LOAD CONDITION DISPLACEMENTS          //
     *         45H DISPLACEMENTS ARE AT THE CENTERS OF MASS OF
     *         45HTHE RESPECTIVE LEVELS                         //
     *         15X,2X1H/21(1H-),16HLOAD  CONDITIONS,21(1H-)1H//
     *         3X5HLEVEL3X4HDIRN11X1HI8X2HII7X3HIII8X2HIV9X1HA9X1HB/)
5002 FORMAT (3X,A5,2X,A5,2X,6F10.5)
5003 FORMAT (1H )
9001 FORMAT (10A3,10A4,/2I5,3F10.0,9X,A1,F10.0)
9002 FORMAT (//23H RESPONSE ANALYSIS DATA  ///
     . 31H ACCELERATION HISTORY HEADING..  10A3//
     1 30H ACCELERATION INPUT FORMAT      10A4/
     1 30H NUMBER OF POINTS ON HISTORY    I10/
     2 30H NUMBER OF OUTPUT TIMES         I10/
     3 30H ACCELERATION SCALE FACTOR      F10.4/
     4 30H ANGLE OF EQ INCIDENCE          F10.4/
     5 30H TIME INCREMENT FOR OUTPUT      F10.4/
     5 30H TIME HISTORY TYPE              9X,A1/
     5 30H TIME STEP IF E-TYPE HISTORY    F10.4)
9003 FORMAT( // 16H MODE    DAMPING )
9004 FORMAT (I5,F10.2)
9005 FORMAT (I4,F12.3)
9006 FORMAT (10A4)
     END


      SUBROUTINE TUBC (F,SD,TEMP,MSS,MST)
C
C     REDEFINING LOAD CONDITIONS A AND B WITH UBC LOADS
C
      DIMENSION F(MSS,2),SD(MST,19),TEMP(MST)
C
      COMMON /GEN1 / NST,NDF,NTF,NLD,NAT,NFQ,NSD,NOPT,NRGD,NDSP,NUBC,
     *               NEGT,NSS,I3,IS
      COMMON /GEN2 / RLAB(3),ANI,ANP,RIGID
      COMMON /IHEDR/ IHED(28),IDATE(2),NPAGE(2),MAXLIN,LINE
      COMMON /FTAPE/ KTMP,KINP,KOUT,KSTR,NTAPE
      COMMON /KOPT / NINP,NOUT
      COMMON /UBC  / Z,TS,UBCI,GRAV,PERIOD(2),UBCK(2),NTOP(2),NBOT(2)
C
      LINE=99
      JJ=I3
      IF (JJ.EQ.3) JJ=2
C
      DO 700 K=1,JJ
C
      IF (PERIOD(K+IS).LE..0) GO TO 700
      IF (NOUT.NE.0) CALL NUPAJ (1,1,ITEST)
C
C     CALCULATE STORY ELEVATION
C
      TEMP(NST)=SD(NST,2)
      IF (NST.EQ.1) GO TO 150
      DO 100 N=2,NST
      L=NST-N+1
  100 TEMP(L)=TEMP(L+1)+SD(L,2)
C
  150 NT=NST-NTOP(K+IS)+1
      NB=NST-NBOT(K+IS)
C
C     CALCULATE STORY HEIGHT ABOVE BOTTOM OF DISTRIBUTION
C
      IF (NB.GE.NST) GO TO 250
      DO 200 N=NT,NB
  200 TEMP(N)=TEMP(N)-TEPP(NB+1)
C
  250 SUMWH=0.
      SUMW=0.
C
C     SIGMA W AND SIGMA WH
C
      DO 300 N=NT,NB
      W=SD(N,3)*GRAV
      TEMP(N)=TEMP(N)*W
      SUMW=SUMW+W
  300 SUMWH=SUMWH+TEMP(N)
C
C     SOIL FACTOR S FROM TS
C
      S=1.5
      IF (TS.LE.1.E-8) GO TO 350
C
      TT=TS
      IF (TT.LT.0.5) TT=0.5
      IF (TT.GT.2.5) TT=2.5
      TT=PERIOD(K+IS)/TT
      IF (TT.LE.1.0) GO TO 360
      GO TO 370
  360 S=1.0+TT-0.5*TT*TT
      GO TO 380
  370 S=1.2+0.6*TT-0.3*TT*TT
  380 IF (S.GT.1.5) S=1.5
      IF (S.LT.1.0) S=1.0
C
C     SEISMIC FACTOR C
C
  350 C=1/15.*SQRT(PERIOD(K+IS))
```

```
      C=1./C
C
      SC=S*C
      IF (SC.GT..14) SC=.14
C
      V=Z*SC*UBCI*UBCK(K)
      FT=0.
      IF (PERIOD(K+IS).GE..7) FT=.07*V*PERIOD(K+IS)
      IF (FT.GT..25*V) FT=.25*V
      IF (NOUT.NE.0)
     *WRITE (KOUT,1000) RLAB(K+IS),Z,S,UBCI,C,UBCK(K+IS),SUMW,V
C
      V=V*SUMW
      FT=FT*SUMW
      IF (NOUT.NE.0)
     *WRITE (KOUT,2000) V,FT
C
      V=V-FT
      V=V/SUMWH
C
C     STORY FORCES
C
      DO 400 N=NT,NB
  400 TEMP(N)=TEMP(N)*V
      TEMP(NT)=TEMP(NT)+FT
C
C     PACKING INTO SD
C
      LL=8+(K+IS-1)*3
      MM=15+(K+IS-1)*2
      DO 900 N=NT,NB
      DO 950 L=1,3
  950 SD(N,LL+L)=0.
      SD(N,LL+K+IS)=TEMP(N)
  900 SD(N,LL+3)=SD(N,LL+1)*(SD(N,MM+2)-SD(N,6))
     *              -SD(N,LL+2)*(SD(N,MM+1)-SD(N,5))
C
  700 CONTINUE
C
C     LOADING INTO STRUCTURAL LOAD VECTOR
C
      DO 800 I=1,NST
      II=13*(I-1)
      DO 800 J=1,13
      DO 800 L=1,2
      LL=5+3*L+J+IS
  800 F(II+J,L)=SD(I,LL)
C
C     PRINTING MODIFIED LOAD CONDITIONS A AND B
C
      IF (NOUT.EQ.0) GO TO 999
      LINE=99
      CALL NUPAJ (1,4,ITEST)
      WRITE (KOUT,4000)
C
      CALL NUPAJ (1,6,ITEST)
      LINE=LINE-1
      WRITE (KOUT,5000)
      DO 500 N=1,NST
      CALL NUPAJ (1,1,ITEST)
      IF (ITEST.EQ.0) GO TO 500
      WRITE (KOUT,5000)
      LINE=LINE+5
  500 WRITE (KOUT,5001) SD(N,1),(SD(N,I),I=9,10),(SD(N,I),I=16,17)
C
      CALL NUPAJ (1,6,ITEST)
      LINE=LINE-1
      WRITE (KOUT,6000)
      DO 600 N=1,NST
      CALL NUPAJ (1,1,ITEST)
      IF (ITEST.EQ.0) GO TO 600
      WRITE (KOUT,6000)
      LINE=LINE+5
  600 WRITE (KOUT,6001) SD(N,1),(SD(N,I),I=12,13),(SD(N,I),I=18,19)
C
  999 RETURN
C
 1000 FORMAT (///
     *        50H UNIFORM BUILDING CODE SEISMIC LOADS FOR DIRECTION A4//
     *        45H V = Z SICK W              SC=0.14 MAX          //
     *        4H Z =F10.4//
     *        4H S =F10.4/
     *        4H I =F10.4/
     *        4H C =F10.4/
     *        4H K =F10.4//
     *        4H W =F10.4//
     *        4H V =F10.4,1HW)
 2000 FORMAT (4H    =F10.4,  //4H FT=,F10.4)
 4000 FORMAT (//
     *        45H STRUCTURAL LATERAL LOAD CONDITIONS            /
     *        45H AS ADJUSTED BY UBC SEISMIC REQUIREMENTS       )
 5000 FORMAT (//
     *        55H STRUCTURAL LATERAL LOAD CONDITION A (X-DIRECTION). . .
     *                                                              //
     *        6H LEVEL10X2HFX10X2HFY11X1HX11X1HY)
 6000 FORMAT (//
     *        55H STRUCTURAL LATERAL LOAD CONDITION B (Y-DIRECTION). . .
     *                                                              //
     *        6H LEVEL10X2HFX10X2HFY11X1HX11X1HY)
 5001 FORMAT (1X,A5,4F12.2)
```

```
 6001 FORMAT (1X,A5,4F12.2)
      END

      SUBROUTINE TABH (H,N,IEGEN,U,NR,X,IQ)
C
      DIMENSION H(N,N),U(N,N),X(N),IQ(N)
C
      IF (IEGEN) 15,10,15
 10   DO 14 I=1,N
      DO 14 J=1,N
      IF(I-J) 12,11,12
 11   U(I,J)=1.0
      GO TO 14
 12   U(I,J)=0.
 14   CONTINUE
C
 15   NR = 0
      IF (N-1) 1000,1000,17
C
C     SCAN FOR LARGEST OFF DIAGONAL ELEMENT IN EACH ROW
C     X(I) CONTAINS LARGEST ELEMENT IN ITH ROW
C     IQ(I) HOLDS SECOND SUBSCRIPT DEFINING POSITION OF ELEMENT
C
 17   NMI1=N-1
      DO 30 I=1,NMI1
      X(I) = 0.
      IPL1=I+1
      DO 30 J=IPL1,N
      IF ( X(I) - ABS( H(I,J))) 20,20,30
 20   X(I)=ABS(H(I,J))
      IQ(I)=J
 30   CONTINUE
C
C     SET INDICATOR FOR SHUT-OFF.RAP=2**-27,NR=NO. OF ROTATIONS
      RAP=7.450580596E-9
      HDTEST=1.0E38
C
C     FIND MAXIMUM OF X(I) S FOR PIVOT ELEMENT AND
C     TEST FOR END OF PROBLEM
C
 40   DO  70  I=1,NMI1
      IF (I-1) 60,60,45
 45   IF ( XMAX- X(I)) 60,70,70
 60   XMAX=X(I)
      IPIV=I
      JPIV=IQ(I)
 70   CONTINUE
C
C     IS MAX. X(I) EQUAL TO ZERO. IF LESS THAN HDTEST, REVISE HDTEST
      IF ( XMAX) 1000,1000,80
 80   IF (HDTEST) 90,90,85
 85   IF (XMAX - HDTEST) 90,90,148
 90   HDIMIN = ABS( H(1,1) )
      DO 110  I= 2,N
      IF (HDIMIN- ABS( H(I,I))) 110,110,100
 100  HDIMIN=ABS(H(I,I))
 110  CONTINUE
C
      HDTEST=HDIMIN*RAP
C
C     RETURN IF MAX.H(I,J)LESS THAN(2**-27)ABSF(H(K,K)-MIN)
      IF (HDTEST- XMAX) 148,1000,1000
 148  NR = NR+1
C
C     COMPUTE TANGENT, SINE AND COSINE,H(I,I),H(J,J)
 150  TANG=SIGN(2.0,(H(IPIV,IPIV)-H(JPIV,JPIV)))*H(IPIV,JPIV)/(ABS(H(I
     1PIV,IPIV)-H(JPIV,JPIV))+SQRT((H(IPIV,IPIV)-H(JPIV,JPIV))**2+4.0*H
     2(IPIV,JPIV)**2))
      COSINE=1.0/SQRT(1.0+TANG**2)
      SINE=TANG*COSINE
      HII=H(IPIV,IPIV)
      H(IPIV,IPIV)=COSINE**2*(HII+TANG*(2.*H(IPIV,JPIV)+TANG*H(JPIV,JPIV
     1)))
      H(JPIV,JPIV)=COSINE**2*(H(JPIV,JPIV)-TANG*(2.*H(IPIV,JPIV)-TANG*H
     1II))
      H(IPIV,JPIV)=0.
C
C      PSEUDO RANK THE EIGENVALUES
C      ADJUST SINE AND COS FOR COMPUTATION OF H(IK) AND U(IK)
      IF ( H(IPIV,IPIV) -  H(JPIV,JPIV)) 152,153,153
 152  HTEMP = H(IPIV,IPIV)
      H(IPIV,IPIV) = H(JPIV,JPIV)
      H(JPIV,JPIV) = HTEMP
C       RECOMPUTE SINE AND COS
      HTEMP = SIGN (1.0, -SINE) * COSINE
      COSINE = ABS (SINE)
      SINE = HTEMP
 153  CONTINUE
C
C     INSPECT THE IQS BETWEEN I+1 AND N-1 TO DETERMINE
C     WHETHER A NEW MAXIMUM VALUE SHOULD BE COMPUTED SINCE
C     THE PRESENT MAXIMUM IS IN THE I OR J ROW.
C
      DO 350 I=1,NMI1
      IF(I-IPIV)210,350,200
 200  IF(I-JPIV)210,350,210
 210  IF(IQ(I)-IPIV)230,240,230
```

```
  230 IF(IQ(I)-JPIV)350,240,350
  240 K=IQ(I)
  250 HTEMP=H(I,K)
      H(I,K)=0.
      IPL1=I+1
      X(I) =0.
C
C     SEARCH IN DEPLETED ROW FOR NEW MAXIMUM
C
      DO 320 J=IPL1,N
      IF ( X(I)- ABS( H(I,J)) ) 300,300,320
  300 X(I) = ABS(H(I,J))
      IQ(I)=J
  320 CONTINUE
      H(I,K)=HTEMP
  350 CONTINUE
C
      X(IPIV) =0.
      X(JPIV) =0.
C
C     CHANGE THE OTHER ELEMENTS OF H
C
      DO 530 I=1,N
C
      IF(I-IPIV)370,530,420
  370 HTEMP = H(I,IPIV)
      H(I,IPIV) = COSINE*HTEMP + SINE*H(I,JPIV)
      IF ( X(I) -  ABS( H(I,IPIV)) )380,390,390
  380 X(I) = ABS(H(I,IPIV))
      IQ(I) = IPIV
  390 H(I,JPIV) = -SINE*HTEMP + COSINE*H(I,JPIV)
      IF ( X(I) -  ABS( H(I,JPIV)) ) 400,530,530
  400 X(I) = ABS(H(I,JPIV))
      IQ(I) = JPIV
      GO TO 530
C
  420 IF(I-JPIV)430,530,480
  430 HTEMP = H(IPIV,I)
      H(IPIV,I) = COSINE*HTEMP + SINE*H(I,JPIV)
      IF ( X(IPIV) -  ABS( H(IPIV,I)) ) 440,450,450
  440 X(IPIV) = ABS(H(IPIV,I))
      IQ(IPIV) = I
  450 H(I,JPIV) = -SINE*HTEMP + COSINE*H(I,JPIV)
      IF ( X(I) -  ABS( H(I,JPIV)) ) 400,530,530
C
  480 HTEMP = H(IPIV,I)
      H(IPIV,I) = COSINE*HTEMP + SINE*H(JPIV,I)
      IF ( X(IPIV) -  ABS( H(IPIV,I)) ) 490,500,500
  490 X(IPIV) = ABS(H(IPIV,I))
      IQ(IPIV) = I
  500 H(JPIV,I) = -SINE*HTEMP + COSINE*H(JPIV,I)
      IF ( X(JPIV) -  ABS( H(JPIV,I)) ) 510,530,530
  510 X(JPIV) = ABS(H(JPIV,I))
      IQ(JPIV) = I
  530 CONTINUE
C
C     TEST FOR COMPUTATION OF EIGENVECTORS
C
      IF(IEGEN)40,540,40
  540 DO 550 I=1,N
      HTEMP=U(I,IPIV)
      U(I,IPIV)=COSINE*HTEMP+SINE*U(I,JPIV)
  550 U(I,JPIV)=-SINE*HTEMP+COSINE*U(I,JPIV)
      GO TO 40
 1000 RETURN
      END


      FUNCTION   TABA (NPC,SF,T,PA)
C
C     RESPONSE SPECTRUM LINEAR INTERPOLATER
C
      DIMENSION PA(2,NPC)
C
      DO 100 I=2,NPC
      T1=PA(1,I-1)
      T2=PA(1,I  )
      IF (T2.LE.T1) CALL TABR (1,29)
      IF (T.LE.T2.AND.T.GE.T1) GO TO 200
  100 CONTINUE
C
      IF (T.GT.PA(1,NPC)) TABA=PA(2,NPC)
      IF (T.LT.PA(1,1)) TABA=PA(2,1)
      GO TO 999
C
  200 R1=(T2-T)/(T2-T1)
      R2=(T-T1)/(T2-T1)
      TABA=SF*(PA(2,I-1)*R1+PA(2,I)*R2)
C
  999 RETURN
C
      END


      FUNCTION COC(Y,RO,NFQ)
C
```

```
C     PERFORMING CQC COMBINATION
C
      DIMENSION V(NFQ),RO(NFQ,NFQ)
C
      CQC=0.
C
      DO 20 J=1,NFQ
      TEMP=0.
      DO 10 I=1,NFQ
   10 TEMP=TEMP+RO(I,J)*V(I)
   20 CQC=CQC+TEMP*V(J)
C
      CQC=ABS(CQC)
      CQC=SQRT(CQC)
C
      RETURN
      END


      SUBROUTINE CQCIN (W,RO,DAMP,NFQ)
C
C     INITIALIZING MODAL CROSS CORRELATION MATRIX FOR CQC
C
      DIMENSION W(NFQ),RO(NFQ,NFQ)
C
      TUPI=8.*ATAN(1.0)
C
      DO 10 I=1,NFQ
      W(I)=TUPI/W(I)
   10 CONTINUE
C
      DO 70 I=1,NFQ
      DO 20 J=1,NFQ
C
      IF (I-J) 40,50,40
   50 RO(I,J)=1.
      GO TO 20
C
   40 IF(DAMP) 30,30,60
   30 RO(I,J)=0.
      GO TO 20
C
   60 R=W(J)/W(I)
      TEMP=(1-R*R)**2+4.*DAMP*DAMP*R*((1+R)**2)
      RO(I,J)=8.*DAMP*DAMP*(1+R)*(R**1.5)/TEMP
C
   20 CONTINUE
   70 CONTINUE
C
      RETURN
      END


      SUBROUTINE TABDY(F,PA,X,T,MSS)
C
C     EVALUATION OF 3D TIME-DEPENDENT LATERAL DISPLACEMENTS
C
      DIMENSION F(MSS,1),PA(2,1),X(1),T(1)
C
      COMMON /GEN1 / NST,NDF,NTF,NLD,NAT,NFQ,NSD,NOPT
      COMMON /IHEDR/ IHED(28),IDATE(2),NPAGE(2),MAXLIN,LINE
      COMMON /FTAPE/ KTMP,KINP,KOUT,KSTR,NTAPE
      COMMON /KOPT / NINP,NOUT
      COMMON /DYN  / NTIME,DT,NPC,DAMP,NMD,IHTYP,HDT
      COMMON /JUNK / DUM(2),SFMT(10)
C
C     ZERO DISPLACEMENTS AND READ GROUND ACCELERATIONS
C
      DO 100 I=1,MSS
      DO 100 K=1,NTIME
  100 F(I,K+6)=0.0
C
      IF (IHTYP.EQ.IME) GO TO 400
      READ  (KINP,SFMT) (PA(1,I),PA(2,I),I=1,NPC)
      GO TO 450
  400 READ  (KINP,SFMT) (PA(2,I),I=1,NPC)
      TIME=0.
      DO 420 I=1,NPC
      PA(1,I)=TIME
  420 TIME=TIME+HDT
  450 CONTINUE
C
      IF (NINP.EQ.0) GO TO 115
      LINE=99
      DO 130 I=1,NPC
      CALL NUPAJ (1,1,ITEST)
      IF (ITEST.EQ.0) GO TO 130
      WRITE (KOUT,2000)
      LINE=LINE+6
  130 WRITE (KOUT,4000) PA(1,I),PA(2,I)
C
C     CHECK GROUND ACCELERATION DATA
C
  115 DO 120 I=2,NPC
      IF (PA(1,I).GT.PA(1,I-1)) GO TO 120
      CALL TABR(I,29)
```

```
  120 CONTINUE
C
      TIME=FLOAT(NTIME)
      TIME=TIME*DT
      ATIME=PA(1,NPC)-PA(1,1)
      IF(ATIME.GE.TIME) GO TO 150
      CALL TABR (1,30)
C
C     FOR EACH MODE CALCULATE RESPONSE AND TOTAL DISPLACEMENTS
C
  150 REWIND 1
      IF (NOPT.EQ.1) GO TO 999
      DO 300 L=1,NFQ
      READ (1) W,DAMP,(T(K),K=1,MSS)
      CALL TABRI (PA,X,W,NTIME,NPC,DT,DAMP)
      DO 200 I=1,MSS
      DO 200 J=1,NTIME
  200 F(I,J+6)=F(I,J+6)+T(I)*X(J)
  300 CONTINUE
C
  999 RETURN
C
 2000 FORMAT (//26H GROUND ACCELERATION CARDS //
     1          11X,4HTIME,3X,12HACCELERATION  )
 4000 FORMAT (F15.4,F15.3)
      END


      SUBROUTINE TABRI(PA,X,PERD,NTIME,NPC,DDT,DAMP)
      DIMENSION PA(2,1),X(1)
C
C     EVALUATION OF MODAL RESPONSE ... USING EXPLICIT INTEGRATION
C
      W=8.0*ATAN(1.0)/PERD
      WW=W*W
      ZW=DAMP*W
      TZW=2.*ZW
      WD=W*SQRT(1.0-DAMP**2)
      FB=TZW/(WW*WW)
      FA=ZW/WW
      FV=ZW*WW
      FVD=WW*(2.*DAMP**2-1.0)
      FBB=(2.*DAMP**2-1.0)/WW

      L=0
      II=1
      VO=0.
      VDO=0.
      TO=PA(1,1)
   10 DT=DDT
   50 B=(PA(2,II+1)-PA(2,II))/(PA(1,II+1)-PA(1,II))
      A=PA(2,II)+B*(TO-PA(1,II))
      TT=TO+DT
      IF(PA(1,II+1).GT.TT) GO TO 100
      DELT=PA(1,II+1)-TO
      GO TO 200
  100 DELT=DT
C
  200 EX=EXP(-ZW*DELT)
      FT=WD*DELT
      CS=COS(FT)
      SN=SIN(FT)
      VT=(VDO+ZW*VO-FA*A+FBB*B)*SN/WD
      VT=VT+(VO-A/WW+FB*B)*CS
      VT=VT*EX+A/WW-FB*B+B*DELT/WW
C
      VDT=(A-WW*VO-ZW*(VDO+B/WW))*SN/WD
      VDT=EX*((VDO-B/WW)*CS+VDT)+B/WW
C
      VDDT=(B+FV*VO+FVD*VDO-ZW*A)/WD
      VDDT=EX*((A-WW*VO-TZW*VDO)*CS+VDDT*SN)
C
      VO=VT
      VDO=VDT

      IF(PA(1,II+1).GT.TT) GO TO 500
      DT=DT-DELT
      II=II+1
      TO=PA(1,II)
      IF(DT.EQ.0.) GO TO 600
      GO TO 50
C
  500 TO=TO+DT
  600 L=L+1
      X(L)=VT
      IF(L.LT.NTIME) GO TO 10
C
      RETURN
      END


      SUBROUTINE TABPLT (U,NST,NTIME,DT)
C
C        PLOT FRAME DISPLACEMENT SHAPES WITH TIME , NAT=4
C
      DIMENSION U(NST,NTIME),TIME(10)
```

```
C
      COMMON /IHEDR/ IHED(28),IDATE(2),NPAGE(2),MAXLIN,LINE
      COMMON /FTAPE/ KTMP,KINP,KOUT,KSTR,NTAPE
      COMMON /KOPT / NINP,NOUT
      COMMON/JUNK/ FHED(4),DUM(2),NS,P(120)
C
      DATA VERT,STAR,HORIZ,BLK,SL/ 1HI,1H*,1H-,1H ,1H/ /
C
      IF (NOUT.EQ.0) RETURN
C
C        CONTROL VARIABLES
C
      NPL=5
      NMAX=10
      RMAX=NMAX
      ISPACE=12/(NS+3)
      TO=0.0
      NLINES=NS*(ISPACE+1)+4
      NPP=45/NLINES
C
C        FIND MAXIMUM ABSOLUTE VALUE
C
      UMAX=0.
      TMAX=0.
      DO 100 I=1,NS
      DO 100 J=1,NTIME
      UM=ABS(U(I,J))
      UU=UM-UMAX
      IF (UU) 100,100,80
   80 UMAX=UM
      TMAX=FLOAT(J)*DT
  100 CONTINUE
      IF(UMAX.NE.0.0) GO TO 110
      WRITE (KOUT,2000)
      RETURN
C
  110 NEWP=0
      DO 500 L=1,NTIME,NPL
      LH=L+NPL-1
      IF(LH.GT.NTIME) LH=NTIME
      IF (NEWP.NE.0) GO TO 115
      LINE=99
      CALL NUPAJ (1,1,ITEST)
      WRITE (KOUT,1000) FHED
C
C        PRINT TIMES
C
  115 TIME(1)=TO+DT
      I=1
      LL=L+1
      DO 400 J=LL,LH
      I=I+1
  400 TIME(I)=TIME(I-1)+DT
      TO=TIME(I)
      WRITE (KOUT,1003) (TIME(J),J=1,I)
      NEWP=NEWP+1
      IF(NEWP.EQ.NPP) NEWP=0
C
C        PLOT SHAPES
C
      DO 300 N=1,NS
      NPTS=(NMAX+1)*(LH-L+2)-1
      DO 150 I=1,NPTS
  150 P(I)=BLK
      I=0
      DO 200 II=L,LH
      I=I+NMAX+1
      P(I)=VERT
      UU=U(N,II)*RMAX/UMAX
      IF (UU) 205,210,205
  205 UA=ABS(UU)
      UU=(UA+0.5)*UU/UA
  210 NU=UU
  200 P(I+NU)=STAR
C
C        PRINT
C
      WRITE (KOUT,1001) (P(I),I=1,NPTS)
C
C        SPACING
C
      IF (ISPACE) 300,300,220
  220 DO 240 I=1,ISPACE
  240 WRITE (KOUT,1002) (VERT,J=L,LH)
  300 CONTINUE
C
C        BASE LINE
C
      WRITE (KOUT,1001) (HORIZ,I=1,NPTS)
      WRITE (KOUT,1005)
  500 CONTINUE
      LINE=99
      CALL NUPAJ (1,1,ITEST)
      WRITE (KOUT,2001) UMAX,TMAX
C
      RETURN
C
 1000 FORMAT (///28H  FRAME DISPLACEMENT PLOT...   4A5)
 1001 FORMAT (1H ,4X,130A1)
 1002 FORMAT (1H ,4X,10(10X,A1))
```

```
 1003 FORMAT (8HOTIME... 10(4X,F7.3))
 1005 FORMAT (/)
 2000 FORMAT (37H DISPLACEMENTS ALL ZERO...NOT PLOTTED )
 2001 FORMAT (19HOSCALE - ONE INCH = E15.6,9X,13HMAXIMUM AT T=    E15.5/
     1 57HONOTE..PLOT IS OF DYNAMIC DISPLACEMENTS ONLY AND DOES NOT /
     2 7X,49HINCLUDE ANY SCALING BY LOAD CASE DEFINITION CARDS  )
      END



      SUBROUTINE TABP (MTOT)
C
C     CALCULATING MEMBER FORCES AND STRESSES
C
      COMMON /GEN1 / NST,NDF,NTF,NLD,NAT,NFQ,NSD,NOPT,NRGD,NDSP,NUBC,
     *               NPLT,NSS,I3,IS,NAL
      COMMON /FTAPE/ KTMP,KINP,KOUT,KSTR,NTAPE,KTAPE
      COMMON /KOPT / NINP,NOUT
      COMMON /JUNK/ FHED(4),IF,IFC
      COMMON /DYN  / NTIME,DT,NPC,DAMP,NMD,IHTYP,HOT
C
      COMMON A(1)
C
      M=NFQ
      IF (NAT.NE.3) M=0
      NO=1
      LO=NO+NST*NDF
      L1=LO+2*NST
      L2=L1+M
      L3=L2+M*M
      L4=L3+11*NLO
      L5=L4+NLD
      N1=L5+10*M
      IF (N1.GT.MTOT) CALL TABR((N1-MTOT),1)
      CALL TABC (A(L3),A(L4),NLD)
      IF (NOPT.EQ.1) GO TO 999
C
      IF (NAT.NE.3) GO TO 40
      REWIND 1
      READ (1) (A(L1+I-1),I=1,NFQ)
      CALL COCIN (A(L1),A(L2),DAMP,NFQ)
   40 CONTINUE
C
      REWIND 2
      REWIND NTAPE
      KO=0
      M=0
      MLD=6+NFQ
      IF (NAT.EQ.4) MLD=6+NTIME
      N2=N1+NST*MLD
      IF (N2.GT.MTOT) CALL TABR((N2-MTOT),1)
C
      DO 500 I=1,NTF
C
      CALL TABU (A(NO),A(LO),A(L1),A(L2),A(L3),A(L4),A(N1),
     *               NST,NLD,MLD,NAT,NFQ)
C
      WRITE (KTAPE) IF,IFC,FHED
      WRITE (NTAPE) IF,IFC,FHED
      IF(IF.EQ.M) GO TO 200
      IF(KO.EQ.0) GO TO 100
      IF(KO.NE.1) GO TO 100
      DO 50 K=1,NBK
   50 BACKSPACE 3
  100 BACKSPACE 3
      READ  (3)  M,NC,NS,NCP,NBP,NFEF,NPAN,NDIG,MCONL
      BACKSPACE 3
C
      KO=1
  200 IF(IFC.NE.0) GO TO 500
      IF(KO.NE.2) GO TO 300
      DO 250 K=1,NS
  250 READ (3) XX
C
  300 MM=2*NC
      CALL SECOND (TS)
C
      NB=NC-1
      NN=4*NC+NS+1
      N2=N1+NST*MLD
      N3=N2+NB
      N4=N3+8*NCP
      N5=N4+8*NBP
      N6=N5+7*NFEF
      N7=N6+NB*NS
      N8=N7+NS*NB*4
      N9=N8+NS*NC
      N10=N9+2*NPAN
      N11=N10+2*NPAN
      N12=N11+3*NDIG
      N13=N12+NDIG
      N14=N13+NFEF
      N15=N14+NFEF*MCONL*2
      N16=N15+NB
      N17=N16+NN*MM
      N18=N17+NN*MLD
      IF(N18.GT.MTOT) CALL TABR((N18-MTOT),1)
C
      WRITE (KTAPE)  M,NC,NS,NCP,NBP,NFEF,NPAN,NDIG,MCONL
```

```
C
      CALL TABO (A(NO),A(LO),A(L1),A(L2),A(L3),A(L4),A(L5),
     +           A(N1),A(N2),A(N3),A(N4),A(N5),A(N6),A(N7),A(N8),
     +           A(N9),A(N10),A(N11),A(N12),A(N13),A(N14),A(N15),
     +           A(N16),A(N17),
     +           NST,NC,NB,NS,NCP,NBP,NFEF,MCONL,NPAN,NDIG,
     +                NN,MM,NLD,MLD,KO,NDSP,NFO,NAL)
C
      IF (NOUT.EQ.0) GO TO 400
      CALL SECOND (TE)
      TE=TE-TS
      CALL NUPAJ (1,4,ITEST)
      WRITE (KOUT,2000) I,TE
  400 KO =2
C
  500 NBK=NS+1
C
      N1=1+NST*NDF
      N2=N1+NST
      N3=N2+NTF*NST*6
      N4=N3+NTF
      N5=N4+NTF*4
      N6=N5+NTF*2
      IF (N6.GT.MTOT) CALL TABR((N6-MTOT),1)
      CALL TABJ (A(N1),A(N2),A(N3),A(N4),A(N5),NST,NTF)
C
  999 RETURN
C
 2000 FORMAT (//13H FRAME NO.  =  I6/13H STRESS TIME= F6.2)
      END


      SUBROUTINE TABC (XM,IXM,NLD)
C
C     READING IN LOAD CASE DATA
C
      DIMENSION XM(11,NLD),IXM(NLD)
C
      COMMON /IHEDR/ IHED(28),IDATE(2),NPAGE(2),MAXLIN,LINE
      COMMON /FTAPE/ KTMP,KINP,KOUT,KSTR,NTAPE
      COMMON /KOPT / NINP,NOUT
C
      READ (KINP,1000) (M,IXM(L),(XM(I,L),I=1,11),L=1,NLD)
C
      IF (NINP.EQ.0) GO TO 999
      LINE=99
      CALL NUPAJ (1,6,ITEST)
      WRITE (KOUT,2000)
C
      DO 100 L=1,NLD
      CALL NUPAJ (1,1,ITEST)
      IF (ITEST.EQ.0) GO TO 100
      WRITE (KOUT,2000)
      LINE=LINE+6
  100 WRITE (KOUT,3000) L,IXM(L),(XM(I,L),I=1,11)
C
      CALL NUPAJ (1,15,ITEST)
      WRITE (KOUT,4000)
C
  999 RETURN
C
 1000 FORMAT (2I5,9F5.0,2A6)
 2000 FORMAT (//
     +          45H LOAD CASE DEFINITION DATA                     //
     +          3X2HNO2X2HIC
     +          5X1HI4X2HII3X3HIII4X2HIV5X1HA5X1HB1X5HDYN-11X5HDYN-2
     +          1X5HDYN-32X12HLOAD CASE ID/)
 3000 FORMAT (I5,I4,9F6.2,2X,2A6)
 4000 FORMAT (///
     +          45H FOR DYNAMICS BY THE RESPONSE SPECTRUM METHOD//
     +          45H DYNAMIC 1 . . . SRSS MODAL COMBINATION       /
     +          45H DYNAMIC 2 . . . ABS  MODAL COMBINATION       /
     +          45H DYNAMIC 3 . . . CQC  MODAL COMBINATION     ///
     +          45H FOR DYNAMICS BY THE TIME HISTORY METHOD     //
     +          45H DYNAMIC 1 . . . NOT USED                     /
     +          45H DYNAMIC 2 . . . NOT USED                     /
     +          45H DYNAMIC 3 . . . TIME HISTORY MODAL ANALYSIS )
       END


      SUBROUTINE TABU (NSC,SD,STOR,RO,XM,IXM,U,NST,NLD,MLD,NAT,NFO)
C
C     PRINTING LATERAL FRAME DISPLACEMENTS
C
      DIMENSION STOR(NFO),RO(NFO,NFO)
      DIMENSION SD(NST,2),U(NST,MLD),XM(11,NLD),IXM(NLD),NSC(NST,1)
C
      COMMON /IHEDR/ IHED(28),IDATE(2),NPAGE(2),MAXLIN,LINE
      COMMON /FTAPE/ KTMP,KINP,KOUT,KSTR,NTAPE
      COMMON /KOPT / NINP,NOUT
      COMMON /DYN  / NTIME,DT,NPC,DAMP,NMD
      COMMON /JUNK/ FHED(4),IF,IFC,NS,UU(4),T1(4),T2(4)
C
      READ (2) IF,IFC,NS,((U(N,L),N=1,NS),L=1,MLD),FHED
      IF (IFC.NE.0) GO TO 600
      IF (NOUT.EQ.0) GO TO 600
```

```
C
      LINE=99
C
      LLO=NLD+NPQ
C
      DO 500 LL=1,LLO,4
C
      IT=NLD-LL+1
      LH=LL+3
      IF (LH.GT.LLO) LH=LLD
      IH=LH+1-LL
C
      DO 150 L=LL,LH
      I=L+1-LL
      IF (L.GT.NLD) GO TO 155
      T1(I)=XM(10,L)
      T2(I)=XM(11,L)
      GO TO 150
  155 T1(I)=6H  MODE
      LLL=L-NLD
      ENCODE (6,1000,T2(I)) LLL
  150 CONTINUE
C
      CALL NUPAJ (1,6,ITEST)
      LINE=LINE-1
      WRITE (ROUT,2000) FHED,(T1(I),T2(I),I=1,IH)
C
      DO 400 N=1,NS
C
  250 DO 360 L=LL,LH
      I=L+1-LL
      UU(I)=0.
C
      IF (L.GT.NLD) GO TO 350
C
      IADD=IXM(L)
C
      DO 300 J=1,6
      IF (XP(J,L).EQ..0) GO TO 300
      UUU=U(N,J)
      IF (IADD.GT.0) UUU=ABS(UUU)
      UU(I)=UU(I)+UUU*XM(J,L)
  300 CONTINUE
C
      NATT=NAT-3
      IF (NATT.LT.0) GO TO 360
C
      SRSS=0.
      SABS=0.
      THST=0.
      DO 410 J=7,MLD
      TEMP=ABS(U(N,J))
      IF (NATT) 160,160,180
  160 SRSS=SRSS+TEMP*TEMP
      SABS=SABS+TEMP
      STOR(J-6)=U(N,J)
      GO TO 410
  180 IF (TEMP.GT.THST) THST=TEMP
  410 CONTINUE
      SRSS=SQRT(SRSS)
      IF (NAT.EQ.3) THST=COC(STOR,RO,NFQ)
C
      TEMP=SRSS*XM(7,L)+SABS*XM(8,L)+THST*XM(9,L)
      UU(I)=UU(I)+TEMP
      GO TO 360
C
  350 UU(I)=U(N,L+6-NLD)
C
  360 CONTINUE
C
  370 K=NST-NS
      NL=NSC(N+K,IF)
      CALL NUPAJ (1,1,ITEST)
      IF (ITEST.EQ.0) GO TO 400
      LINE=LINE+5
      WRITE (ROUT,2000) FHED,(T1(I),T2(I),I=1,IH)
  400 WRITE (6,2001) SO(NL,IF,(UU(I),I=1,IH)
C
  500 CONTINUE
C
C     PLOT DYNAMIC DISPLACEMENTS
C
      IF (NAT.NE.4) GO TO 600
      CALL TABPLT (U(1,6),NST,NTIME,DT)
C
  600 RETURN
C
 1000 FORMAT (I6)
 2000 FORMAT (//
     *        29H LATERAL FRAME DISPLACEMENTS 4H IN ,4A5//
     *        6H LEVEL,4(3X,2A6))
 2001 FORMAT (1X,A5,4F15.6)
      END




      SUBROUTINE TABO
     *             (NSC,SD,STOR,RO,XM,IXM,FM,U,BW,CP,BP,FEF,LB,LDB,LC,LP,
     *                   PP,LDIG,PDIG,IFTY,CONL,SL,S,R,
```

```
     *                NST,NC,NB,NS,NCP,NBP,NFEF,MCONL,NPAN,NDIG,
     *                NN,MM,NLD,MLD,KD,NDSP,NFQ,NAL)
C
      DIMENSION STOR(NFQ),RO(NFQ,NFQ)
      DIMENSION BW(NB),CP(8,NCP),BP(8,NBP),FEF(7,NFEF),SD(NST,2),
     1 LB(NS,NB),LDB(NS,NB,4),LC(NS,NC),LP(2,NPAN),PP(2,NPAN)
     2,S(MP,NN),R(NN,MLD),U(NST,MLD),XM(11,NLD),LDIG(3,NDIG),PDIG(NDIG)
     3,IFTY(NFEF),CONL(2,MCONL,NFEF),IXM(NLD),SL(NB),FM(10,NFQ)
     4,NSC(NST,1)
C
      COMMON /GEN2 / RLAB(3),ANI,ANP,RIGID
      COMMON /IHEDP/ IHED(28),IDATE(2),NPAGE(2),MAXLIN,LINE
      COMMON /FTAPE/ KTMP,KINP,KOUT,KSTR,NTAPE,KTAPE
      COMMON /KOPT / NINP,NOUT
      COMMON /DYN  / NTIME,DT,NPC,DAMP,NMD
      COMMON /JUNK / FHED(4),IF,IFC,ND,NF,LM(6),CA(10,6),F(10,9),Q(4,4),
     *               P(10),SV(6),T(6),SDI,LTYP
C
      INTEGER PP
      INTEGER PDIG
C
      COM=ANP/ANI/ANI
C
C     SET FOUNDATION DISPLACEMENTS AND LATERAL DISPLACEMENTS
C
      IF(KD.EQ.2) GO TO 50
      BACKSPACE 3
      READ  (3) BW,CP,BP,FEF,LB,LDB,LC,LP,IFEF,PP,LDIG,PDIG,IFTY,CONL,SL
      BACKSPACE 3
   50 MN=4*NC
      KL=MM+1
      DO 475 L=1,MLD
      DO 425 K=1,MN
  425 R(K,L)=0.0
      DO 450 N=1,NS
      KK=MN+N
  450 R(KK,L)=U(N,L)
  475 R(NN,L)=0.0
C
      NL=NST-NSC(NST-NS+1,IF)+1
      WRITE (NTAPE) NL
      WRITE (KTAPE)
     *            BW,CP,BP,FEF,LB,LDB,LC,LP,IFEF,PP,LDIG,PDIG,IFTY,CONL,SL
C
C     STORY BY STORY  CALCULATION OF MEMBER FORCES
C
      N=NS
C
C     CALCULATE ROTATIONS AND VERTICAL DISPLACEMENT AT LEVEL N
C
  300 BACKSPACE 3
      READ  (3) (((S(I,J),J=1,NN),(R(I,L),L=1,4)),I=1,MM)
      BACKSPACE 3
      DO 100 I=1,10
      DO 100 J=1,6
  100 CA(I,J)=0.0
C
      DO 310 I=1,6
  310 SV(I)=0.
      DO 525 II=1,MM
      I=MM-II+1
      KK=I+1
      DO 525 K=KK,NN
      DO 525 L=1,MLD
  525 R(I,L)=R(I,L)-S(I,K)*R(K,L)
C
      LV=NST-N+1
      K=NST-NS
      NL=NSC(N+K,IF)
      SDI=SD(NL,1)
C
C     CALCULATING REAL STORY HEIGHT OF FRAME
C
      SXL=0.
      I=NSC(N+K,IF)
      L=NST
      IF (N.NE.NS) L=NSC(N+K+1,IF)-1
      DO 710 J=I,L
  710 SXL=SXL+SD(J,2)
C
C     JOINT DISPLACEMENTS AND ROTATIONS
C
      IF (NOUT.EQ.0) GO TO 909
      IF (NDSP.EQ.0) GO TO 909
      CALL TABN (R,XM,IXP,STOR,RO,NLD,MLD,NC,NN,NFQ,NMD)
  909 CONTINUE
C
C     CALCULATE COLUMN FORCES
C
  911 ND=6
      NF=4
      LTYP=1
      LINE=99
      DO 500 M=1,NC
C
      MC=LC(N,M)
      IF(MC.EQ.NCP) GO TO 500
C
      A=0.
      B=0.
```

```
      XL=SXL
      IF (NB.EQ.0) GO TO 420
      IF (N.NE.NS) GO TO 900
C
C     N.EQ.NS
C
      IF (M.EQ.1) GO TO 905
      IF (M.EQ.NC) GO TO 906
      ML=LB(N,M-1)
      MR=LB(N,M)
      A=BP(5,ML)
      IF (A.LT.BP(5,MR)) A=BP(5,MR)
      B=SL(M-1)
      IF (B.LT.SL(M)) B=SL(M)
      GO TO 420
  905 MR=LB(N,M)
      A=BP(5,MR)
      B=SL(M)
      GO TO 420
  906 ML=LB(N,M-1)
      A=BP(5,ML)
      B=SL(M-1)
      GO TO 420
C
C     N.NE.NS
C
  900 IF (M.EQ.1) GO TO 901
      IF (M.EQ.NC) GO TO 910
      MRR=LB(N+1,M)
      MR=LB(N,M)
      MLL=LB(N+1,M-1)
      ML=LB(N,M-1)
      A=BP(5,ML)
      B=BP(6,MLL)
      IF (A.LT.BP(5,MR)) A=BP(5,MR)
      IF (B.LT.BP(6,MRR)) B=BP(6,MRR)
      GO TO 420
  901 MR=LB(N,M)
      MRR=LB(N+1,M)
      A=BP(5,MR)
      B=BP(6,MRR)
      GO TO 420
  910 ML=LB(N,M-1)
      MLL=LB(N+1,M-1)
      A=BP(5,ML)
      B=BP(6,MLL)
C
  420 D=CP(5,MC)*RIGID
      XLL=XL-A-B
      AAA=A
      BBB=B
      A=A-D
      B=B-D
      IF (B.LT.0.0) B=0.0
      IF (A.LT.0.0) A=0.0
      XL=XL-A-B
      AAA=AAA-A
      BBB=BBB-B
C
      BB=0.0
      IF ( CP(4,MC) ) 430,430,429
  429 BB=14.4*CP(3,MC)/(XL**2*CP(4,MC))
  430 COMM=2.*CP(1,MC)*CP(3,MC)/(XL*(1.+2.*BB))
      SA=COMM*(2.+BB)
      SB=COMM*(1.-BB)
      SC=CP(2,MC)*CP(1,MC)/XL
      CALL TABZ(SA,SB,SC,XL,A,B)
C
      DO 435 I=1,6
      CA(4,I)=-(CA(1,I)+CA(2,I))/XL
      CA(1,I)=CA(1,I)+CA(4,I)*BBB
  435 CA(2,I)=CA(2,I)+CA(4,I)*AAA
C
      LM(3)=2*M
      LM(1)=LM(3)+MM
      LM(5)=LM(1)-1
      LM(6)=LM(3)-1
      LM(4)=MN+N
      LM(2)=LM(4)+1
C
      CALL TABW (R,XM,IXP,STOR,RO,FM,NN,NLD,MLD,NFQ,M,1,1)
      IF (NOUT.EQ.0) GO TO 438
C
      CALL TABST (FM,CP(2,MC),CP(4,MC),CP(3,MC),CP(5,MC),NMD,COM,1)
      CALL TABST (F ,CP(2,MC),CP(4,MC),CP(3,MC),CP(5,MC),  9,COM,1)
C
      CALL TABW (R,XM,IXP,STOR,RO,FM,NN,NLD,MLD,NFQ,M,1,2)
C
  438 DO 440 I=1,6
  440 SV(I)=SV(I)+F(4,I)
  500 CONTINUE
C
      DO 70 I=1,10
      DO 70 J=1,6
   70 CA(I,J)=0.0
C
C     CALCULATE BEAM FORCES
C
      IF(NB.EQ.0) GO TO 565
      IF (NOUT.EQ.0) GO TO 770
```

```
      ND=4
      NF=5
      LTYP=1
      DO 600 M=1,NB
      MB=LB(N,M)
      IF(MB.EQ.NBP) GO TO 600
C
      ML=LC(N,M)
      MR=LC(N,M+1)
      A=CP(5,MR)/2.
      B=CP(5,ML)/2.
      D=(BP(5,MB)+BP(6,MB))*RIGID
      AAA=A
      BBB=B
      A=A-D
      B=B-C
      IF (A.LT.0.0) A=0.0
      IF (B.LT.0.0) B=0.0
      XL=BW(M)-A-B
      BB=0.0
      IF (BP(3,MB).NE.4.0.OR.BP(4,MB).NE.0.5) GO TO 431
      IF (BP(7,MB)) 426,426,427
  427 BB=14.4*BP(2,MB)/(XL**2*BP(7,MB))
  426 COMM=2.*BP(1,MB)*BP(2,MB)/(XL*(1+2.*BB))
      SA=COMM*(2.+BB)
      SB=COMM*(1.-BB)
      GO TO 428
  431 EIOL=BP(1,MB)*BP(2,MB)/XL
      SA=BP(3,MB)*EIOL
      SB=BP(4,MB)*SA
  428 CALL TABZ(SA,SB,0.0,XL,A,B)
      LM(1)=2*M
      LM(2)=LM(1)-1
      LM(3)=LM(1)+2
      LM(4)=LM(3)-1
      XLL=XL
      XL=BW(M)-AAA-BBB
      AA=A
      BB=B
      A=AAA
      B=BBB
      AAA=AAA-AA
      BBB=BBB-BB
C
      DO 439 I=1,4
      CA(4,I)= (CA(1,I)+CA(2,I))/XLL
      CA(5,I)=-CA(4,I)
      CA(1,I)=CA(1,I)-CA(4,I)*BBB
      CA(2,I)=CA(2,I)+CA(5,I)*AAA
  439 CA(3,I)=(CA(1,I)-CA(2,I))/2.0
C
      DO 540 L=1,4
C
      DO 528 I=1,4
  528 Q(I,L)=0.
      J=LDB(N,M,L)
      IF (J.LE.0) GO TO 540
C
      WW=FEF(5,J)
      IF (L.EQ.NAL) WW=WW+BP(7,MB)*BP(8,MB)
      V=WW*XL/2.
      YM=V*XL/6.
      Q(1,L)=-FEF(1,J)-YM
      Q(2,L)=-FEF(2,J)-V
      Q(3,L)= FEF(3,J)+YM
      Q(4,L)=-FEF(4,J)-V
C
      NCON=IFTY(J)
      IF (NCON.EQ.0) GO TO 530
      DO 535 NCN=1,NCON
      P1=CONL(2,NCN,J)
      DIST=CONL(1,NCN,J)
      IF (DIST.LT..0) DIST=BW(M)*ABS(DIST)
      AA=DIST-B
      BB=XL-AA
      TEMP=P1*AA*BB/XL/XL
      Q(1,L)=Q(1,L)-TEMP*BB
      Q(3,L)=Q(3,L)+TEMP*AA
      TEMP=P1/XL**3
      Q(2,L)=Q(2,L)-TEMP*BB*BB*(3.*AA+BB)
      Q(4,L)=Q(4,L)-TEMP*AA*AA*(AA+3.*BB)
  535 CONTINUE
C
C     CORRECTIONS FOR PIN ENDS
C
  530 CALL TABG (BP(2,MB),BP(3,MB),BP(4,MB),
     *               Q(1,L),Q(3,L),Q(2,L),Q(4,L),XLL,L)
  540 CONTINUE
C
      CALL TABW (R,XM,IXM,STOR,RO,FM,NN,NLD,MLD,NFO,M,2,1)
C
C     CORRECTIONS TO BEAM FORCES DUE TO SPAN LOADINGS
C
      CALL TABB (F,Q,LDB,BP,IFTY,FEF,CONL,N,M,NS,NB,MB,NBP,NFEF,MCONL,
     *               XL,AAA,BBB,A,B,XLL,NAL)
C
      CALL TABST (FM,0.,BP(7,MB),BP(2,MB),(BP(5,MB)+BP(6,MB)),NFO,COM,2)
      CALL TABST (F ,0.,BP(7,MB),BP(2,MB),(BP(5,MB)+BP(6,MB)),  9,COM,2)
C
      CALL TABW (R,XM,IXM,STOR,RO,FM,NN,NLD,MLD,NFO,M,2,2)
```

```
C
  550 CONTINUE
  600 CONTINUE
C
C     CALCULATE PANEL FORCES
C
  770 IF(NPAN.EQ.0) GO TO 811
      ND=6
      NF=4
      LTYP=1
C
      DO 775 I=1,6
      DO 775 J=1,10
  775 CA(J,I)=0.0
C
      DO 800 L=1,NPAN
      NP=NS-LP(1,L)+1
      IF(NP.NE.N) GO TO 800
C
      XL=SXL
      M=LP(2,L)
      J=PP(1,L)
      MC=PP(2,L)
      IF (MC.EQ.0) GO TO 800
C
      D=0.0
      DO 740 I=P,J
  740 D=D+BW(I)
C
      BB=XL**2*CP(4,MC)*CP(1,MC)/2.4
      IF (BB) 730,730,725
  725 B9=6.0*CP(1,MC)*CP(3,MC)/BB
  730 COMM=2.0*CP(1,MC)*CP(3,MC)/(XL*(1.+2.*BB))
      SA=COMM*(2.+BB)
      SB=COMM*(1.-BB)
      SC=CP(1,MC)*CP(2,MC)/XL
C
      CALL TABT (2,SA,SB,SC,XL,D,SS,CA)
C
      LM(2)=PN+N
      LM(1)=LM(2)+1
      LM(5)=2*M-1
      LM(6)=LM(5)+2*(J-M+1)
      LM(3)=LM(5)+MM
      LM(4)=LM(6)+MM
C
      CALL TABW (R,XM,IXM,STOR,RO,FM,NN,NLD,PLD,NFO,L,3,1)
      IF (NOUT.EQ.0) GO TO 730
C
      CALL TABST (FM,CP(2,MC),CP(4,MC),CP(3,MC),CP(5,MC),NMO,COM,1)
      CALL TABST (F ,CP(2,MC),CP(4,MC),CP(3,MC),CP(5,MC),  9,COM,1)
C
      CALL TABW (R,XM,IXM,STOR,RO,FM,NN,NLD,MLD,NFO,L,3,2)
C
  730 DO 790 I=1,6
  790 SV(I)=SV(I)+F(4,I)
  800 CONTINUE
  811 CONTINUE
C
C     CALCULATE DIAGONAL FORCES
C
      IF (NDIG.EQ.0) GO TO 565
      ND=6
      NF=4
      LTYP=1
C
      DO 785 I=1,6
      DO 785 J=1,10
  785 CA(J,I)=0.
C
      DO 560 M=1,NDIG
      NDI=NS-LDIG(1,M)+1
      IF (N.NE.NDI) GO TO 560
C
      DY=SXL
      IB=LDIG(2,M)
      IT=LDIG(3,M)
      MC=PDIG(M)
      IF (MC.EQ.0) GO TO 560
C
      DX=0.
      IF (IT.EQ.1) GO TO 521
      DO 510 I=2,IT
  510 DX=DX+BW(I-1)
  521 IF (IB.EQ.1) GO TO 541
      DO 531 I=2,IB
  531 DX=DX-BW(I-1)
  541 XL=SQRT(DX**2+DY**2)
C
      BB=0.0
      IF ( CP(4,MC) ) 580,580,585
  585 BB=14.4*CP(3,MC)/(XL**2*CP(4,MC))
  580 COMM=2.*CP(1,MC)*CP(3,MC)/(XL*(1.+2.*BB))
      SA=COMM*(2.+BB)
      SB=COMM*(1.-BB)
      SC=CP(2,MC)*CP(1,MC)/XL
C
      CALL TABV (2,SA,SB,SC,XL,DX,DY,SS,CA)
C
      LM(1)=2*IB+MM
```

```
      LM(2)=PN+N+1
      LM(3)=2*IT
      LM(4)=LM(2)-1
      LM(5)=LM(1)-1
      LM(6)=LM(3)-1
C
      CALL TABW (R,XM,IXP,STOR,RO,FM,NN,NLO,MLD,NFQ,M,4,1)
      IF (NOUT.EQ.0) GO TO 747
C
      CALL TABST (FM,CP(2,MC),CP(4,MC),CP(3,MC),CP(5,MC),NMO,COM,1)
      CALL TABST (F ,CP(2,MC),CP(4,MC),CP(3,MC),CP(5,MC),  9,COM,1)
C
      CALL TABW (R,XP,IXP,STOR,RO,FM,NN,NLO,MLD,NFQ,P,4,2)
C
  747 DO 791 I=1,6
  791 SV(I)=SV(I)+F(3,I)*DX/XL+F(4,I)*DY/XL
C
  560 CONTINUE
C
C     SHIFT DISPLACEMENTS
C
  569 DO 570 K=1,MM
      KK=MP+K
      DO 570 L=1,MLD
      R(KK,L)=R(K,L)
  570 R(K,L)=0.0
C
      IF (NOUT.EQ.0) GO TO 581
      CALL NUPAJ (1,6,ITEST)
      WRITE (6,1000) SV
  581 K=NST-NS
      I=NST-NSC(N+K,IF)+1
      L=0
      IF (N.NE.NS) L=NST-NSC(N+K+1,IF)+1
      L=L+1
      DO 620 J=L,I
  620 WRITE (NTAPE) SV
C
  610 N=N-1
      IF (N.NE.0) GO TO 300
C
  810 RETURN
C
 1000 FORMAT (//20H STORY SHEAR
     *        1H/,21(1H-),16HLOAD  CONDITIONS,21(1H-),1H//
     *        20X,9X1H18X2HI17X3HIII8X2HIV9X1HA9X1HB/20X,6F10.2)
      END
      SUBROUTINE TANN (R,XM,IXM,STOR,RO,NLD,MLD,NC,NN,NFQ,NMO)
C
C     PRINTING OUT JOINT DISPLACEMENTS AND ROTATIONS
C
      DIMENSION STOR(NFQ),RO(NFQ,NFQ)
      DIMENSION R(NN,MLD),XP(11,NLD),IXM(NLD)
C
      COMMON /GENL/ IDUM(4),NAT
      COMMON /IHEDR/ IHED(28),IDATE(2),NPAGE(2),MAXLIN,LINE
      COMMON /FTAPE/ KTMP,KINP,KOUT,KSTR,NTAPE
      COMMON /JUNK/ FHED(4),IF,IFC,ND,NF,LM(6),DD(2,4),
     *              SRSS(2),SABS(2),THST(2),DUM(174),SDI
C
      DIMENSION IHD(2)
C
      DATA IHD/4HVERT,4HROTN/
      DATA COLUMN/6HCOLUMN/
C
      LINE=99
C
      DO 500 N=1,NC,4
C
      NH=N+3
      IF (NH.GT.NC) NH=NC
      JH=NH+1-N
C
      CALL NUPAJ (1,9,ITEST)
      LINE=LINE-3
      WRITE (KOUT,2004) SDI,FHED
      WRITE (KOUT,2000) (COLUMN,J,J=N,NH)
      WRITE (KOUT,2005)
C
      LLD=NLD+NMO
C
      DO 400 L=1,LLD
      LL=L-NLD+6
C
      IF (L.GT.NLD) GO TO 130
C
      IADD=IXM(L)
      T1=XM(10,L)
      T2=XM(11,L)
      GO TO 140
C
  130 T1=6H  MODE
      LLL=L-NLD
      ENCODE (6,1000,T2) LLL
  140 CONTINUE
C
C     STATIC COMPONENTS
```

```
C
      DO 300 K=N,NH
      KK=2*(K-1)
      I=K-N+1
C
      IF (L.GT.MLD) GO TO 340
C
      DO 110 M=1,2
      DD(M,I)=0.
      DO 120 J=1,6
      TEMP=R(KK+M,J)
      IF (IADD.GT.0) TEMP=ABS(TEMP)
      DD(M,I)=DD(M,I)+TEMP*XM(J,L)
  120 CONTINUE
  110 CONTINUE
C
C     DYNAMIC COMPONENTS
C
  150 NATT=NAT-3
      IF (NATT.LT.0) GO TO 300
C
      DO 170 M=1,2
      SRSS(M)=0.
      SABS(M)=0.
  170 THST(M)=0.
C
      DO 200 M=1,2
      DO 210 J=7,MLD
      TEMP=ABS(R(KK+M,J))
      IF (NATT) 160,160,180
  160 SRSS(M)=SRSS(M)+TEMP*TEMP
      SABS(M)=SABS(M)+TEMP
      STOR(J-6)=R(KK+M,J)
      GO TO 210
  180 IF (TEMP.GT.THST(M)) THST(M)=TEMP
  210 CONTINUE
      IF (NAT.EQ.3) THST(M)=COC(STOR,RO,NFO)
  200 CONTINUE
C
      DO 220 M=1,2
      SRSS(M)=SQRT(SRSS(M))
      TEMP=SRSS(M)*XM(7,L)+SABS(M)*XM(8,L)+THST(M)*XM(9,L)
  220 DD(M,I)=DD(M,I)+TEMP
      GO TO 300
C
C     MODAL BREAKDOWN
C
  340 DO 310 M=1,2
  310 DD(M,I)=R(KK+M,LL)
C
  300 CONTINUE
C
      CALL NUPAJ (1,3,ITEST)
      IF (ITEST.EQ.0) GO TO 350
      LINE=LINE+6
      WRITE (KOUT,2004) SDI,FMED
      WRITE (KOUT,2000) (COLUMN,J,J=N,NH)
      WRITE (KOUT,2005)
C
  350 DO 250 M=1,2
  250 WRITE (KOUT,2001) T1,T2,IMD(M),(DD(M,I),I=1,IM)
      WRITE (KOUT,2005)
      LINE=LINE+1
C
  400 CONTINUE
  500 CONTINUE
C
      RETURN
C
 1000 FORMAT (I6)
 2000 FORMAT (//1H ,4X,4HLOAD,4X,6X,3X,9HDIRECTION,8(3X,A6,I3))
 2001 FORMAT (1H ,2A6,6X,8X,A4,4F12.5)
 2004 FORMAT (//43H JOINT DISPLACEMENTS AND ROTATIONS AT LEVEL,1H ,A5,
     +             3H IN,1H ,4A5)
 2005 FORMAT (1H )
      END




      SUBROUTINE TAB2(SA,SB,SC,XL,A,B)
C
C     FORCE-DISPLACEMENT TRANSFORMATION MATRICES FOR BEAMS AND COLUMNS
C
      COMMON /JUNK/ FMED(4),IF,IFC,ND,NF,LM(6),CA(10,6),F(10,9),T(2,4)
C
      T(2,1)=B/XL
      T(1,1)=1.0+T(2,1)
      T(1,2)=1.0/XL
      T(2,2)=T(1,2)
      T(1,3)=A/XL
      T(2,3)=1.0+T(1,3)
      T(1,4)=-T(1,2)
      T(2,4)=T(1,4)
C
      DO 100 I=1,4
      CA(1,I)=SA*T(1,I)+SB*T(2,I)
  100 CA(2,I)=SB*T(1,I)+SA*T(2,I)
      CA(3,5)=SC
      CA(3,6)=-SC
```

```
  120 RETURN                                                TABS80 4090
      END                                                   TABS80 4091


      SUBROUTINE TABST (F,AA,AV,AI,DD,NLO,COM,ITAG)         TABS80 4092
C                                                           TABS80 4093
C     CALCULATING STRESSES                                  TABS80 4094
C                                                           TABS80 4095
      DIMENSION F(10,NLD)                                   TABS80 4096
C                                                           TABS80 4097
      DATA TOL /1.E-8/                                      TABS80 4098
C                                                           TABS80 4099
      IF (NLD.EQ.0) GO TO 999                               TABS80 4100
C                                                           TABS80 4101
      ZZ=0.                                                 TABS80 4102
      IF (DD.GT.TOL) ZZ=2.*AI/DD                            TABS80 4103
C                                                           TABS80 4104
      IF (ITAG.EQ.2) GO TO 200                              TABS80 4105
C                                                           TABS80 4106
  100 DO 10 L=1,NLD                                         TABS80 4107
      IF (ZZ.LE..0) GO TO 15                                TABS80 4108
      F(5,L)=F(1,L)*COM/ZZ                                  TABS80 4109
      F(6,L)=F(2,L)*COM/ZZ                                  TABS80 4110
   15 IF (AA.GT..0) F(7,L)=F(3,L)*COM/AA                    TABS80 4111
      IF (AV.GT..0) F(8,L)=F(4,L)*COM/AV                    TABS80 4112
   10 CONTINUE                                              TABS80 4113
      GO TO 999                                             TABS80 4114
C                                                           TABS80 4115
  200 DO 20 L=1,NLD                                         TABS80 4116
      IF (ZZ.LE..0) GO TO 25                                TABS80 4117
      F(6,L)=F(1,L)*COM/ZZ                                  TABS80 4118
      F(7,L)=F(2,L)*COM/ZZ                                  TABS80 4119
      F(8,L)=F(3,L)*COM/ZZ                                  TABS80 4120
   25 IF (AV.LE..0) GO TO 20                                TABS80 4121
      F(9,L)=F(4,L)*COM/AV                                  TABS80 4122
      F(10,L)=F(5,L)*COM/AV                                 TABS80 4123
   20 CONTINUE                                              TABS80 4124
C                                                           TABS80 4125
  999 RETURN                                                TABS80 4126
C                                                           TABS80 4127
      END                                                   TABS80 4128


      SUBROUTINE TABG (I,K,C,MI,MJ,VI,VJ,XL,ITAG)           TABS80 4129
C                                                           TABS80 4130
C     RELEASING FIXED END FORCES FOR PINNED CONDITIONS      TABS80 4131
C                                                           TABS80 4132
      REAL K,I,MI,MJ                                        TABS80 4133
C                                                           TABS80 4134
C     ASSUMES PRISMATIC BEAM                                TABS80 4135
C                                                           TABS80 4136
      IF (K.EQ..0) GO TO 10                                 TABS80 4137
      IF (C.EQ..0) GO TO 10                                 TABS80 4138
      IF (I.EQ..0) GO TO 10                                 TABS80 4139
      GO TO 999                                             TABS80 4140
C                                                           TABS80 4141
C     PINS AT BOTH ENDS                                     TABS80 4142
C                                                           TABS80 4143
   10 TEMP=(MI+MJ)/XL                                       TABS80 4144
      MI=0.                                                 TABS80 4145
      MJ=0.                                                 TABS80 4146
C                                                           TABS80 4147
      IF (ITAG.EQ.2) GO TO 999                              TABS80 4148
C                                                           TABS80 4149
      VI=VI-TEMP                                            TABS80 4150
      VJ=VJ+TEMP                                            TABS80 4151
C                                                           TABS80 4152
  999 RETURN                                                TABS80 4153
      END                                                   TABS80 4154


      SUBROUTINE TABB (F,Q,LDB,RP,IFEF,FEF,CONL,N,M,NS,NA,NB,NBP,NFEF,  TABS80 4155
     *                 MCONL,XL,AAA,BBB,A,B,XLL,NAL)        TABS80 4156
C                                                           TABS80 4157
C     CORRECTIONS TO BEAM MOMENTS AND SHEARS DUE TO SPAN LOADS  TABS80 4158
C                                                           TABS80 4159
      DIMENSION LDB(NS,NB,4),RP(8,NRP)                      TABS80 4160
      DIMENSION IFEF(NFEF),FEF(7,NFEF),CONL(2,MCONL,NFEF)   TABS80 4161
      DIMENSION F(10,9)                                     TABS80 4162
      DIMENSION Q(4,4)                                      TABS80 4163
C                                                           TABS80 4164
      DO 500 L=1,4                                          TABS80 4165
      J=LDB(N,M,L)                                          TABS80 4166
      IF (J.LE.0) GO TO 500                                 TABS80 4167
C                                                           TABS80 4168
      F(1,L)=F(1,L)+Q(1,L)                                  TABS80 4169
      F(2,L)=F(2,L)+Q(3,L)                                  TABS80 4170
      F(4,L)=F(4,L)+Q(2,L)                                  TABS80 4171
      F(5,L)=F(5,L)+Q(4,L)                                  TABS80 4172
C                                                           TABS80 4173
C     SPAN MOMENT                                           TABS80 4174
C                                                           TABS80 4175
      F(3,L)=.5*(F(1,L)-F(2,L))                             TABS80 4176
                                                            TABS80 4177
```

```
      WW=FEF(5,J)                                                     TABS80 4178
      IF (L.EQ.NAL) WW=WW+BP(7,MB)*BP(8,MB)                           TABS80 4179
      VL=-F(4,L)                                                      TABS80 4180
      TEMP=SIGN(1.,VL)                                                TABS80 4181
      XBAR=0.                                                         TABS80 4182
      PZS=0.                                                          TABS80 4183
      NPASS=0                                                         TABS80 4184
      NCON=IFFF(J)                                                    TABS80 4185
      IF (NCON.EQ.0) GO TO 60                                         TABS80 4186
                                                                      TABS80 4187
      CALCULATING POINT OF ZERO SHEAR                                 TABS80 4188
                                                                      TABS80 4189
      DO 70 NCN=1,NCON                                                TABS80 4190
                                                                      TABS80 4191
      UNIFORM LOAD BEFORE NCN-TH CONCENTRATED LOAD                    TABS80 4192
                                                                      TABS80 4193
      AA=CONL(1,NCN,J)-B                                              TABS80 4194
      XDEL=AA-XBAR                                                    TABS80 4195
      VL=VL-WW*XDEL                                                   TABS80 4196
      XBAR=AA                                                         TABS80 4197
      IF (TEMP.NE.SIGN(1.,VL)) GO TO 60                               TABS80 4198
                                                                      TABS80 4199
      NCN-TH CONCENTRATED LOAD                                        TABS80 4200
                                                                      TABS80 4201
      VL=VL-CONL(2,NCN,J)                                             TABS80 4202
      NPASS=NCN                                                       TABS80 4203
      IF (TEMP.NE.SIGN(1.,VL)) GO TO 90                               TABS80 4204
                                                                      TABS80 4205
      UNIFORM LOAD AFTER LAST CONCENTRATED LOAD                       TABS80 4206
                                                                      TABS80 4207
      IF (NCN.NE.NCON) GO TO 70                                       TABS80 4208
      VL=VL-WW*(XL-XBAR)                                              TABS80 4209
      XBAR=XL                                                         TABS80 4210
      IF (TEMP.NE.SIGN(1.,VL)) GO TO 60                               TABS80 4211
   70 CONTINUE                                                        TABS80 4212
                                                                      TABS80 4213
      GO TO 500                                                       TABS80 4214
                                                                      TABS80 4215
      POINT OF ZERO SHEAR OCCURED DUE TO  --  UNIFORM LOAD            TABS80 4216
                                                                      TABS80 4217
   60 IF (ABS(WW).LT.1.E-10) GO TO 500                                TABS80 4218
      PZS=XBAR+VL/WW                                                  TABS80 4219
      IF (PZS.GE.XL) PZS=0.                                           TABS80 4220
      IF (PZS.LT..0) PZS=0.                                           TABS80 4221
      GO TO 100                                                       TABS80 4222
                                                                      TABS80 4223
      POINT OF ZERO SHEAR OCCURED DUE TO  --  CONCENTRATED LOAD       TABS80 4224
                                                                      TABS80 4225
   90 PZS=XBAR                                                        TABS80 4226
                                                                      TABS80 4227
```

```
  100 IF (PZS.EQ..0) GO TO 500                                        TABS80 4228
C                                                                     TABS80 4229
C     CALCULATING MAXIMUM POSITIVE OR MINIMUM NEGATIVE MOMENT         TABS80 4230
C                                                                     TABS80 4231
      F(3,L)=F(1,L)-F(4,L)*PZS-WW*PZS*PZS/2.                          TABS80 4232
      IF (NPASS.EQ.0) GO TO 500                                       TABS80 4233
      DO 110 NCN=1,NPASS                                              TABS80 4234
  110 F(3,L)=F(3,L)-(PZS+B-CONL(1,NCN,J))*CONL(2,NCN,J)               TABS80 4235
C                                                                     TABS80 4236
  500 CONTINUE                                                        TABS80 4237
C                                                                     TABS80 4238
      RETURN                                                          TABS80 4239
      END                                                             TABS80 4240


      SUBROUTINE TABW (R,XM,IXM,STOR,RO,FM,NN,MLD,MLD,NFQ,M,ITYP,     TABS80 4241
     *                 ITAG)                                          TABS80 4242
C                                                                     TABS80 4243
C     PRINTING FORCES AND STRESSES                                    TABS80 4244
C                                                                     TABS80 4245
      DIMENSION STOR(NFQ),RO(NFQ,NFQ)                                 TABS80 4246
      DIMENSION R(NN,MLD),XM(11,MLD),IXM(MLD),FR(10,NFQ)              TABS80 4247
      DIMENSION IFMT(6)                                               TABS80 4248
C                                                                     TABS80 4249
      COMMON /GENL / NST,NDF,NTF,KLD,NAT                              TABS80 4250
      COMMON /IMEDR/ IMED(28),IDATE(2),NPAGE(2),MAXLIN,LINE           TABS80 4251
      COMMON /FTAPE/ KTMP,KINP,KOUT,KSTR,NTAPE,KTAPE                  TABS80 4252
      COMMON /DYN  / NTIPE,DT,NPC,DAMP,NMD                            TABS80 4253
      COMMON /JUNK / FHED(4),IF,IFC,ND,NF,LM(6),CA(10,6),F(10,9),Q(4,4),TABS80 4254
     *               P(10),SY(6),T(6),SDI,LTYP                        TABS80 4255
C                                                                     TABS80 4256
      DATA IFMT /4H(I6,,4H4X  ,4H,2A6,4H,4X,,4H10F1,4H0.3)/           TABS80 4257
C                                                                     TABS80 4258
      IF (ITAG.EQ.2) GO TO 560                                        TABS80 4259
C                                                                     TABS80 4260
      LLD=MLD+NMD                                                     TABS80 4261
C                                                                     TABS80 4262
      NFORC=2*NF                                                      TABS80 4263
      DO 100 I=1,NFORC                                                TABS80 4264
      DO 100 J=1,9                                                    TABS80 4265
  100 F(I,J)=0.0                                                      TABS80 4266
C                                                                     TABS80 4267
C     CALCULATE STATIC FORCES                                         TABS80 4268
C                                                                     TABS80 4269
  150 DO 200 K=1,ND                                                   TABS80 4270
      KK=LP(K)                                                        TABS80 4271
      DO 200 I=1,NF                                                   TABS80 4272
      DO 200 J=1,6                                                    TABS80 4273
```

```
  200 F(I,J)=F(I,J)+CA(I,K)*R(KK,J)
C
C     CALCULATE DYNAMIC FORCES
C
      NATT=NAT-3
      IF (NATT) 550,250,250
  250 DO 500 I=1,NF
      DO 400 J=7,NLD
      X=0.0
      DO 300 K=1,ND
      KK=LM(K)
  300 X=X+CA(I,K)*R(KK,J)
      IF (NPD.NE.0) FM(I,J-6)=X
      XA=ABS(X)
      IF (NATT) 330,330,350
  330 F(I,8)=F(I,8)+XA
      F(I,7)=F(I,7)+X*X
      STOP(J-6)=X
      GO TO 400
  350 IF (XA.GT.F(I,9)) F(I,9)=XA
  400 CONTINUE
      IF (NAT.EQ.3) F(I,9)=COC(STOR,RO,NFO)
C
  500 F(I,7)=SQRT(F(I,7))
C
  550 GO TO 999
C
C     PRINT MEMBER FORCES FOR ALL LOAD CONDITIONS
C
  560 NSTR=NF+1
C
      WRITE (KTAPE) NF,((F(I,J),I=1,NF),J=1,9)
C
      DO 700 L=1,LLO
C
      IF (L.GT.NLO) GO TO 650
C
      IADD=IXM(L)
      T1=XM(10,L)
      T2=XM(11,L)
      DO 660 I=1,NFORC
      P(I)=0.
      DO 600 J=1,9
      IF (XM(J,L).EQ..0) GO TO 600
      PP=F(I,J)
      IF (IADD.GT.0) PP=ABS(PP)
      P(I)=P(I)+PP*XM(J,L)
  600 CONTINUE
  660 CONTINUE
      GO TO 750
C
C     MODAL SEPARATION
C
  650 T1=6H  MODE
      LL=L-NLO
      ENCODE (6,2000,T2) LL
      DO 760 I=1,NFORC
  760 P(I)=FM(I,L-NLO)
C
C     CHECKING RANGE FOR FORMAT
C
  750 IFMT(5)=4H10F1
      IFMT(6)=4H0.2)
      DO 10 I=1,NFORC
      IF (ABS(P(I)).LT.1.0E+5) GO TO 10
      IFMT(5)=4H10E1
      IFMT(6)=4H0.3)
      GO TO 20
   10 CONTINUE
C
   20 CALL NUPAJ (2,L,ITEST)
      IF (LTYP.EQ.L) ITEST=1
      IF (ITEST.EQ.0) GO TO 30
      CALL OUT (LTYP,KOUT,KSTR,FHED,SDI)
      LTYP=0
   30 WRITE (KOUT,IFMT) P,T1,T2,(P(I),I=1,NF)
      WRITE (KSTR,IFMT) P,T1,T2,(P(I),I=NSTR,NFORC)
C
  700 CONTINUE
C
      WRITE (KSTR,1000)
      WRITE (KOUT,1000)
      LINE=LINE+1
C
  999 RETURN
C
 1000 FORMAT (1H )
 2000 FORMAT (I6)
      END


      SUBROUTINE OUT (LTYP,KOUT,KSTR,FHED,SDI)
C
C     MEMBER FORCE AND STRESS OUTPUT HEADERS
C
      DIMENSION FHED(4)
      DIMENSION THED(2,4)
C
```

```
      DATA THED/5HCOLUM,5HN     ,5HBEAM ,5H     ,5HPANEL,5H     ,
     *          5HDIAGO,5HNAL   /
C
      CALL NUPAJ (2,6,ITEST)
C
      WRITE (KOUT,9998) THED(1,ITYP),THED(2,ITYP),SDI,FHED
      WRITE (KSTR,9999) THED(1,ITYP),THED(2,ITYP),SDI,FHED
C
      GO TO (1,2,3,4),ITYP
C
    1 WRITE (KOUT,1000)
      WRITE (KSTR,9001)
      GO TO 99
C
    2 WRITE (KOUT,2000)
      WRITE (KSTR,9002)
      GO TO 99
C
    3 WRITE (KOUT,3000)
      WRITE (KSTR,9003)
      GO TO 99
C
    4 WRITE (KOUT,4000)
      WRITE (KSTR,9004)
      GO TO 99
C
   99 RETURN
C
 1000 FORMAT (//
     *        1X5HCOLMN8X4HLOAD12X6HBOTTOM7X3HTOP5X5HAXIAL5X5HSHEAR/
     *        4X2HNO3X14HIDENTIFICATION7X6HMOMENT4X6HMOMENT2(5X5HFORCE))
 2000 FORMAT (//
     *        2X4HREAP8X4HLOAD14X4HLEFT5X5HRIGHT6X4HSPAN6X4HLEFT
     *        5X5HRIGHT/
     *        4X2HNO3X14HIDENTIFICATION3X3(4X6HMOMENT)2(5X5HSHEAR))
 3000 FORMAT (//
     *        1X5HPANEL8X4HLOAD12X6HBOTTOM7X3HTOP5X5HAXIAL5X5HSHEAR/
     *        4X2HNO3X14HIDENTIFICATION7X6HMOMENT4X6HMOMENT2(5X5HFORCE))
 4000 FORMAT (//
     *        1X5H DIAG8X4HLOAD12X6HBOTTOM7X3HTOP5X5HAXIAL5X5HSHEAR/
     *        4X2HNO3X14HIDENTIFICATION7X6HMOMENT4X6HMOMENT2(5X5HFORCE))
 9001 FORMAT (//
     *        1X5HCOLMN8X4HLOAD10X8HBOT-BEND2X8HTOP-BEND5X5HAXIAL
     *        5X5HSHEAR/
     *        4X2HNO3X14HIDENTIFICATION3X4(4X6HSTRESS))
 9002 FORMAT (//
     *        2X4HBEAM8X4HLOAD11X7HLT-BEND3X7HRT-BEND3X7HSP-BEND
     *        2X8HLT-SHEAR2X8HRT-SHEAR/
     *        4X2HNO3X14HIDENTIFICATION3X5(4X6HSTRESS))
 9003 FORMAT (//
     *        1X5HPANEL8X4HLOAD10X8HBOT-BEND2X8HTOP-BEND5X5HAXIAL
     *        5X5HSHEAR/
     *        4X2HNO3X14HIDENTIFICATION3X4(4X6HSTRESS))
 9004 FORMAT (//
     *        1X5H DIAG8X4HLOAD10X8HBOT-BEND2X8HTOP-BEND5X5HAXIAL
     *        5X5HSHEAR/
     *        4X2HNO3X14HIDENTIFICATION3X4(4X6HSTRESS))
 9998 FORMAT (/1X,A5,A4,16HFORCES AT LEVEL A5,4H IN ,9A5)
 9999 FORMAT (/1X,A5,A4,18HSTRESSES AT LEVEL A5,4H IN ,9A5)
      END


      SUBROUTINE TABJ (SD,VV,NSF,FHED,IIF,NST,NTF)
C
C     GENERATING STORY SHEAR DISTRIBUTION SUMMARY STORY BY STORY
C
      DIMENSION SD(NST,1),VV(NTF,NST,6),NSF(NTF),FHED(NTF,4),IIF(NTF,2)
C
      COMMON /IHEDR/ IHED(20),IDATE(2),NPAGE(2),MAXLIN,LINE
      COMMON /FTAPE/ KTMP,KINP,KOUT,KSTR,NTAPE
C
      REWIND NTAPE
C
      DO 200 II=1,NTF
      I=NTF-II+1
      READ (NTAPE) IF,IFC,(FHED(I,J),J=1,4)
      IIF(I,1)=IF
      IIF(I,2)=IFC
      IF (IFC.EQ.1) GO TO 200
      READ (NTAPE) NS
      NSF(I)=NS
      DO 100 NN=1,NS
      N=NST-NN+1
      READ (NTAPE) (VV(I,N,J),J=1,6)
  100 CONTINUE
  200 CONTINUE
C
      LINE=99
      CALL NUPAJ (1,7,ITEST)
      WRITE (KOUT,1000)
C
      DO 400 LN=1,NST
      N=NST-LN+1
      CALL NUPAJ (1,2,ITEST)
      IF (ITEST.EQ.0) GO TO 250
      LINE=LINE+7
      WRITE (KOUT,1000)
  250 WRITE (KOUT,3000) SD(N,1)
```

```
C                                                                       TABS80 4466
      DO 300 I=1,NTF                                                    TABS80 4467
      IF (ITF(I,2).NE.0) GO TO 300                                      TABS80 4468
      IF (LN.GT.NSF(I))  GO TO 300                                      TABS80 4469
      CALL NUPAJ (I,1,ITEST)                                            TABS80 4470
      IF (ITEST.EQ.0) GO TO 350                                         TABS80 4471
      LINE=LINE+9                                                       TABS80 4472
      WRITE (ROUT,1000)                                                 TABS80 4473
      WRITE (ROUT,3000) SD(N,1)                                         TABS80 4474
  350 WRITE (ROUT,2000) (FHED(I,J),J=1,4),(VV(I,N,J),J=1,6)             TABS80 4475
C                                                                       TABS80 4476
  300 CONTINUE                                                          TABS80 4477
  400 CONTINUE                                                          TABS80 4478
C                                                                       TABS80 4479
      RETURN                                                            TABS80 4480
C                                                                       TABS80 4481
 1000 FORMAT (//                                                        TABS80 4482
     *          45H SUMMARY OF STORY SHEAR DISTRIBUTION          /      TABS80 4483
     *          45H STORY-BY-STORY / FRAME-BY-FRAME              //     TABS80 4484
     *          1X5HLEVEL22X1H/19(1H-)16HLOAD  CONDITIONS13(1H-)1H//    TABS80 4485
     *          4X2HID2X20H/--FRAME LOCATION--/                         TABS80 4486
     *          8X1HI7X2HI(6X3HI(1)7X2HIV8X1HA8X1HB)                    TABS80 4487
 2000 FORMAT (8X,4A5,6F9.2)                                             TABS80 4488
 3000 FORMAT (1H0,A5)                                                   TABS80 4489
      END                                                               TABS80 4490


      SUBROUTINE FFRAME (NSC,SD,X,Y,DD,NST,NS,NC,IF,ITAG)               TABS80 4491
C                                                                       TABS80 4492
C     PLOTTING FRAME ELEVATION                                          TABS80 4493
C                                                                       TABS80 4494
      COMMON /PLOTS/ XO,YO,XDMX,XDMN,XDIM,YDIM,SZ,B1,B2,Q,RAD           TABS80 4495
                                                                        TABS80 4496
      DIMENSION SD(NST),X(NC),Y(NS)                                     TABS80 4497
      DIMENSION NSC(NST,1)                                              TABS80 4498
      DIMENSION DD(4)                                                   TABS80 4499
C                                                                       TABS80 4500
      IF (ITAG.EQ.2) GO TO 200                                          TABS80 4501
C                                                                       TABS80 4502
  100 CALL FSCALE (SD,X,Y,DD,NST,NC,NS,IF)                              TABS80 4503
      CALL FGRID (NSC,SD,X,Y,DD,NST,NC,NS,IF)                           TABS80 4504
      GO TO 999                                                         TABS80 4505
C                                                                       TABS80 4506
  200 XO=0.                                                             TABS80 4507
      YO=0.                                                             TABS80 4508
      CALL CALCOM (XDIM+6.,0.,8,8,8,4,-3,2)                             TABS80 4509
      GO TO 999                                                         TABS80 4510
C                                                                       TABS80 4511
  999 RETURN                                                            TABS80 4512
C                                                                       TABS80 4513
      END                                                               TABS80 4514


      SUBROUTINE FSCALE (SD,X,Y,D,NST,NC,NS,IF)                         TABS80 4515
C                                                                       TABS80 4516
C     CALCULATING SCALE OF FRAME ELEVATION PLOT                         TABS80 4517
C                                                                       TABS80 4518
      COMMON /PLOTS/ XO,YO,XDMX,XDMN,XDIM,YDIM,SZ,B1,B2,Q,RAD           TABS80 4519
      DIMENSION SD(NST),X(NC),Y(NS)                                     TABS80 4520
      DIMENSION D(4)                                                    TABS80 4521
C                                                                       TABS80 4522
      XMIN=-D(1)+X(1)                                                   TABS80 4523
      XMAX= D(2)+X(NC)                                                  TABS80 4524
      YMIN=-D(4)                                                        TABS80 4525
      YMAX= D(3)+Y(1)                                                   TABS80 4526
C                                                                       TABS80 4527
      CX=4.*B1+2.*B2+2.*(SZ+.05)                                        TABS80 4528
      CY=3.*B1                                                          TABS80 4529
C                                                                       TABS80 4530
      CALL CSCALE (CX,CY,XMIN,XMAX,YMIN,YMAX)                           TABS80 4531
C                                                                       TABS80 4532
C     SCALE TRANSFORMATION                                              TABS80 4533
C                                                                       TABS80 4534
      DO 30 I=1,4                                                       TABS80 4535
   30 D(I)=D(I)/Q                                                       TABS80 4536
      DO 70 N=1,NS                                                      TABS80 4537
   70 Y(N)=Y(N)/Q                                                       TABS80 4538
      DO 80 M=1,NC                                                      TABS80 4539
   80 X(M)=X(M)/Q                                                       TABS80 4540
C                                                                       TABS80 4541
      CALL FBOX (IF)                                                    TABS80 4542
C                                                                       TABS80 4543
      XO=.5*(XDIM-CX-(XMAX-XMIN))+D(1)+2.*B1                            TABS80 4544
      YO=.5*(YDIM-CY-(YMAX-YMIN))+D(4)+2.*B1                            TABS80 4545
C                                                                       TABS80 4546
      RETURN                                                            TABS80 4547
      END                                                               TABS80 4548


      SUBROUTINE FBOX (I)                                               TABS80 4549
C                                                                       TABS80 4550
C     BORDERLINE PLOT                                                   TABS80 4551
C                                                                       TABS80 4552
      COMMON /PLOTS/ XO,YO,XDMX,XDMN,XDIM,YDIM,SZ,B1,B2,Q,RAD,FHED(4)   TABS80 4553
```

```
C
      DATA TABS80 /6HTABS80/
      DATA FRAME  /6H FRAME/
      DATA TYPE   /6H TYPE /
C
      CALL BOX
C
      XX=0.
      YY=-.05-SZ
      CALL CALCOM (XX,YY,SZ,TABS80,R,0.,6,3)
      XX=XX+12.*SZ
      CALL CALCOM (XX,YY,SZ,FRAME,B,0.,6,3)
      XX=XX+6.*SZ
      CALL CALCOM (XX,YY,SZ,TYPE,R,0.,6,3)
      XX=XX+6.*SZ
      B=FLOAT(I)
      CALL CALCOM (XX,YY,SZ,B,R,0.,-1,4)
      XX=XX+6.*SZ
C
      DO 10 J=1,14
      CALL CALCOM (XX,YY,SZ,FMED(J),B,0.,5,3)
   10 XX=XX+5.*SZ
C
      RETURN
      END


      SUBROUTINE FGRID (NSC,SD,X,Y,D,NST,NC,NS,IF)
C
C     PLOTTING RECTANGULAR GRID OF FRAME
C
      DIMENSION SD(NST),X(NC),Y(NS)
      DIMENSION NSC(NST,1)
      DIMENSION D(4)
C
      COMMON /PLOTS/ XO,YO,XDMX,XDMN,XDIM,YDIM,SZ,B1,B2,Q,RAD
C
      DATA SZ/.05/
      DATA S /.10/
      DATA C /1HC/
      DATA B /1HB/
C
C     COLUMN LINES
C
      DO 10 M=1,NC
      X1=X(M)-S
      Y1=-B2-2.*S
      CALL CALCOM (X1,Y1,S,C,R,0.,1,1)
      X1=X1+S
      R=FLOAT(M)
      CALL CALCOM (X1,Y1,S,B,R,0.,-1,4)
C
      X1=X(M)
      Y1=-B2
      X2=X1
      Y2=Y(1)+D(3)+B2
   10 CALL LYNE (X1,Y1,X2,Y2,1)
C
C     STORY LINES
C
      DO 20 N=1,NS
      X1=-B2-D(1)
      Y1=Y(N)+.05
      NL=NSC(N+NST-NS,IF)
      R=FLOAT(NST-NL+1)
      CALL CALCOM (X1,Y1,SZ,R,R,0.,-1,4)
C
      X1=-B2-D(1)
      X2=X(NC)+D(2)+B1
      Y1=Y(N)
      Y2=Y1
      CALL LYNE (X1,Y1,X2,Y2,1)
C
      X1=X(NC)+D(2)+.25
      Y1=Y(N)+.05
      NL=NSC(N+NST-NS,IF)
   20 CALL CALCOM (X1,Y1,SZ,SD(NL),B,0.,5,3)
C
C     BASE LINE
C
      X1=-B2-D(1)
      Y1=.05
      R=0.
      CALL CALCOM (X1,Y1,SZ,R,R,0.,-1,4)
C
      X1=-B2-D(1)
      X2=X(NC)+D(2)+B1
      Y1=0.
      Y2=0.
      CALL LYNE (X1,Y1,X2,Y2,1)
C
      IF (NB.EQ.0) GO TO 999
C
      NR=NC-1
      Y1=-B2/2.
      Y2=Y1+.05
      CALL LYNE (X(1),Y1,X(NC),Y1,1)
C
```

```
      DO 30 M=1,NB
      J=6
      X1=X(M)+S2
      X2=X(M+1)-S2
      XX=(X1+X2)/2.-S
      CALL CALCOM (X1,Y1,S,J,R,90.,-1,3)
      CALL CALCOM (XX,Y2,S,B,B,0.,1,3)
      R=FLOAT(M)
      XX=XX+S
      CALL CALCOM (XX,Y2,S,B,R,0.,-1,4)
   30 CALL CALCOM (X2,Y1,S,J,R,-90.,-1,3)
C
  999 RETURN
C
      END


      SUBROUTINE ELPLOT (X1,X2,Y1,Y2,DD,D1,D2,JD,IPLT,IGOTO)
C
C     ELEMENT PLOT CONTROL ROUTINE
C
      COMMON /PLOTS/ XO,YO,XOMX,XOMN,XDIM,YDIM,SZ,B1,B2,Q
      DIMENSION DD(4)
      DIMENSION JD(6)
C
      DO 1 I=1,4
    1 DD(I)=DD(I)/Q
      W1=D1/Q
      W2=D2/Q
C
      GO TO (10,20,30,40,50),IGOTO
C
   10 CALL FCOLM (X1,Y1,Y2,DD,JD,IPLT,W1)
      GO TO 999
C
   20 CALL FBEAM (Y1,X1,X2,DD,JD,IPLT,W1,W2,1)
      GO TO 999
C
   30 CALL FPANL (X1,X2,Y1,Y2,W1,JD,IPLT)
      GO TO 999
C
   40 CALL FDIAG (X1,X2,Y1,Y2,W1,JD,IPLT)
      GO TO 999
C
   50 CALL FBEAM (Y1,X1,X2,DD,JD,IPLT,W1,W2,2)
      GO TO 999
C
  999 RETURN
C
      END


      SUBROUTINE FCOLM (X,YB,YT,DD,JD,IPLT,W)
C
C     PLOTTING COLUMNS
C
      DIMENSION DD(4)
      DIMENSION JD(5)
C
      DATA S /.1/
      DATA T /1HT/
      DATA TOL /1.E-10/
C
      W=W/2.
C
      X1=X-W
      X2=X1
      Y1=YB+DD(3)
      Y2=YT-DD(1)
      YY=.5*(Y1+Y2)
      IF (W.LE.TOL) GO TO 10
      CALL LYNE (X1,Y1,X2,Y2,1)
C
   10 X1=X+W
      X2=X1
      Y1=YB+DD(4)
      Y2=YT-DD(2)
      IF (W.LE.TOL) GO TO 777
      CALL LYNE (X2,Y2,X1,Y1,1)
C
  777 IF (IPLT.NE.2) GO TO 999
      R=FLOAT(JD(5))
      X1=X+W+S
      Y1=YY
      CALL CALCOM (X1,Y1,S,T,R,.0,1,3)
      X1=X1+S
      CALL CALCOM (X1,Y1,S,B,R,0.,-1,4)
C
  999 RETURN
C
      END


      SUBROUTINE FBEAM (Y,XL,XR,DD,JD,IPLT,D1,D2,ITAC)
```

```
C
C     PLOTTING BEAMS
C
      DIMENSION DD(4)
      DIMENSION JD(6)
C
      DATA TOL /1.E-10/
      DATA C /1H./
      DATA T /1HT/
      DATA S /0.1/
      DATA PR /1H//
      DATA PL /2HL//
C
      IF (ITAG.EQ.2) GO TO 20
C
      X1=XL+DD(4)
      X2=XR-DD(2)
      Y1=Y-D1
      Y2=Y1
      XX=.5*(X1+ X2)
      IF (D1.LT.TOL) GO TO 20
      CALL LYNE (X1,Y1,X2,Y2,1)
C
   20 X1=XL+DD(3)
      X2=XR-DD(1)
      Y1=Y+D2
      Y2=Y1
      IF (D2.LT.TOL) GO TO 777
      CALL LYNE (X2,Y2,X1,Y1,1)
C
  777 IF (ITAG.EQ.2) GO TO 999
      IF (IPLT.NE.2) GO TO 999
C
      R=FLOAT(JD(6))
      X1=XX-2.*S
      Y1=Y+D2+.5*S
      CALL CALCOM (X1,Y1,S,T,B,.0,1,3)
      X1=X1+S
      CALL CALCOM (X1,Y1,S,R,R,0.,-1,4)
C
      IF (JD(5).EQ.0) GO TO 999
      X1=XX-4.*S
      Y1=Y-D1-1.5*S
      CALL CALCOM (X1,Y1,S,PL,B,0.,2,3)
      X1=X1+2.*S
      DO 30 I=1,4
      N=JD(I)
      A=1.
      IF (N.LT.10) GO TO 40
      A=FLOAT(N)
      A=ALOG10(A)
      A=A+1.
   40 R=FLOAT(JD(I))
      CALL CALCOM (X1,Y1,S,B,R,0.,-1,4)
      X1=X1+A*S
      IF (I.EQ.4) GO TO 30
      CALL CALCOM (X1,Y1,S,C,B,0.,1,3)
      X1=X1+S
   30 CONTINUE
      CALL CALCOM (X1,Y1,S,PR,B,0.,1,3)
C
  999 RETURN
C
      END
```

```
      SUBROUTINE FPANL (XL,XR,Y1,Y2,W,JD,IPLT)
C
C     PLOTTING PANELS
C
      DIMENSION JD(5)
C
      DATA T /1HT/
      DATA P/1HP/
      DATA S /.10/
      DATA S2/.05/
C
      W=JD(4)
      W=W/2.
      X=(XL+XR)/2.
      Y=(Y1+Y2)/2.
C
      X1=X-W
      IF (X1.GT.XL) X1=XL
      X2=X+W
      IF (X2.LT.XR) X2=XR
C
      IF (X1.NE.XL)
     *CALL LYNE (X1,Y1,X1,Y2,1)
      IF (X2.NE.XR)
     *CALL LYNE (X2,Y1,X2,Y2,1)
C
C      LABELING PANEL
C
      J=6
      CALL CALCOM (X1+S2,Y-S,J,R,90.,-1,3)
      CALL LYNE (X1+S,Y,X-S2,Y,2)
      J=5
```

```
      CALL CALCOM (X,Y,S,J,R,0.,-1,3)
      CALL CALCOM (X+S,Y+S2,S,P,B,0.,1,3)
      R=FLOAT(P)
      CALL CALCOM (X+2*S,Y+S2,S,B,R,0.,-1,4)
C
      IF (IPLT.NE.2) GO TO 777
      R=FLOAT(JD(5))
      X1=X+S
      Y1=Y-S-S2
      CALL CALCOM (X1,Y1,S,T,B,0.,1,3)
      X1=X1+S
      CALL CALCOM (X1,Y1,S,B,R,0.,-1,4)
C
  777 CALL LYNE (X+S2,Y,X2-S,Y,2)
      J=6
      CALL CALCOM (X2-S2,Y,S,J,R,-90.,-1,3)
C
  999 RETURN
C
      END


      SUBROUTINE FDIAG (X1,X2,Y1,Y2,W,JD,IPLT)
C
C     PLOTTING DIAGONALS
C
      COMMON /PLOTS/ XO,YO,XDMX,XDMN,XDIM,YDIM,SZ,B1,B2,Q,RAD
C
      DIMENSION X(4),Y(4)
      DIMENSION JD(5)
C
      DATA S /.10/
      DATA D /1HD/
      DATA T /1HT/
      DATA TOL /1.E-10/
C
      M=JD(4)
      W=W/2.
C
      DX=X2-X1
      DY=Y2-Y1
      DL=SQRT(DX*DX+DY*DY)
      THETA=ATAN(DY/DX)
      TT=THETA/RAD
      COST=DX/DL
      SINT=DY/DL
C
      XX=X1+DX/4.
      YY=Y1+DY/4.
      DX=W*SINT
      DY=W*COST
C
      X(1)=X1+DX
      X(2)=X2+DX
      X(3)=X1-DX
      X(4)=X2-DX
      Y(1)=Y1-DY
      Y(2)=Y2-DY
      Y(3)=Y1+DY
      Y(4)=Y2+DY
C
      DX=XX
      DY=YY
C
      CALL LYNE (X(3),Y(3),X(4),Y(4),1)
      IF (W.LE.TOL) GO TO 100
      CALL LYNE (X(4),Y(4),X(2),Y(2),1)
      CALL LYNE (X(2),Y(2),X(1),Y(1),1)
      CALL LYNE (X(1),Y(1),X(3),Y(3),1)
C
C     LABELING DIAGONAL
C
  100 SINT=SIN(THETA)
      COST=COS(THETA)
      XX=XX-(W+.05)*SINT
      YY=YY+(W+.05)*COST
      CALL CALCOM (XX,YY,S,D,B,TT,1,3)
      XX=XX+S*COST
      YY=YY+S*SINT
      R=FLOAT(M)
      CALL CALCOM (XX,YY,S,B,R,TT,-1,4)
C
      IF (IPLT.NE.2) GO TO 999
      XX=DX+(W+1.5*S)*SINT
      YY=DY-(W+1.5*S)*COST
      CALL CALCOM (XX,YY,S,T,B,TT,1,3)
      XX=XX+S*COST
      YY=YY+S*SINT
      R=FLOAT(JD(5))
      CALL CALCOM (XX,YY,S,B,R,TT,-1,4)
C
  999 RETURN
C
      END
```

```
      SUBROUTINE SFRAME
C
C     PLOTTING STRUCTURAL PLAN
C
      COMMON /PLOTS/ XO,YO,XDMX,XDMN,XDIM,YDIM,SZ,B1,B2,Q
C
      CALL SSCALE (XMIN,YMIN)
C
      CALL SPLOT (XMIN,YMIN)
C
      XO=0.
      YO=0.
      CALL CALCOM (XDIM+6.,0.,B,8,B,8,-3,2)
C
      RETURN
      END


      SUBROUTINE SSCALE (XMIN,YMIN)
C
C     PLAN PLOT SCALE
C
      COMMON /GEN1 / NST,NDF,NTF
      COMMON /PLOTS/ XO,YO,XDMX,XDMN,XDIM,YDIM,SZ,B1,B2,Q
C
      XMIN= 1.E20
      YMIN= 1.E20
      XMAX=-1.E20
      YMAX=-1.E20
C
      CX=2.*B1+2.*B2+2.*(SZ+.05)
      CY=2.*B1
C
      REWIND 1
C
      DO 10 I=1,NTF
      READ (1) IF,IFC,X1,Y1,X2,Y2
      IF (X1.GT.XMAX) XMAX=X1
      IF (X2.GT.XMAX) XMAX=X2
      IF (X1.LT.XMIN) XMIN=X1
      IF (X2.LT.XMIN) XMIN=X2
      IF (Y1.GT.YMAX) YMAX=Y1
      IF (Y2.GT.YMAX) YMAX=Y2
      IF (Y1.LT.YMIN) YMIN=Y1
      IF (Y2.LT.YMIN) YMIN=Y2
   10 CONTINUE
C
      CALL CSCALE (CX,CY,XMIN,XMAX,YMIN,YMAX)
C
      CALL SBOX
C
      XO=0.5*(XDIM-CX-(XMAX-XMIN))+B1
      YO=0.5*(YDIM-CY-(YMAX-YMIN))+B1
C
      RETURN
      END


      SUBROUTINE SBOX
C
C     BORDER LINE OF STRUCTURAL PLAN
C
      COMMON /PLOTS/ XO,YO,XDMX,XDMN,XDIM,YDIM,SZ,B1,B2,Q
C
      DIMENSION L(13)
      DATA L /6HPLAN V,6HIEW OF,6H FRAME,6HS, DIS,6HPLAYIN,6HG FRAM,
     *        6HE NUMB,6HERS (N,6HM), TYP,6HES (T),6H AND L,6HOCAL A,
     *        6HXES ( /
      DATA TABS80 /6HTABS80/
C
      CALL BOX
C
      XX=0.
      YY=-0.05-SZ
      CALL CALCOM (XX,YY,SZ,TABS80,8,0.,6,3)
      XX=XX+12.*SZ
C
      DO 10 I=1,13
      CALL CALCOM (XX,YY,SZ,L(I),8,0.,6,3)
   10 XX=XX+6.*SZ
C
      J=6
      XX=XX-.5*SZ
      CALL CALCOM (XX,YY+.5*SZ,SZ,J,9,T,-1,3)
      J=1H)
      XX=XX+SZ
      CALL CALCOM (XX,YY,SZ,J,B,0.,1,3)
C
      RETURN
      END


      SUBROUTINE SPLOT (XMIN,YMIN)
C
```

```
C     PLOTTING PLAN
C
      COMMON /GEN1 / NST,NOF,NTF
      COMMON /PLOTS/ XO,YO,XDMX,XDMN,XDIM,YDIM,SZ,B1,B2,Q,RAD
C
      DATA NN /1HN/
      DATA TT /1HT/
      DATA S  /.10/
      DATA TOL/1.E-8/
C
      REWIND 1
C
      DO 10 I=1,NTF
      READ (1) IF,IFC,X1,Y1,X2,Y2
C
      X1=X1/Q-XMIN
      Y1=Y1/Q-YMIN
      X2=X2/Q-XMIN
      Y2=Y2/Q-YMIN
      X=(X1+X2)/2.
      Y=(Y1+Y2)/2.
C
      J=11
      CALL CALCOM (X1,Y1,.07,J,R,0.,-1,3)
      CALL LYNE (X1,Y1,X2,Y2,1)
C
      DX=X2-X1
      DY=Y2-Y1
      DL=SQRT(DX*DX+DY*DY)
      CC=DX/DL
      SS=DY/DL
      T=ACOS(CC)/RAD
      IF (SS.LT..0) T=360.-T
C
      J=6
      CALL CALCOM (X,Y,S,J,B,T-90.,-1,3)
C
      IF (ABS(DX).GT.TOL) GO TO 30
      T=90.*RAD
      GO TO 20
   30 T=ATAN(DY/DX)
   20 CC=COS(T)
      SS=SIN(T)
      T=T/RAD
C
      R=FLOAT(I)
      XX=X+S*CC-.5*S*SS
      YY=Y+S*SS+.5*S*CC
      CALL CALCOM (XX,YY,S,NN,R,T,1,3)
      XX=XX+S*CC
      YY=YY+S*SS
      CALL CALCOM (XX,YY,S,B,R,T,-1,4)
C
      R=FLOAT(IF)
      XX=X+S*CC+1.5*S*SS
      YY=Y+S*SS-1.5*S*CC
      CALL CALCOM (XX,YY,S,TT,B,T,1,3)
      XX=XX+S*CC
      YY=YY+S*SS
      CALL CALCOM (XX,YY,S,B,R,T,-1,4)
   10 CONTINUE
C
      RETURN
      END




      SUBROUTINE CSCALE (CX,CY,XMIN,XMAX,YMIN,YMAX)
C
C     CALCULATING SCALE
C
      COMMON /PLOTS/ XO,YO,XDMX,XDMN,XDIM,YDIM,SZ,B1,B2,Q
C
      DATA TOL /1.E-15/
C
      XDIM=XDMN
      XX=XDIM-CX
      YY=YDIM-CY
C
      SX=ABS(XMIN-XMAX)/XX
      SY=ABS(YMIN-YMAX)/YY
C
      Q=SY
      IF (SY.GT.SX) GO TO 50
C
      XDIM=XDMX
      IF (SY.LT.TOL) GO TO 80
C
      XDIM=SX*XX/SY+CX
      IF (XDIM.GT.XDMX) XDIM=XDMX
C
   80 XX=XDIM-CX
      Q=ABS(XMIN-XMAX)/XX
C
   50 XMIN=XMIN/Q
      YMIN=YMIN/Q
      XMAX=XMAX/Q
      YMAX=YMAX/Q
C
```

```
      RETURN
      END

      SUBROUTINE LYNE (X1,Y1,X2,Y2,ITAG)
C
C     PLOTTING LINE BETWEEN POINTS 1 AND 2
C
      IF (ITAG.EQ.2) GO TO 20
C
   10 CALL CALCOM (X1,Y1,8,8,8,8,3,2)
      CALL CALCOM (X2,Y2,8,8,8,8,2,2)
      GO TO 999
C
   20 CALL DASH (X1,Y1,X2,Y2)
C
  999 RETURN
      END

      SUBROUTINE DASH (XI,YI,XJ,YJ)
C
C     PLOTTING A DASHED LINE BETWEEN POINT I AND POINT J
C
      DIMENSION DX(3),DY(3),DL(3)
      DATA DELTD /0.1/
C
C
      AX=XJ-XI
      AY=YJ-YI
      AL=SQRT(AX*AX+AY*AY)
      IF (AL-0.005) 20,20,60
C
   60 COST=AX/AL
      SINT=AY/AL
      DX(3)=DELTD*COST
      DY(3)=DELTD*SINT
      DL(3)=DELTD
      DX(1)=0.5*DX(3)
      DY(1)=0.5*DY(3)
      DL(1)=0.5*DL(3)
C
      I=-1
      XF=XI
      YF=YI
      XL=0.0
C
   10 I=-I
      JJ=I+2
      XL=XL+DL(JJ)
      IF (XL.GT.AL) GO TO 30
      GO TO 40
   30 DL(JJ)=DL(JJ)-XL+AL
      XL=AL+0.005
      DX(JJ)=DL(JJ)*COST
      DY(JJ)=DL(JJ)*SINT
   40 XS=XF
      YS=YF
      XF=XF+DX(JJ)
      YF=YF+DY(JJ)
      IF (I.LT.0) GO TO 50
C
      CALL CALCOM (XS,YS,8,8,8,8,3,2)
      CALL CALCOM (XF,YF,8,8,8,8,2,2)
C
   50 IF (XL-AL) 10,20,20
C
   20 RETURN
      END

      SUBROUTINE BOX
C
      COMMON /IHEDR/ IHED(20)
      COMMON /PLOTS/ XO,YO,XOMX,XOMN,XDIM,YDIM,SZ,B1,B2,Q
C
      XO=0.
      YO=0.
C
C     PLOTTING BORDER LINE
C
      CX=2.*B2+2.*(SZ+.05)
C
      X1=XDIM-CX
      Y1=0.
      X2=X1
      Y2=YDIM
      CALL LYNE (X1,Y1,X2,Y2,1)
C
      X1=XDIM
      Y1=YDIM
      CALL LYNE (X2,Y2,X1,Y1,1)
C
      X2=XDIM
```

```
      Y2=0.
      CALL LYNE (X1,Y1,X2,Y2,1)
C
      X1=0.
      Y1=0.
      CALL LYNE (X2,Y2,X1,Y1,1)
C
      X2=0.
      Y2=YDIM
      CALL LYNE (X1,Y1,X2,Y2,1)
C
      X1=XDIM-CX
      Y1=YDIM
      CALL LYNE (X2,Y2,X1,Y1,1)
C
      X1=X1+.025
      Y1=YDIM
      X2=X1
      Y2=0.
      CALL LYNE (X1,Y1,X2,Y2,1)
C
C     PLOTTING THE TWO HEADING CARDS
C
      X=XDIM-CX+B2+SZ
      LOC=0
      DO 10 N=1,2
      Y=B1
      DO 20 I=1,14
      LOC=LOC+1
      CALL CALCOM (X,Y,SZ,IHED(LOC),8,90.,5,3)
   20 Y=Y+5.*SZ
   10 X=X+SZ+.05
C
      RETURN
      END


      SUBROUTINE CALCOM (X,Y,SZ,IDIM,R,TANT,KO,KODE)
C
C     SUBROUTINE CALLING CALCOMP PLOTTING ROUTINES
C
      COMMON /PLOTS/ XO,YO
C
      DIMENSION IDIM(1)
C
      XX=X+XO
      YY=Y+YO
C
      GO TO (10,20,30,40),KODE
C
   10 JDIM=IFIX(R)
      CALL PLOTS (IDIM,JDIM,KO)
      GO TO 100
C
   20 CALL PLOT (XX,YY,KO)
      GO TO 100
C
   30 CALL SYMBOL (XX,YY,SZ,IDIM,TANT,KO)
      GO TO 100
C
   40 CALL NUMBER (XX,YY,SZ,R,TANT,KO)
      GO TO 100
C
  100 RETURN
C
      END
```

In accordance with letter from DAEN-RDC, DAEN-ASI dated 22 July 1977, Subject: Facsimile Catalog Cards for Laboratory Technical Publications, a facsimile catalog card in Library of Congress MARC format is reproduced below.

Wilson, Edward L.
Theoretical basis for CTABS80, a computer program for three-dimensional analysis of building systems / by Edward L. Wilson, H.H. Dovey, Ashraf Habibullah (Computer/Structures International, Oakland, Calif.). -- Vicksburg, Miss. : U.S. Army Engineer Waterways Experiment Station ; Springfield, Va. : available from NTIS, 1981.
72, 56 p. : ill. ; 27 cm. -- (Technical report / U.S. Army Engineer Waterways Experiment Station ; K-81-2)
Cover title.
"September 1981."
Final report.
"Prepared for Office, Chief of Engineers, U.S. Army."
"Monitored by Automatic Data Processing Center, U.S. Army Engineer Waterways Experiment Station."
"A report under the Computer-Aided Structural Engineering (CASE) Project."
Bibliography: p. 70-72.

Wilson, Edward L.
Theoretical basis for CTABS80, a computer program : ... 1981.
(Card 2)

1. Buildings. 2. Computer programs. 3. CTABS80 (Computer program). 4. Structural frames. 5. Structures, Theory of. I. Dovey, H.H. II. Habibullah, Ashraf. III. United States. Army. Corps of Engineers. Office of the Chief of Engineers. IV. Computer/Structures International. V. U.S. Army Engineer Waterways Experiment Station. Automatic Data Processing Center. VI. Title VII. Series: Technical report (U.S. Army Engineer Waterways Experiment Station) ; K-81-2.
TA7.W34 no.K-81-2

WATERWAYS EXPERIMENT STATION REPORTS PUBLISHED UNDER THE COMPUTER-AIDED STRUCTURAL ENGINEERING (CASE) PROJECT

	Title	Date
Technical Report K-78-1	List of Computer Programs for Computer-Aided Structural Engineering	Feb 1978
Instruction Report O-79-2	User's Guide: Computer Program with Interactive Graphics for Analysis of Plane Frame Structures (CFRAME)	Mar 1979
Technical Report K-80-1	Survey of Bridge-Oriented Design Software	Jan 1980
Technical Report K-80-2	Evaluation of Computer Programs for the Design Analysis of Highway and Railway Bridges	Jan 1980
Instruction Report K-80-1	User's Guide: Computer Program for Design Review of Curvilinear Conduits/Culverts (CURCON)	Feb 1980
Instruction Report K-80-3	A Three-Dimensional Finite Element Data Edit Program	Mar 1980
Instruction Report K-80-4	A Three-Dimensional Stability Analysis/Design Program (3DSAD)	
	Report 1: General Geometry Module	Jun 1980
Instruction Report K-80-6	Basic User's Guide: Computer Program for Design and Analysis of Inverted-T Retaining Walls and Floodwalls (TWDA)	Dec 1980
Instruction Report K-80-7	User's Reference Manual: Computer Program for Design and Analysis of Inverted-T Retaining Walls and Floodwalls (TWDA)	Dec 1980
Technical Report K-80-4	Documentation of Finite Element Analyses	
	Report 1: Longview Outlet Works Conduit	Dec 1980
	Report 2: Anchored Wall Monolith, Bay Springs Lock	Dec 1980
Technical Report K-80-5	Basic Pile Group Behavior	Dec 1980
Instruction Report K-81-2	User's Guide: Computer Program for Design and Analysis of Sheet Pile Walls by Classical Methods (CSHTWAL)	
	Report 1: Computational Processes	Feb 1981
	Report 2: Interactive Graphics Options	Mar 1981
Instruction Report K-81-3	Validation Report: Computer Program for Design and Analysis of Inverted-T Retaining Walls and Floodwalls (TWDA)	Feb 1981
Instruction Report K-81-4	User's Guide: Computer Program for Design and Analysis of Cast-in-Place Tunnel Linings (NEWTUN)	Mar 1981
Instruction Report K-81-6	User's Guide: Computer Program for Optimum Nonlinear Dynamic Design of Reinforced Concrete Slabs Under Blast Loading (CBARCS)	Mar 1981
Instruction Report K-81-7	User's Guide: Computer Program for Design or Investigation of Orthogonal Culverts (CORTCUL)	Mar 1981
Instruction Report K-81-9	User's Guide: Computer Program for Three-Dimensional Analysis of Building Systems (CTABS80)	Aug 1981
Technical Report K-81-2	Theoretical Basis for CTABS80: A Computer Program for Three-Dimensional Analysis of Building Systems	Sep 1981