THE UNIVERSITY OF MANITOBA

ANALYSIS OF THREE DIMENSIONAL FRAMES
WITH FLAT PLATE FLOORS

by

K. H. MA

A Thesis
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE IN CIVIL ENGINEERING

WINNIPEG, MANITOBA

October, 1976

"ANALYSIS OF THREE DIMENSIONAL FRAMES
WITH FLAT PLATE FLOORS"

by

K. H. MA

A dissertation submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

ACKNOWLEDGMENTS

The author wishes to express his sincere thanks to his adviser, Dr. G.A. Morris, for his excellent guidance and assistance rendered during the preparation of this thesis.

The guidance and criticism given by Dr. A.H. Shah were invaluable. His sincere interest in this work is gratefully acknowledged.

Thanks are also due to Dr. N. Popplewell, who read through the thesis and gave the author very valuable ideas and comments.

Finally, thanks are extended to Mrs. V. Ring for her patience and efficient typing of the entire manuscript.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

CHAPTER I

INTRODUCTION

1.1 Introduction

The analysis of the structural framing systems for large multi-storey buildings is one of the most complex of the commonly encountered problems in structural engineering. In many structural framing systems the degree of statical indeterminacy is very high and a large number of simultaneous equations must be solved. Hence, considerations of simplifying assumptions, structural modellings and analysis techniques are necessary for the reduction of the number of simultaneous equations and the minimization of computational effort. In many cases, even with consideration of all possible simplifying techniques, the number of simultaneous equations to be solved is still beyond the capacity of hand manipulation and electronic data processing is required.

Figure 1.1 shows a typical structural framing system for a multi-storey building. The framing system includes structural elements such as columns, beams, shear walls and, sometimes, bracing elements. Especially in reinforced concrete structures, shear walls are the most commonly used lateral load resisting elements. They have the advantage of providing spacious interior enclosures for elevator cores and stair wells hence serving structural and non-structural functions. Bracing elements are less common than other structural elements. They enhance the lateral load resisting capacity of the frame as a whole. For purposes of lateral load analysis, the floors are usually not considered as part of the framing system because of the complexity of determining floor stiffness characteristics.

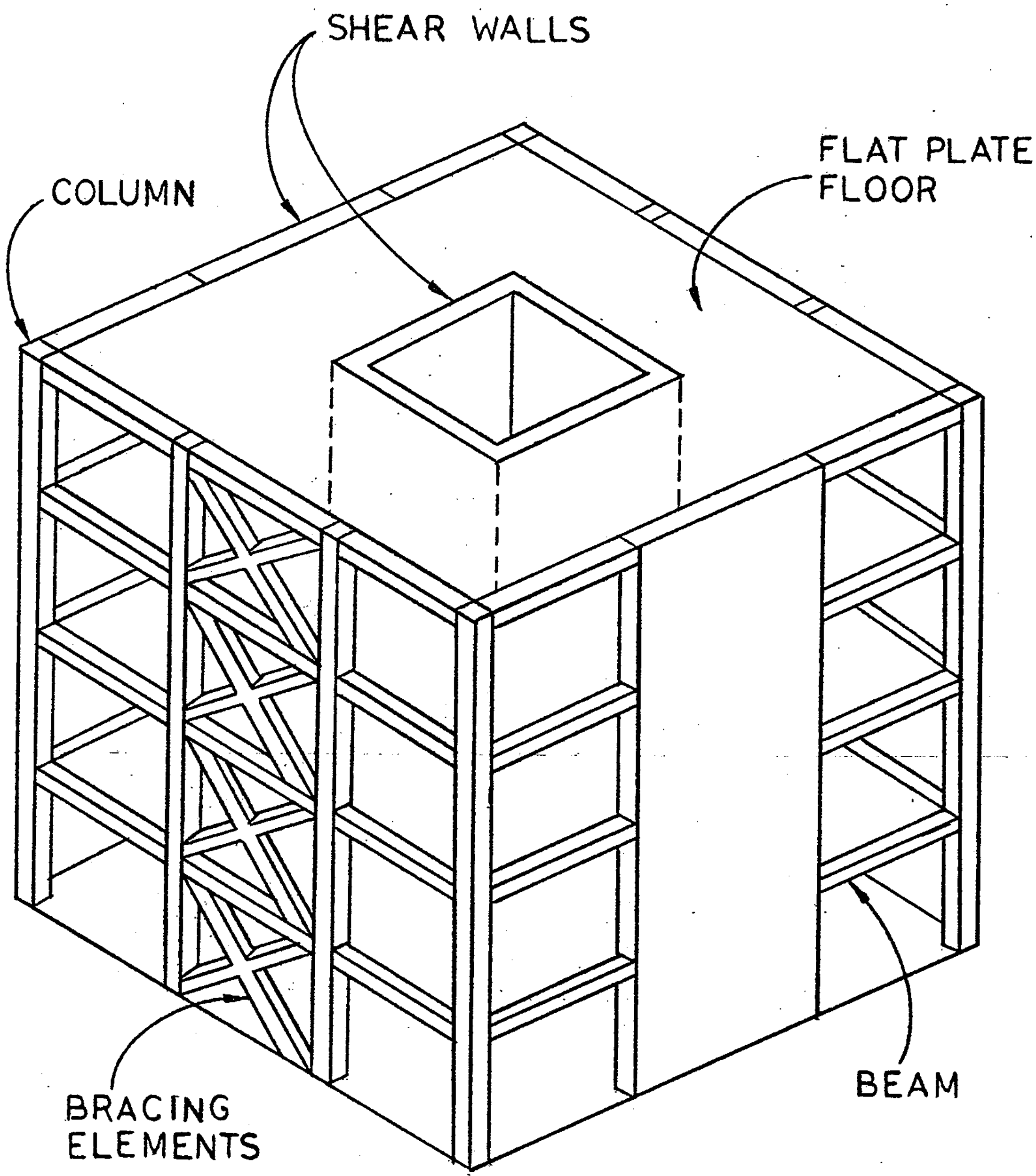


FIGURE 1·1 TYPICAL STRUCTURAL FRAMING SYSTEM FOR MULTISTOREY BUILDING

The behaviour of structural framing systems in multistorey buildings has been extensively investigated [1] and a number of computer programs are written for the analysis of shear wall-frame structures. In most of these programs, the shear walls are rather carefully modelled and the interaction between the shear walls and the remainder of the "frame" incorporated. Most such programs, however, require the user to model floors as systems of equivalent beams, sometimes with stiff in-plane bracing to represent that diaphragm action of the floor [2,3,4]. While several analytical and experimental studies of the interaction between flat plate floors, and columns and/or shear walls have been carried out [5,6], this interaction has not been incorporated into analysis programs.

1.2 Historical Background

Many investigations have been carried out on shear wall-frame structures. They can be classified into two categories; namely, two dimensional frame analysis and three dimensional frame analysis. In the two dimensional frame analysis, the overall structure is treated as a series of two dimensional frames. The frames are connected, at arbitrary intervals of height, by horizontal diaphragms (floors) which are rigid in their own planes. In the three dimensional frame analysis, the overall structure is considered as a series of three dimensional frames interconnected also by rigid horizontal diaphragms at each floor level. A single three dimensional frame may also be used to model the whole structure.

The rigid diaphragm characteristic of the floor systems has been recognized by many investigators and become universally adopted as a realistic approximation. If all joints or nodes (points of interaction between framing elements) in the structure are assumed to lie at floor levels, then all joints displacements in the horizontal plane are uniquely determined by the floor displacements--two horizontal translations and one rotation in the horizontal plane at each floor. By taking advantages of this characteristic, the analysis of framing systems can be significantly simplified.

Clough, King and Wilson [7] presented a computer program for the linear structural analysis of multistorey buildings in which they modelled the structure as two sets of parallel planar frames acting in perpendicular directions. The frames were connected at each floor level by the floor diaphragm. Each joint in each frame was assumed to have three degrees of freedom--horizontal and vertical translations and in-plane rotations. Out of plane displacements were ignored. Frame stiffness equations were assembled for the frame and static condensations were performed on them so that only stiffnesses relating to horizontal translations at each floor levels remained. Lateral frame stiffnesses were formed for each frame in turn. The lateral stiffness of the complete building was obtained as the superposition of lateral frame stiffnesses from all frames. Both static and dynamic analyses were permitted. A shortcoming of the analysis is that compatibility of vertical displacements is not assured for the duplicated columns on the boundaries between the assumed frames.

Weaver and Nelson [3] performed a three dimensional analysis of tier buildings (structures under the category of rigid bodies interconnected by structural frames which are arranged in a rectangular pattern). In this analysis, the overall structure was treated as a single three dimensional frame. The stiffness matrix formed by assembling the stiffnesses of all structural elements was condensed to a size corresponding to the three floor displacements on each floor level. Weaver's three dimensional analysis program was limited to a building with a maximum of twenty five joints per floor and twenty four storeys of height. However, such an analysis becomes mandatory when the structure is unsymmetrical.

Viswanathan [8] developed a computer program for the analysis of two dimensional shear wall-frame structures subjected to lateral loads. Shear walls were represented by rectangular finite element arrays. Hence shear walls with openings could be modelled in the analysis. Four types of rectangular finite elements with different degrees of freedom were employed for different locations in the shear wall, in order to minimize computational effort. Floor slabs were assumed to be infinitely rigid in their own planes and there was thus one transverse degree of freedom at each floor level. The program could handle large shear wall-frame structures economically but only one frame could be considered at a time. Several applications of the program were required to complete the analysis of the whole structure.

Wilson and Dovey [2] extended Clough's work [7] to enable the consideration of nonsymmetrical and non-rectangular buildings. The building was idealized as a system of independent planar frames and arbitrarily located shear walls. Shear walls could be considered either as continuous columns or as series of in-fill panel elements connected together at each floor level. Column bending, axial and shearing deformation were included. Non-prismatic beams and girders with bending and shearing deformation were permitted. Again, compatibility was not enforced with regard to displacements at joints which were common to two frames. Thus for a column common to two intersecting frames, two different axial deformations would be calculated; one from each frame. It was suggested that for design purposes, the axial forces in such columns be added directly to give reasonable results. As for joint rotations, if the frames with common members are perpendicular in plan view, then the rotations are uncoupled. For structure in which frames are not arranged orthogonally, this program is to be used with caution.

R. Shankaran Nair [9] presented a computer program for the linear structural analysis of multistorey buildings, which was similar to Wilson's [2] except that additional efforts were made to eliminate the incompatibility of joint displacements at joints common to more than one frame. The overall building was again idealized as a system of independent frames and shear walls. Frame stiffness equations were formed for each frame and condensation performed on them. The condensed frame stiffness matrix consisted of stiffnesses relating the lateral degree-of-freedom of the frame at each floor level plus the stiffnesses relating the degrees of freedom at joints which are common to more than one frame. The overall structural stiffness matrix was obtained by superimposing all condensed frame stiffness matrices and was in the order of three degrees of freedom (two horizontal translations and one rotation in the plane of the floor) per floor level plus three degrees of freedom (axial deformation and two orthogonal rotations) for each joint which is common to two frames. Hence, additional equations (from joints common to two frames) had to be solved. This analysis had the advantage that equilibrium and compatibility condition at all joints of the structure were satisfied. It is considered to be an improvement to previous techniques.

French, Kabaila and Pulmano [10] developed a computer program for the analysis of flat plate structures. A single finite element was used to model a complete panel of a flat plate floor system. In the case of a multistorey flat-plate building, the resulting structure system would consist of columns with single floor panel as connecting members. In deriving the stiffness of a single element panel, the important parameters related to the slab and column dimensions were taken into account. Each single element panel had 16 generalized coordinates. In order to conform with the column constraints, condensations were performed to yield a 12 x 12 stiffness matrix. No compatibility of either deflection or transverse slope along boundaries of panels was ensured. Results from this analysis were in agreement with the results of a finite element solution [11]. One shortcoming of this analysis is the inability to obtain the forces in the slab.

Wilson, Hollings and Dovey [4] extended their previous work [2] to permit the analysis of three dimensional frames. The building was idealized as a system of independent three dimensional frames interconnected by rigid floor diaphragms. Each joint in the frames was assumed to have six degrees of freedom (displacement along, and rotation about each coordinate axis). Three of these displacements (two translations and one rotation in the floor plane) could be transformed to the frame degrees-of-freedom at the floor level. The remaining degrees of freedom were condensed off. The entire structure stiffness matrix, which was formed by superimposing all frame stiffnesses, corresponded to three degrees of freedom per floor. The shortcoming of this analysis is that modelling of the slab is not included. Hence, users of the program are required to model the bending stiffness of the floor before analysing the structure. Again, incompatibility of displacements exists at duplicated columns on the boundaries between assumed frames. This incompatibility is significant for structures subjected to torsion as a result of horizontal loading.

1.3 Objective of Study

Many of the previous analyses are based on the assumptions that each floor diaphragm is rigid in its own plane, that the structure can be idealized as a series of two or three dimensional frames, and that compatibility at common joints of frames is not enforced. The more assumptions made to simplify the analysis, the less accurate the results. With the advancement of the technology of the high speed digital computer, an accurate method of analysis becomes desirable to describe how a structure behaves under the action of applied loadings.

Hence, the scope of this study is to develop a method of analysis of multistorey building systems which is sufficiently accurate to describe the behavior of the real structure. A computer program, coded in FORTRAN, has been written for the analysis of laterally loaded three dimensional shear wall-frame structures with flat plate floors properly modelled as series of rectangular plate panels. Efforts have been made to make the program as simple and convenient as possible to use. In addition to statical horizontal loading, gravity loading and dynamic loading are permitted. The matrix stiffness method of analysis is employed.

In this analysis, the entire structure is represented as one three dimensional frame. Thus, conditions of equilibrium and compatibility are satisfied at every joint in the structure. Each floor is modelled as a series of rectangular plate panels. Each panel, in turn, is represented by a rectangular finite element array. The bending stiffness of the floor is thus quite accurately represented, while its in-plane stiffness is assumed to be infinite. The floor bending stiffness matrix from the above finite element representation is incorporated into a three-dimensional frame analysis program [4]. Shear walls are considered as series of vertical wall panels connected together at floor levels. Column axial, bending and shearing deformations are included. Beams and girders may be nonprismatic and bending and shearing deformations are included. Finite column widths and beam depths are included in the formulation. Nonsymmetric buildings which have shear walls located arbitrarily in plan can be considered. However, the building must be laid out in a rectangular grid pattern, as shown in Figure 1.2. Up to two lateral loading cases and three vertical loading cases are permitted. If so desired, an earthquake spectrum and response analysis are also permitted (see Appendix D).

1.4 Assumptions and Limitations

The assumptions made in the analysis are:

(i) Each floor slab is assumed to be infinitely rigid in its own plane. Consequently, the axial deformations of beams are neglected.

(ii) The structure must be laid out in a rectangular grid pattern. Slab panels must be rectangular and only rectangular finite elements are considered. This assumption enables the program to generate the floor bending stiffness automatically.

(iii) All joints (points of intersection between framing elements) in the

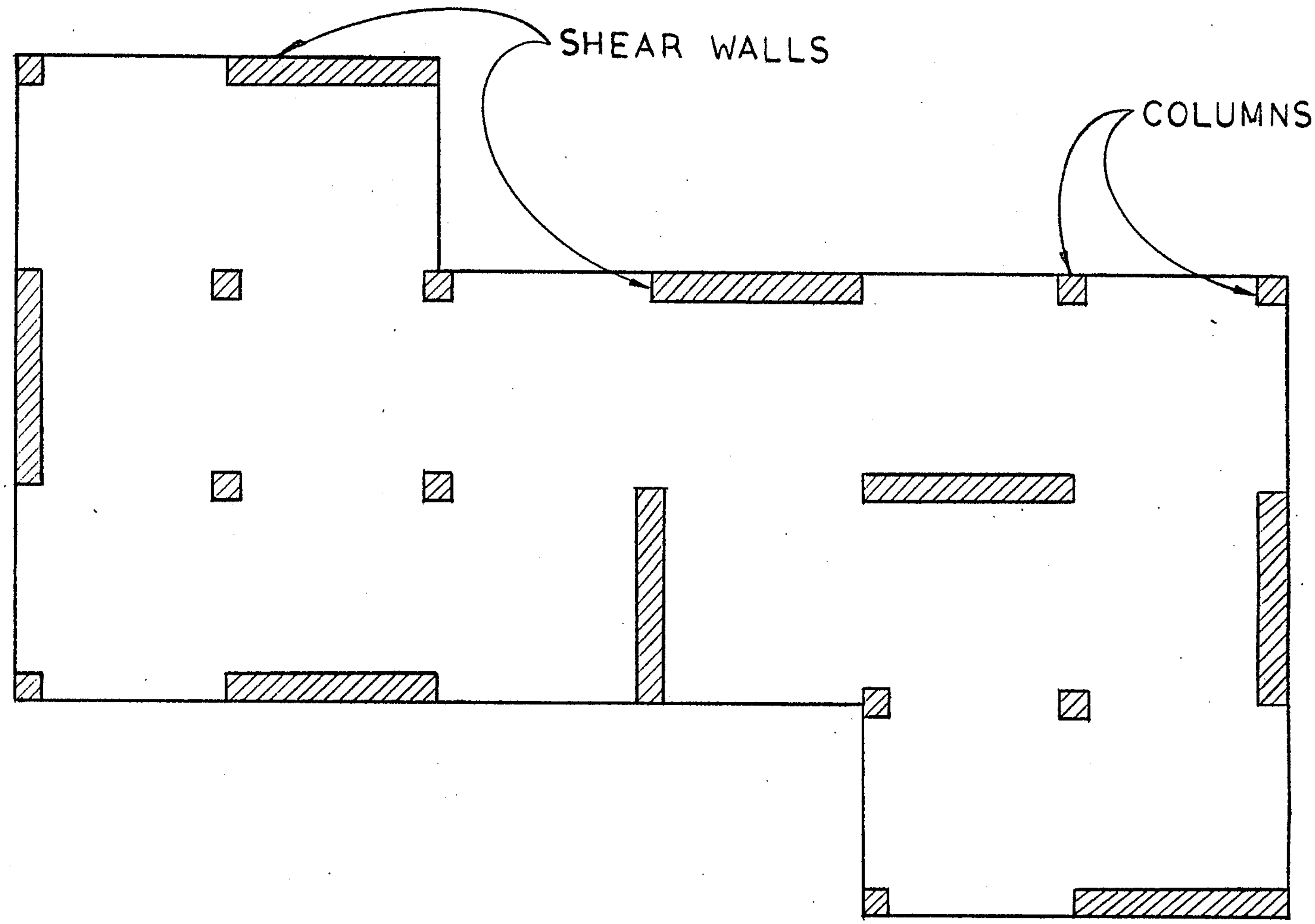


FIGURE 1·2 BUILDING LAYOUT SHOWING
RECTANGULAR PATTERN

structure must lie at floor level. Hence all joint displacements in the horizontal plane are uniquely determined by the rigid floor displacements.

(iv) Lateral loads can be applied at floor levels only. The lateral load at each floor level must be treated as a single concentrated load vector.

(v) All floors must be identical, i.e. they must have the same floor plan and the same material properties.

(vi) All slab panels in a given floor must have the same thickness, modulus of elasticity and Poisson's ratio.

(vii) Vertical loadings are assumed to act on beam members only. Consequently, vertical loading distributed over the flat plate floor, is not permitted.

(viii) Shear walls must lie on grid lines. The width of a given shear wall must be the same at all floor levels. However, the thickness may vary.

(ix) Beams (girders) may be nonprismatic, but they must be symmetrical about the vertical central line of the beam.

(x) The structure is assumed to be linearly elastic.

(xi) Finite column dimensions are not considered in formulating slab stiffness matrix (see discussion in Chapter VI).

CHAPTER II

STRUCTURAL IDEALIZATION

2.1 Introduction

An exact three dimensional structural analysis is required for only a limited number of buildings. For the majority of buildings the assumption that floors are rigid in their own planes is a realistic approximation which greatly simplifies the analysis. With this assumption made, the horizontal loads (which are assumed to act at floor levels) are transferred to the columns and shear wall elements through the rigid floor diaphragms. This results in three degrees of freedom at each floor level--two translations in the floor plane and a rotation about the vertical axis. Hence the main effort in this analysis is to assemble stiffness of various structural elements to form the structure stiffness matrix and then to reduce it to such a size that only those three degrees of freedom at each floor level are involved.

Although each floor is assumed to be rigid in its own plane, bending deformations of the floor slab and beams are considered. Special modelling of floor slab as equivalent beams is not required.

2.2 Overall Structure

The overall building considered in this analysis is composed of structural elements which together form a single three dimensional frame. All structural elements, except the floor slabs, can have different properties at different floor levels.

Shear walls are considered as series of wall panels connected together at each floor level. Thus, wall panels can be missed from or terminated at any floor level. Two dummy column lines, one at each side of the shear wall, are required to define the position of the wall panels. A sketch of a typical shear wall is shown in Figure 2.1.

2.3 Flat Plate Floor

Figure 2.2 shows a typical floor layout. The flat plate floor considered in this analysis, is modelled as a series of rectangular slab panels with columns (or dummy columns) at its corners. The floor plan must have a rectangular grid pattern. Any of the slab panels may be omitted, thus permitting irregular floor plans. A right hand coordinate system with the positive z-axis coming out of the plane of paper as shown in Figure 2.2 is employed. The origin of the coordinate system is taken as the reference point.

To enable the bending stiffness of the slab panel to be properly modelled, each slab panel in turn is subdivided into an n x n finite element mesh, as shown in Figure 2.2. The accuracy of the calculated floor stiffness increases as n increases. However, the computational effort increase drastically with an increase in n. An n value of 4 has been arbitrarily employed in this study. The effect of the number of finite element divisions on structural modelling and computational requirements will be discussed in more detail in Chapter V.

The assumption that floor slabs act as rigid diaphragms enables in-plane deformation of floor slab to be ignored and bending deformations only to be considered. Hence, each discrete finite element node has three degrees of freedom: bending about x and y axes and displacement in the z-direction.

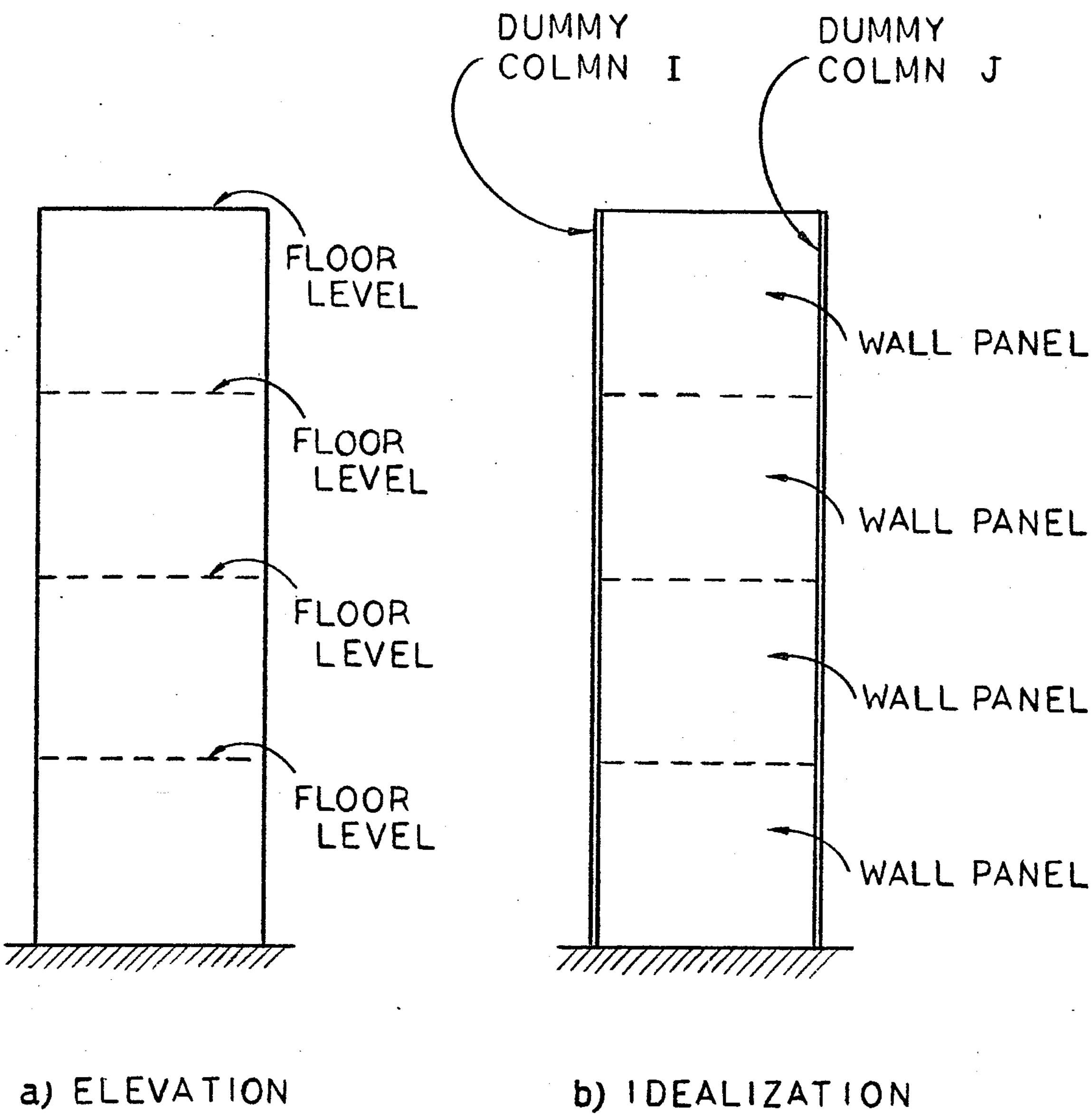


FIGURE 2·1 SHEAR WALL IDEALIZATION

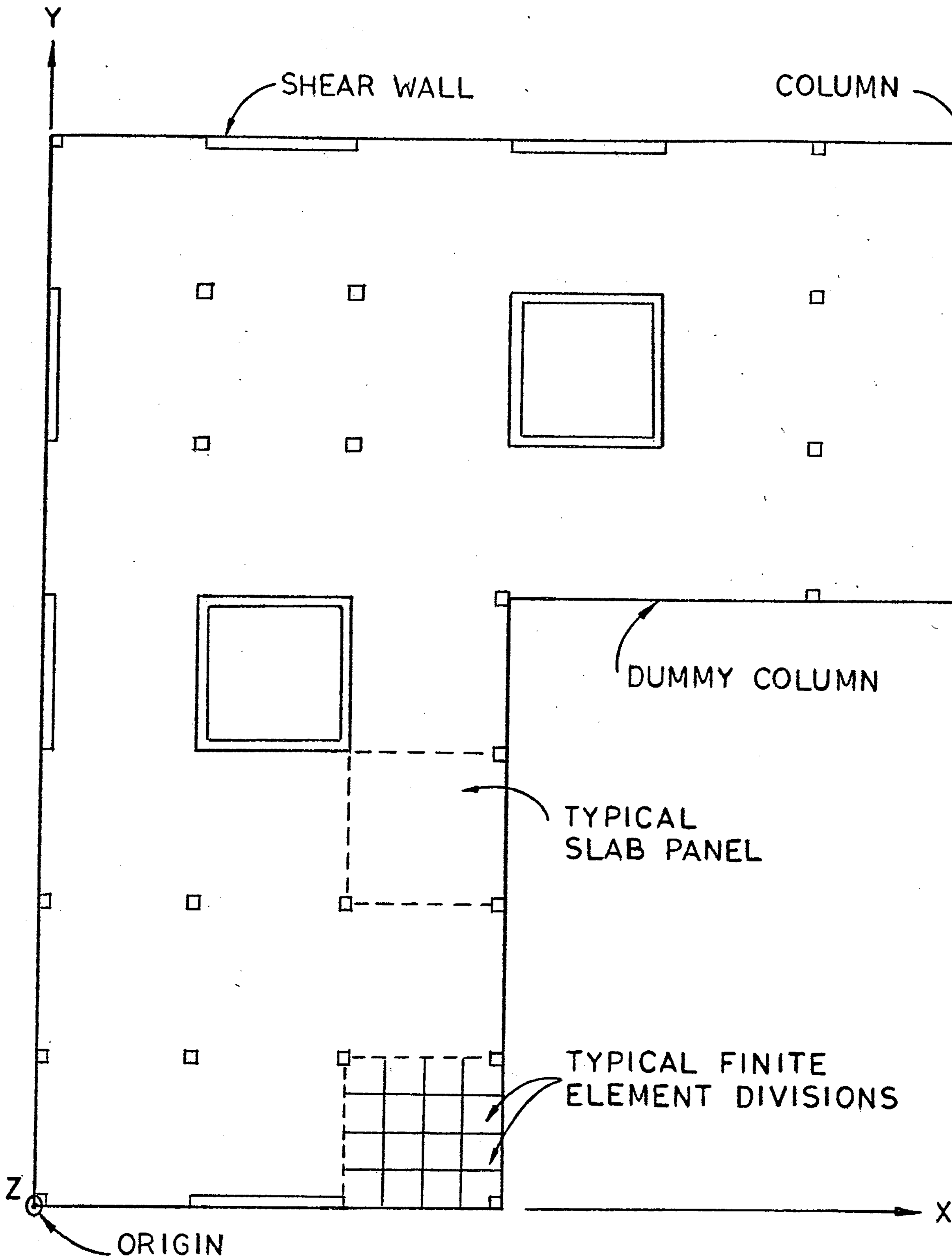


FIGURE 2·2 LAYOUT OF A TYPICAL FLOOR

Rectangular finite elements are used because the element mesh and nodal coordinates can be easily generated automatically. Floor bending stiffness equations can be generated systematically in the global co-ordinate system for the structure and no transformation is required.

2.4 Individual Element Stiffness Matrices

The development of the stiffness matrices for plate, column, beam, wall panel and diagonal bracing element is summarized in this section.

(1) Rectangular Plate Stiffness Matrix

A displacement model was used in generating the plate element stiffness matrix. The transverse displacement function [12] used was

$$w = A\,\bar{u} \tag{2.1}$$

where the positive directions of nodal displacement components

$$\bar{u} = \left\{ u_1 \; u_2 \; u_3 \; . \; . \; . \; u_{12} \right\} \tag{2.2}$$

are indicated in Figure 2.3 and

$$A^T = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \end{array} \begin{bmatrix} (1-\xi)\eta(1-\eta)^2 b \\ -\xi(1-\xi)^2(1-\eta)a \\ 1-\xi\eta-(3-2\xi)\xi^2(1-\eta)-(1-\xi)(3-2\eta)\eta^2 \\ -(1-\xi)(1-\eta)\eta^2 b \\ -\xi(1-\xi)^2\eta a \\ (1-\xi)(3-2\eta)\eta^2+\xi(1-\xi)(1-2\xi)\eta \\ -\xi(1-\eta)\eta^2 b \\ (1-\xi)\xi^2\eta a \\ (3-2\xi)\xi^2\eta-\xi\eta(1-\eta)(1-2\eta) \\ \xi\eta(1-\eta)^2 b \\ (1-\xi)\xi^2(1-\eta)a \\ (3-2\xi)\xi^2(1-\eta)+\xi\eta(1-\eta)(1-2\eta) \end{bmatrix} \tag{2.3}$$

In Equation (2.3)

$\xi = x/a$

and $\eta = y/b$

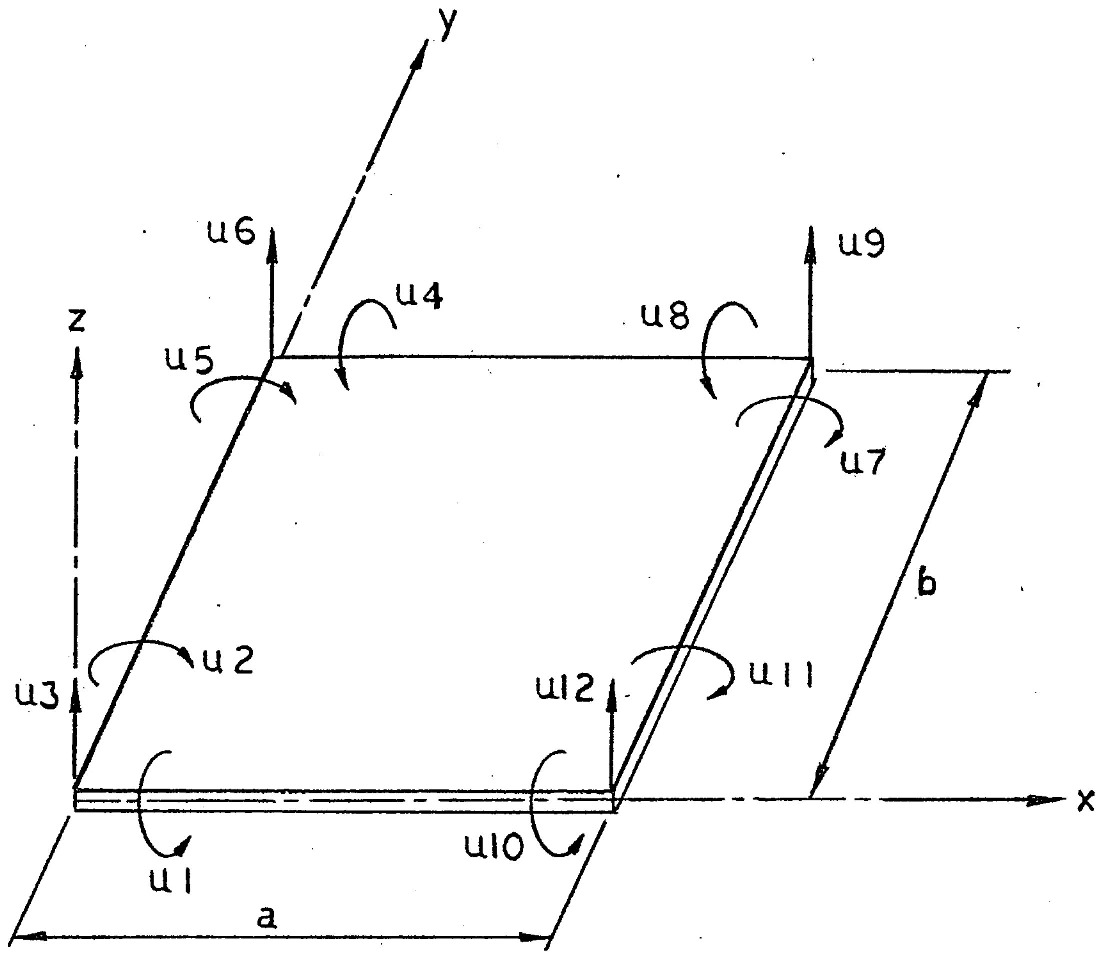


FIGURE 2·3 RECTANGULAR PLATE ELEMENT

are dimensionless natural coordinates; a and b are element dimensions. The displacement function represented by Equations (2.1) and (2.3) insures that the boundary deflections on adjacent plate elements are compatible. However, rotations of the element edges on a common boundary are not compatible, and consequently discontinuities in slope exist across the boundaries.

The strain in the plate can be expressed as

$$\bar{e} = \begin{bmatrix} e_{xx} \\ e_{yy} \\ e_{xy} \end{bmatrix} = \begin{bmatrix} -z\dfrac{\partial^2 w}{\partial x^2} \\ -z\dfrac{\partial^2 w}{\partial y^2} \\ -2z\dfrac{\partial^2 w}{\partial x \partial y} \end{bmatrix} \tag{2.4}$$

using Equations (2.3) and (2.4),

$$\bar{e} = B\,\bar{u} \tag{2.5}$$

where

$$B^T = \begin{array}{r} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \end{array} \begin{bmatrix} 0 & (1-\xi)(2-3\eta)2z/b & (1-4\eta+3\eta^2)2z/a \\ -(2-3\xi)(1-\eta)2z/a & 0 & -(1-4\xi+3\xi^2)2z/b \\ (1-2\xi)(1-\eta)6z/a^2 & (1-\xi)(1-2\eta)6z/b^2 & [1-6\xi(1-\xi)-6\eta(1-\eta)]2z/(ab) \\ 0 & (1-\xi)(1-3\eta)2z/b & -\eta(2-3\eta)2z/a \\ -(2-3\xi)\eta 2z/a & 0 & (1-4\xi+3\xi^2)2z/b \\ (1-2\xi)\eta 6z/a^2 & -(1-\xi)(1-2\eta)6z/b^2 & [-1+6\xi(1-\xi)+6\eta(1-\eta)]2z/(ab) \\ 0 & \xi(1-3\eta)2z/b & \eta(2-3\eta)2z/a \\ -(1-3\xi)\eta 2z/a & 0 & -\xi(2-3\xi)2z/b \\ -(1-2\xi)\eta 6z/a^2 & -\xi(1-2\eta)6z/b^2 & [1-6\xi(1-\xi)-6\eta(1-\eta)]2z/(ab) \\ 0 & \xi(2-3\eta)2z/b & -(1-4\eta+3\eta^2)2z/a \\ -(1-3\xi)(1-\eta)2z/a & 0 & \xi(2-3\xi)2z/b \\ -(1-2\xi)(1-\eta)6z/a & \xi(1-2\eta)6z/b^2 & [-1+6\xi(1-\xi)+6\eta(1-\eta)]2z/(ab) \end{bmatrix}$$

(2.5a)

For an isotropic plate, the plate moments can be related to the plate curvature by the elasticity matrix for the plate, as given in Equation (2.6).

$$M = D\bar{e}* \tag{2.6}$$

In Equation (2.6),

$$M = \begin{bmatrix} M_x \\ M_y \\ M_{xy} \end{bmatrix} , \quad \bar{e}* = \begin{bmatrix} -\dfrac{\partial^2 w}{\partial x^2} \\ -\dfrac{\partial^2 w}{\partial y^2} \\ -2\dfrac{\partial^2 w}{\partial x \partial y} \end{bmatrix}$$

and
$$D = \frac{Et^3}{12(1-v^2)} \begin{bmatrix} 1 & v & 0 \\ v & 1 & 0 \\ 0 & 0 & (1-v)/2 \end{bmatrix} \tag{2.7}$$

where v is the Poisson's ratio and t is the plate thickness.

The stiffness matrix for the plate is given by

$$K = \int_{Vol} B^T DB \, dx \, dy \, dz \tag{2.8}$$

. Substituting B and D into Equation (2.8), the stiffnesss matrix is obtained in the form

$$K = \left[\begin{array}{c|c} K_{I,I} & \text{Symmetric} \\ \hline K_{II,I} & K_{II,II} \end{array}\right]_{12 \times 12}$$

where the submatrices $K_{I,I}$, $K_{II,I}$ and $K_{II,II}$ are presented separately in Table 2.1. to 2.3.

$C_1 = (\frac{b}{a})^2$ $\quad C_4 = 0.2(1+4v)$ $\quad C_6 = 0.2(1-v)$

$C_2 = (\frac{a}{b})^2$ $\quad C_5 = 0.2(14-4v)$

	1	2	3	4	5	6
1	$\frac{4}{3}(C_2 + C_6)b^2$					
2	$-v\ a\ b$	$\frac{4}{3}(C_1+C_6)a^2$		SYMMETRIC		
3	$(2C_2 + C_4)b$	$-(2C_1 + C_4)a$	$4(C_1 + C_2) + C_5$			
4	$\frac{2}{3}(C_2 - C_{6/2})b^2$	0	$(2C_2 + C_6)b$	$\frac{4}{3}(C_2 + C_6)b^2$		
5	0	$\frac{2}{3}(C_1 - 2C_6)a^2$	$(-C_1 + C_4)a$	$v\ a\ b$	$\frac{4}{3}(C_1 + C_6)a^2$	
6	$-(2C_2 + C_6)b$	$(-C_1 + C_4)a$	$2(C_1-2C_2) - C_5$	$-(2C_2 + C_4)b$	$-(2C_1 + C_4)a$	$4(C_1 + C_2) + C_5$

Table 2.1 Stiffness Matrix for Rectangular Plates in Bending: Submetrix $K_{II,I}$

(All coefficients to be multiplied by $Et^3/12(1-v^2)ab$)

	1	2	3	4	5	6
7	$\frac{1}{3}(C_2 + C_6)b^2$	0	$(C_2 - C_6)b$	$\frac{2}{3}(C_2 - 2C_6)b^2$	0	$-(C_2 - C_4)b$
8	0	$\frac{1}{3}(C_1 + C_6)a^2$	$(-C_1 + C_6)a$	0	$\frac{1}{3}(2C_1 - C_6)a^2$	$(2C_1 + C_6)a$
9	$(-C_2 + C_6)b$	$(C_1 - C_6)a$	$-2(C_1 + C_2) + C_5$	$-(C_2 - C_4)b$	$(2C_1 + C_6)a$	$-2(2C_1 - C_2) - C_5$
10	$\frac{2}{3}(C_2 - 2C_6)b^2$	0	$(C_2 - C_4)b$	$\frac{1}{3}(C_2 + C_6)b^2$	0	$-(C_2 - C_6)b$
11	0	$\frac{1}{3}(2C_1 - C_6)a^2$	$-(2C_1 + C_6)a$	0	$\frac{1}{3}(C_1 + C_6)a^2$	$(-C_1 + C_6)a$
12	$(C_2 - C_4)b$	$(2C_1 + C_6)a$	$-2(2C_1 - C_2) - C_5$	$-(C_2 + C_6)b$	$(C_1 - C_6)a$	$-2(C_1 + C_2) + C_5$

Table 2.2 Stiffness Matrix for Rectangular Plates in Bending: Submatric $K_{II,I}$

(All coefficients to be multifplied by $Et^3/12(1-v^2)ab$)

7	$\frac{4}{3}(C_2 + C_6)b^2$					
8	$-\nu\ a\ b$	$\frac{4}{3}(C_1 + C_6)a^2$		SYMMETRIC		
9	$-(2C_2 + C_4)b$	$(2C_1 + C_4)a$	$4(C_1 + C_2) + C_5$			
10	$\frac{2}{3}(C_2 - C_6/2)b^2$	0	$-(2C_2 + C_6)b$	$\frac{4}{3}(C_2 + C_6)b^2$		
11	0	$\frac{2}{3}(C_1 - 2C_6)a^2$	$-(-C_1 + C_4)a$	$\nu\ a\ b$	$\frac{4}{3}(C_1 + C_6)a^2$	
12	$(2C_2 + C_6)b$	$-(-C_1 + C_4)a$	$2(C_1-2C_2) -C_5$	$(2C_2 + C_4)b$	$(2C_1 + C_4)a$	$4(C_1 + C_2) + C_5$
	7	8	9	10	11	12

Table 2.3 Stiffness Matrix for Rectangular Plates in Bending: Submatrix $K_{II,II}$

(All coefficients to be multiplied by $Et^3/12(1-\nu^2)ab$)

Once the plate nodal displacements have been computed, the nodal moments can be determined from equation (2.8a) as shown on the following page. In equation (2.8a), all coefficients not shown are zero, and

u_1, u_2, . . . u_{12} are the displacement coordinates as indicated in Figure 2.3;

M_x, M_y and M_{xy} are plate moments at each of the nodes, I, J, K and L, as indicated in Figure 2.3a;

a and b are dimensions of the plate elements;

v is the Poisson's ratio of the material;

$$C_1 = \frac{Et^3}{12ab(1-v^2)} ;$$

$$C_2 = 6\,\frac{a}{b} ;$$

$$C_3 = 6\,\frac{b}{a} ;$$

E is the elastic modulus;

and t is the thickness of the slab.

Equation (2.8a) may be written as

$$\overline{M} = S\overline{u} \tag{2.8b}$$

$$\begin{bmatrix} M_{xI} \\ M_{yI} \\ M_{xyI} \\ M_{xJ} \\ M_{yJ} \\ M_{xyJ} \\ M_{xK} \\ M_{yK} \\ M_{xyK} \\ M_{xL} \\ M_{yL} \\ M_{xyL} \end{bmatrix} = C_1 \begin{bmatrix}
-4va & 4b & \begin{matrix}-C_3\\-C_2v\end{matrix} & -2va & & C_2v & & & & & 2b & C_3 \\
-4a & 4bv & \begin{matrix}-C_2\\-C_3v\end{matrix} & -2a & & C_2 & & & & & 2bv & C_3v \\
b(1-v) & -a(1-v) & 1-v & & a(1-v) & v-1 & & & 1-v & b(v-1) & & v-1 \\
2va & & C_2v & 4va & 4b & \begin{matrix}-C_3\\-C_2v\end{matrix} & & 2b & C_3 & & & \\
2a & & C_2 & 4a & 4bv & \begin{matrix}-C_2\\-C_3v\end{matrix} & & 2bv & C_3v & & & \\
 & -a(1-v) & 1-v & b(1-v) & a(1-v) & v-1 & b(v-1) & & 1-v & & & v-1 \\
 & & & & -2b & C_3 & 4va & -4b & \begin{matrix}-C_3\\-C_2v\end{matrix} & 2va & & C_2v \\
 & & & & -2bv & C_3v & 4a & -4bv & \begin{matrix}-C_2\\-C_3v\end{matrix} & 2a & & C_2 \\
 & & 1-v & b(1-v) & & v-1 & b(v-1) & a(1-v) & 1-v & & -a(1-v) & v-1 \\
 & -2b & C_3 & & & & -2va & & C_2v & -4va & -4b & \begin{matrix}-C_3\\-C_2v\end{matrix} \\
 & -2bv & C_3v & & & & -2a & & C_2 & -4a & -4bv & \begin{matrix}-C_2\\-C_3v\end{matrix} \\
b(1-v) & & 1-v & & & v-1 & & a(1-v) & 1-v & b(v-1) & -a(1-v) & v-1
\end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ U_7 \\ u_8 \\ u_9 \\ u_{10} \\ u_{11} \\ u_{12} \end{bmatrix} \quad (2.8a)$$

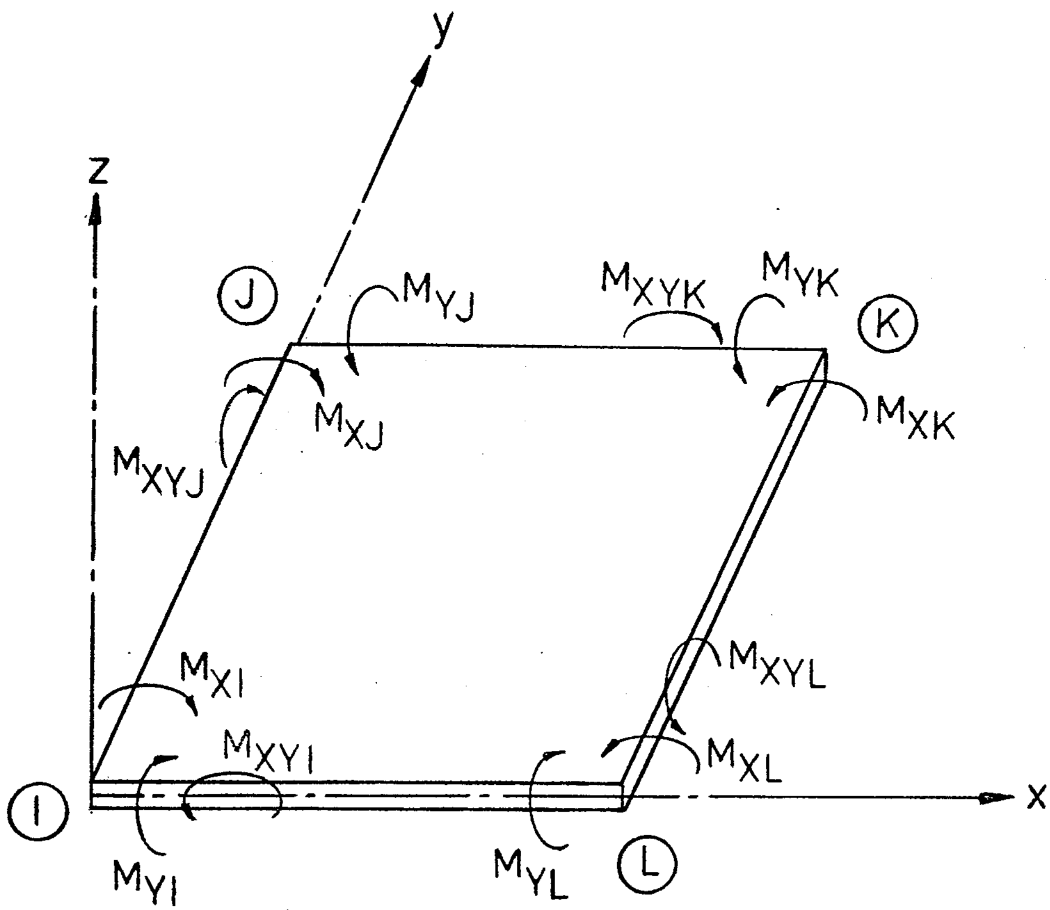


FIGURE 2·3·a MOMENTS OF RECTANGULAR PLATE ELEMENT

(ii) Column Stiffness Matrix

In term of the displacement coordinates shown in Fig. 2.4(a), the column stiffness may be defined as follow.

$$\begin{Bmatrix} M_T \\ M_{ix} \\ M_{jx} \\ P \\ M_{iy} \\ M_{jy} \end{Bmatrix} = \begin{bmatrix} S_T & & & & & \\ & S_a & S_c & & & \\ & S_c & S_a & & & \\ & & & S_A & & \\ & & & & S_b & S_d \\ & & & & S_d & S_b \end{bmatrix} \begin{Bmatrix} \theta_T \\ \theta_{ix} \\ \theta_{jx} \\ \delta \\ \theta_{iy} \\ \theta_{jy} \end{Bmatrix} \tag{2.9a}$$

or

$$\underline{S}_c = \underline{K}_c \, \underline{\theta}_c \tag{2.9b}$$

where

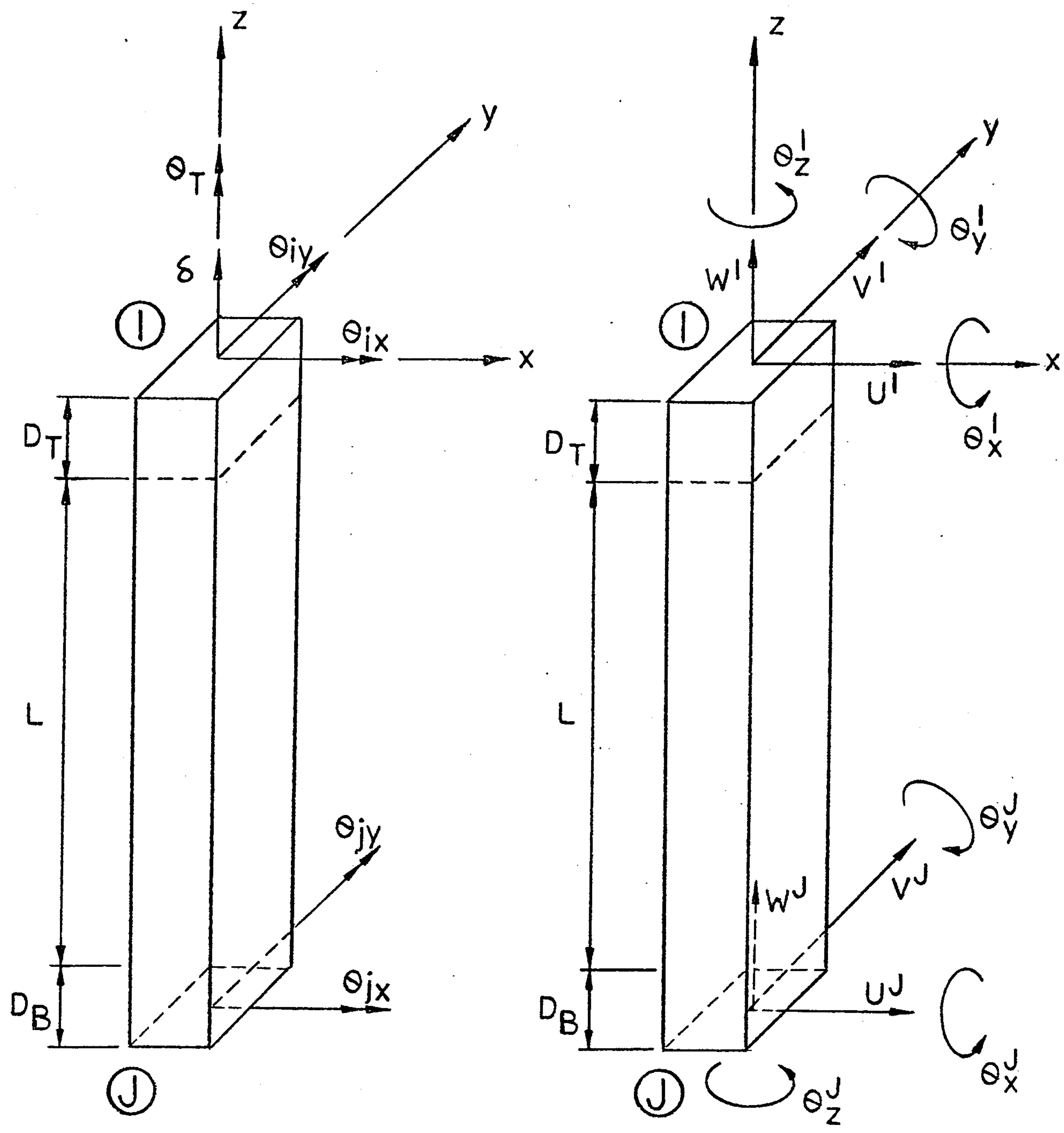
$$S_T = \frac{GI_{zz}}{L} \tag{2.9c}$$

$$S_a = \frac{2EI_{xx}}{L} \frac{(2+\beta)}{(1+2\beta)} \tag{2.9d}$$

$$S_b = \frac{2EI_{yy}}{L} \frac{(2+\beta)}{(1+2\beta)} \tag{2.9e}$$

$$S_c = \frac{2EI_{xx}}{L} \frac{(1-\beta)}{(1+2\beta)} \tag{2.9f}$$

$$S_d = \frac{2EI_{yy}}{L} \frac{(1-\beta)}{(1+2\beta)} \tag{2.9g}$$



a) COLUMN DEFORMATIONS   b) JOINT DISPLACEMENTS

FIGURE 2.4 COLUMN DEFORMATION COORDINATES

$$S_A = \frac{EA}{L} \tag{2.9h}$$

$$\beta = \frac{6EI}{L^2\overline{A}G} \tag{2.9i}$$

where

E = elastic modulus of the material,

G = shearing modulus,

I_{xx}, I_{yy}, I_{zz} = moment of inertia of the cross-section about the xx, yy and zz axis respectively,

A = cross-sectional area,

L = length of member,

β = shear flexibility factor,

$\overline{A}$ = effective shear area with respect to the axis of bending under consideration.

In line with the direct stiffness technique a transformation between the member displacements and member end displacements as shown in Figure 2.4, is now developed. The transformation for displacements at end "I" of the member is given below.

$$\begin{Bmatrix} \theta_T \\ \theta_{ix} \\ \theta_{jx} \\ \delta \\ \theta_{iy} \\ \theta_{jy} \end{Bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & \frac{1}{L} & 0 & 1+\frac{DT}{L} & 0 & 0 \\ 0 & \frac{1}{L} & 0 & \frac{DT}{L} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -\frac{1}{L} & 0 & 0 & 0 & 1+\frac{DT}{L} & 0 \\ -\frac{1}{L} & 0 & 0 & 0 & \frac{DT}{L} & 0 \end{bmatrix} \begin{Bmatrix} U^I \\ V^I \\ \theta_z^{\ I} \\ \theta_x^{\ I} \\ \theta_y^{\ I} \\ W^I \end{Bmatrix} \tag{2.10}$$

Equations 2.9(a) and 2.10 may be written symbolically as follows:

$$\underline{S}_c = \underline{K}_c \, \underline{\theta}_c \tag{2.11}$$

and

$$\underline{\theta}_c = \underline{a}_c \, \underline{r}_c \tag{2.12}$$

where the subscript c indicates column, $\underline{a}_c$ is the transformation matrix and $\underline{r}_c$ denotes the member end displacements. There is a further transformation from member end displacements to frame displacements at the origin, as shown in Figure 2.5. The assumption of rigid in-plane floor diaphragms is utilized to express those displacements and rotation in the plane of the floor slab in terms of a master node for that level, located at the origin of the coordinate axes of the frame.

The required transformation is

$$\begin{Bmatrix} U^I \\ V^I \\ \theta_z^{\,I} \\ \theta_x^{\,I} \\ \theta_y^{\,I} \\ W^I \end{Bmatrix} = \begin{bmatrix} 1 & 0 & -Y & 0 & 0 & 0 \\ 0 & 1 & X & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} R_x^{\,f} \\ R_y^{\,f} \\ \theta_z^{\,f} \\ \theta_x^{\,f} \\ \theta_y^{\,f} \\ R_z^{\,f} \end{Bmatrix} \tag{2.13a}$$

or

$$\underline{r}_x = \underline{b}_c \, \underline{r}_f \tag{2.13b}$$

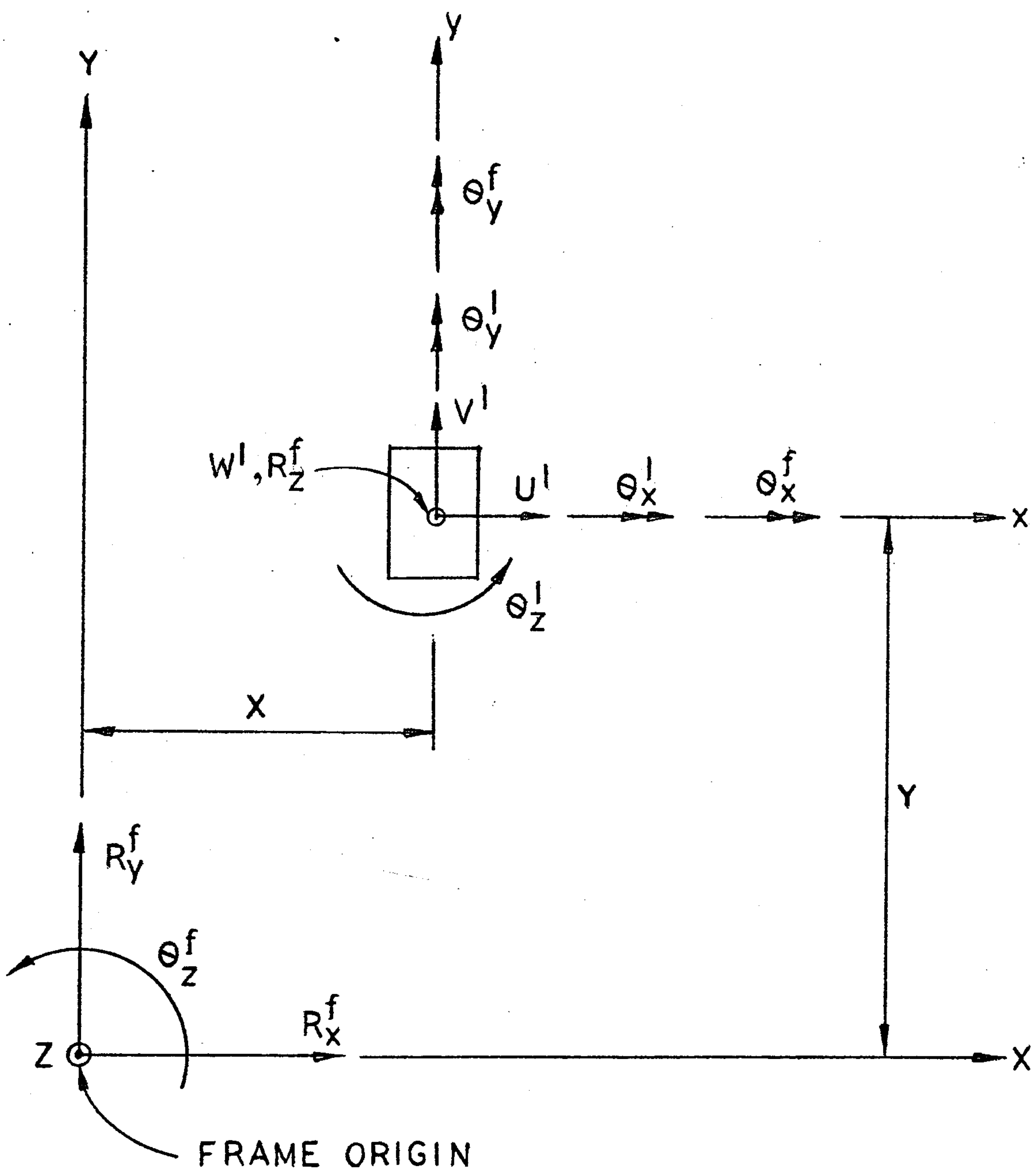


FIGURE 2·5 COLUMN MEMBER END AND FRAME DISPLACEMENTS

Equations 2.9a, 2.10, 2.13a may be written as

$$\underline{S}_c = \underline{K}_c \, \underline{\theta}_c$$

$$\underline{\theta}_c = \underline{a}_c \, \underline{r}_c$$

$$\underline{r}_c = \underline{b}_c \, \underline{r}_f$$

or

$$\underline{S}_c = \underline{K}_c \, \underline{a}_c \, \underline{b}_c \, \underline{r}_f$$

The stiffness matrix for an individual column in terms of the frame displacement is given by

$$K_c = \underline{b}_c^T \, \underline{a}_c^T \, \underline{K}_c \, \underline{a}_c \, \underline{b}_c$$

(iii) Beam Stiffness Matrix

The beam stiffness matrix is derived in a similar fashion to that for the column, except that bending about the vertical axis and axial deformations are neglected, as illustrated in Figure 2.6(a). The beam stiffness is defined as follow.

$$\begin{Bmatrix} M_T \\ M_I \\ M_J \end{Bmatrix} = \begin{bmatrix} S_T & 0 & 0 \\ 0 & S_a & S_b \\ 0 & S_b & S_a \end{bmatrix} \begin{Bmatrix} \theta_T \\ \theta_i \\ \theta_j \end{Bmatrix} \qquad (2.14)$$

or

$$\underline{S}_b = \underline{K}_b \, \underline{\theta}_b \qquad (2.14b)$$

With reference to Figure 2.6, the transformation from the member displacements to the member end displacements is as follows.

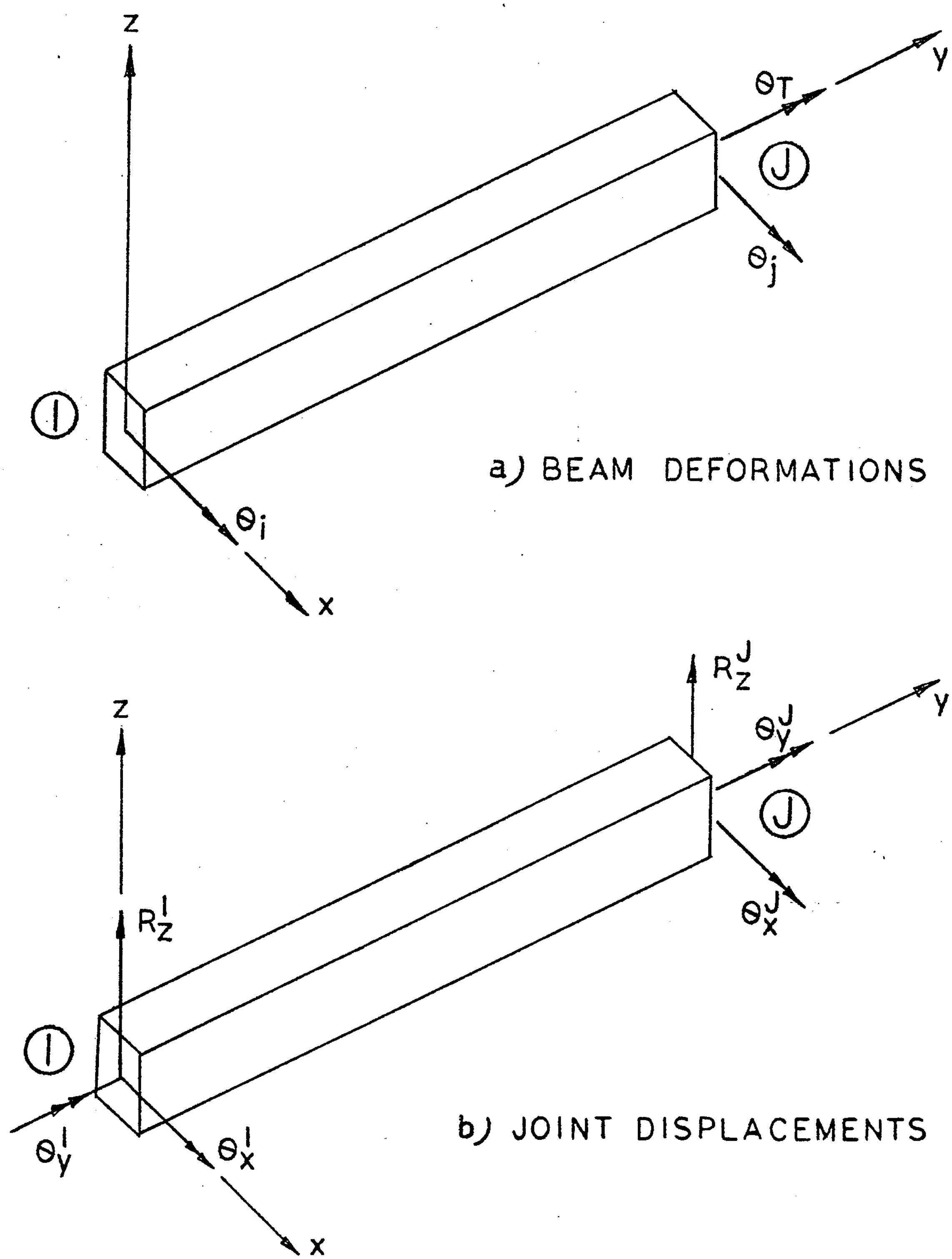


FIGURE 2·6 BEAM DEFORMATION COORDINATES

$$\begin{Bmatrix} \theta_T \\ \theta_i \\ \theta_j \end{Bmatrix} = \begin{bmatrix} 0 & -1 & 0 & 0 & 1 & 0 \\ 1+\frac{b}{L} & 0 & \frac{1}{L} & \frac{a}{L} & 0 & -\frac{1}{L} \\ \frac{b}{L} & 0 & \frac{1}{L} & 1+\frac{a}{L} & 0 & -\frac{1}{L} \end{bmatrix} \begin{Bmatrix} \theta_x^{\,I} \\ \theta_y^{\,I} \\ R_z^{\,I} \\ \theta_x^{\,J} \\ \theta_y^{\,J} \\ R_z^{\,J} \end{Bmatrix} \tag{2.15}$$

or

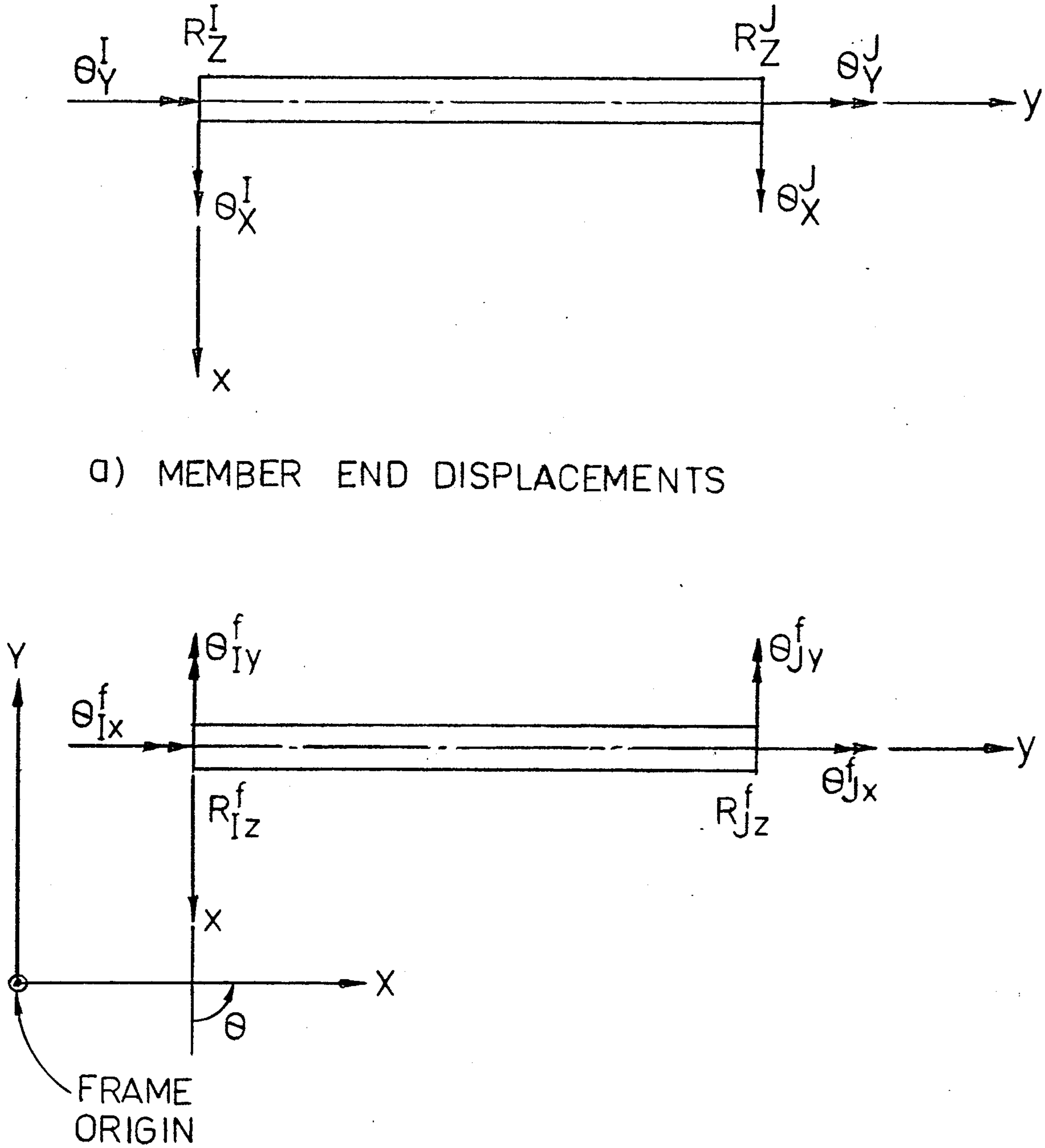
$$\underline{\theta}_b = \underline{a}_b \, \underline{r}_b$$

The further rotation transformation to the global coordinate system for the overall frame (illustrated in Figure 2.7) is:

$$\begin{Bmatrix} \theta_x^{\,I} \\ \theta_y^{\,I} \\ R_z^{\,I} \\ \theta_x^{\,J} \\ \theta_y^{\,J} \\ R_z^{\,J} \end{Bmatrix} = \begin{bmatrix} \sin\theta & -\cos\theta & 0 & 0 & 0 & 0 \\ \cos\theta & \sin\theta & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sin\theta & -\cos\theta & 0 \\ 0 & 0 & 0 & \cos\theta & \sin\theta & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} \theta^f_{Ix} \\ \theta^f_{Iy} \\ R^f_{Iz} \\ \theta^f_{Jx} \\ \theta^f_{jy} \\ R^f_{Jz} \end{Bmatrix} \tag{2.16}$$

or

$$\underline{r}_b = \underline{b}_b \, \underline{r}_f$$



a) MEMBER END DISPLACEMENTS

b) FRAME DISPLACEMENTS

FIGURE 2-7 MEMBER END AND FRAME DISPLACEMENTS — BEAM

Equation 2.14, 2.15, 2.16 may be written as

$$\underline{S}_b = \underline{K}_b \, \underline{\theta}_b$$

$$\underline{\theta}_b = \underline{a}_b \, \underline{r}_b$$

$$r_b = \underline{b}_b \, \underline{r}_f$$

The beam stiffness matrix is then given by

$$K_b = \underline{b}_b^T \, \underline{a}_b^T \, \underline{K}_b \, \underline{a}_b \, \underline{b}_b$$

(iv) Wall Panel Stiffness Matrix

Two types of wall panel can be considered.

(a) Flexural Panel

The standard in-plane column stiffness matrix, including bending and shearing deformation, is used. The flexural wall stiffness matrix is as follow: (Figure 2.8a)

$$\begin{Bmatrix} M_T \\ M_B \\ P \end{Bmatrix} = \begin{bmatrix} S_a & S_c & 0 \\ S_c & S_a & 0 \\ 0 & 0 & S_A \end{bmatrix} \begin{Bmatrix} \theta_T \\ \theta_B \\ \delta \end{Bmatrix} \qquad (2.17)$$

or

$$\underline{S}_p = \underline{K}_p \, \underline{\theta}_p \qquad (2.17b)$$

With reference to Figure 2.8, the deformation transformation matrix is given by

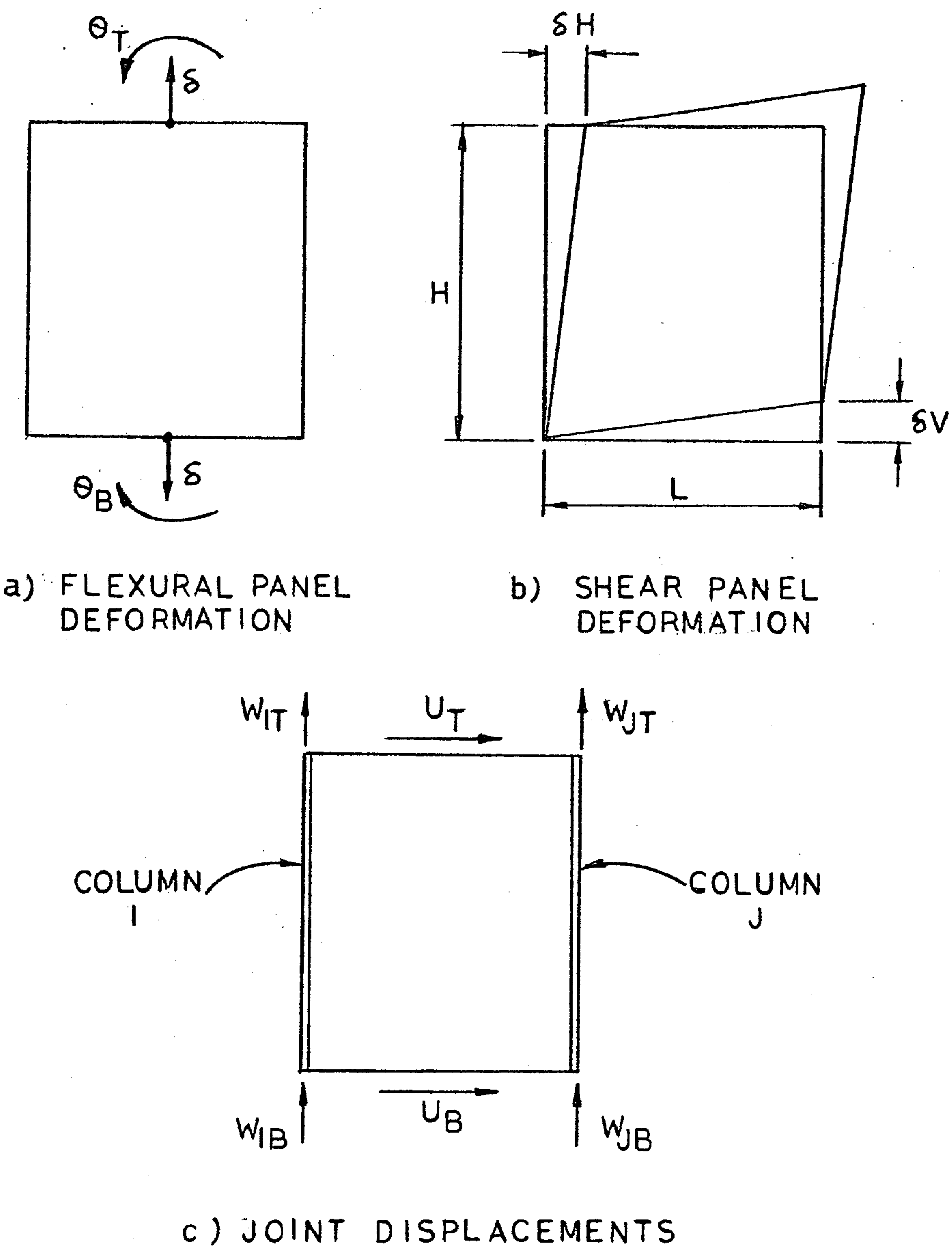


FIGURE 2·8 WALL PANEL DEFORMATION COORDINATES

$$\begin{Bmatrix} \theta_T \\ \theta_\beta \\ \delta \end{Bmatrix} = \begin{bmatrix} \frac{1}{H} & -\frac{1}{L} & \frac{1}{L} & -\frac{1}{H} & 0 & 0 \\ \frac{1}{H} & 0 & 0 & -\frac{1}{H} & -\frac{1}{L} & \frac{1}{L} \\ 0 & \frac{1}{2} & \frac{1}{2} & 0 & -\frac{1}{2} & -\frac{1}{2} \end{bmatrix} \begin{Bmatrix} U_T \\ W_{IT} \\ W_{JT} \\ U_B \\ W_{IB} \\ W_{JB} \end{Bmatrix} \tag{2.18}$$

or

$$\underline{\theta}_p = \underline{a}_p \, \underline{r}_p$$

(b) Pure Shear Panel

The deformation of the pure shear panel is as shown in Figure 2.8(b). The panel is assumed to carry only pure shear. Thus, the simple shear constitutive relationship;

$$\tau = G\gamma \tag{2.19}$$

is employed,

where τ denotes the shear stress and γ is the shear strain.

From Figure 2.8(b), it is noted that shear strain is induced by relative horizontal and vertical displacement of the sides of the panel. That is

$$\gamma = \frac{\delta H}{H} + \frac{\delta v}{L} \tag{2.20}$$

The average values of vertical joint displacement on each side of the panel are used to calculate the relative vertical displacement. Thus the following deformation - displacement transformation matrix is obtained:

$$\gamma = \begin{bmatrix} \frac{1}{H} & -\frac{1}{2L} & \frac{1}{2L} & -\frac{1}{H} & \frac{1}{2L} & \frac{1}{2L} \end{bmatrix} \begin{Bmatrix} U_T \\ W_{IT} \\ W_{JT} \\ U_B \\ W_{IB} \\ W_{JB} \end{Bmatrix} \tag{2.21}$$

or

$$\gamma = \underline{a}_p \, \underline{r}_p$$

The shear panel stiffness matrix is obtained as

$$\underline{K}_p = \int_{Vol} \underline{a}_p^T \, G \, \underline{a}_p \, dv$$

or

$$\underline{K}_p = H \, A \, G \, \underline{a}_p^T \, \underline{a}_p \tag{2.22}$$

where A is the cross-sectional area of the wall panel.

The transformation to frame displacements is shown in Figure 2.9 and is the same for both flexural and shear panel.

$$\begin{Bmatrix} U_T \\ W_{IT} \\ W_{JT} \end{Bmatrix} = \begin{bmatrix} \cos\beta & \sin\beta & -D & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} R_x^f \\ R_y^f \\ R_\theta^f \\ R_{Iz}^f \\ R_{Jz}^f \end{Bmatrix} \tag{2.23}$$

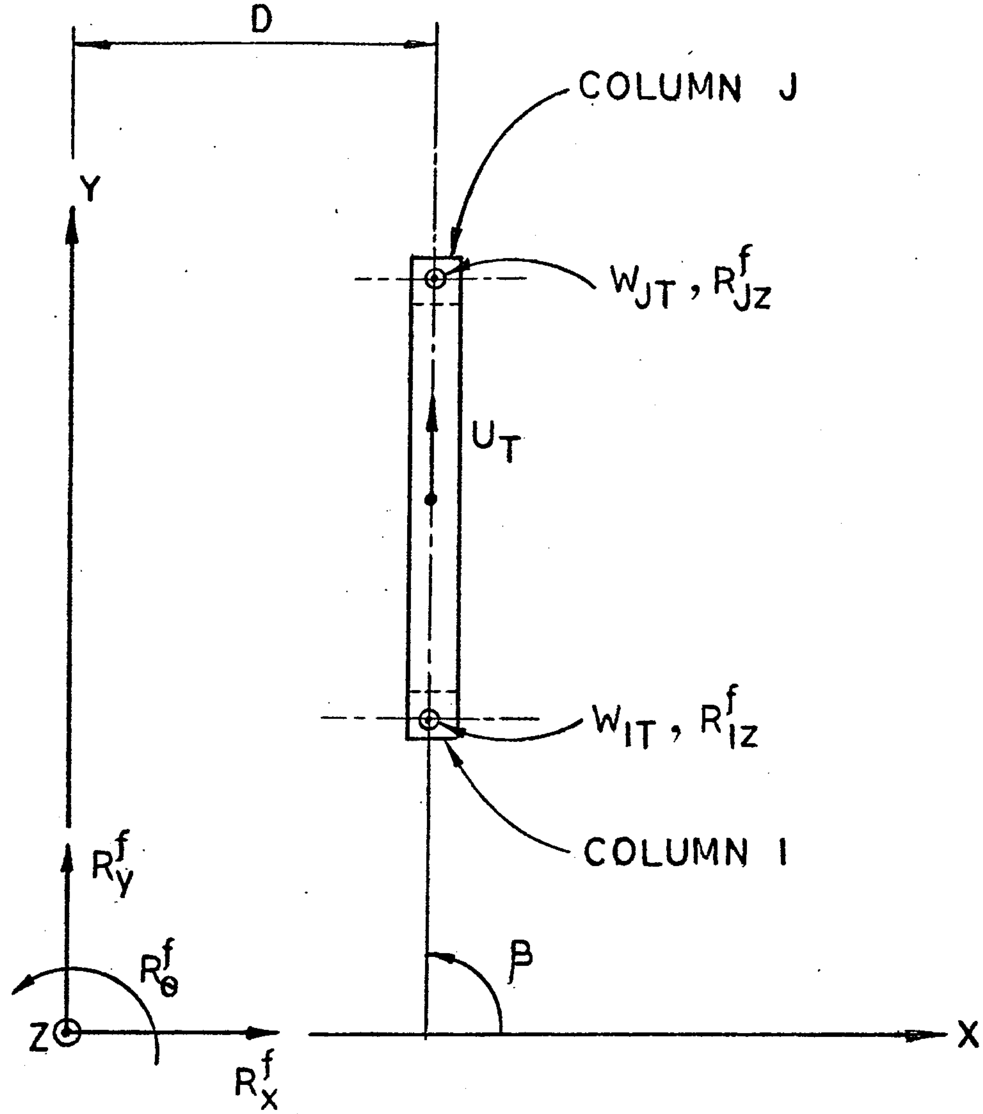


FIGURE 2·9 PANEL AND FRAME DISPLACEMENTS or

$$\underline{r}_p = \underline{b}_p \, \underline{r}_f$$

From Equations 2.17, 2.18, 2.22 and 2.23 the panel stiffness matrix with respect to the frame displacements, is

$$K_p = \underline{b}_p^T \, \underline{a}_p^T \, \underline{K}_p \, \underline{a}_p \, \underline{b}_p$$

(v) <u>Diagonal Bracing Stiffness Matrix</u>

The diagonal bracing member axial stiffness is defined as

$$S = \frac{EA}{L} \delta \tag{2.24}$$

where

A = cross-sectional area,

L = length of elements,

E = elastic modulus,

or

$$\underline{S}_D = \underline{K}_D \, \underline{\theta}_D \tag{2.24b}$$

The transition from member to frame displacement coordinates involves two rotation transformations.

(a) Transformation to horizontal and vertical displacements as shown in Figure 2.10

$$\delta = \begin{bmatrix} \sin\alpha & \cos\alpha & -\sin\alpha & -\cos\alpha \end{bmatrix} \begin{Bmatrix} V^T \\ U^T \\ V^B \\ U^B \end{Bmatrix} \tag{2.25}$$

or

$$\underline{\theta}_D = \underline{a}_D \, \underline{r}_D$$

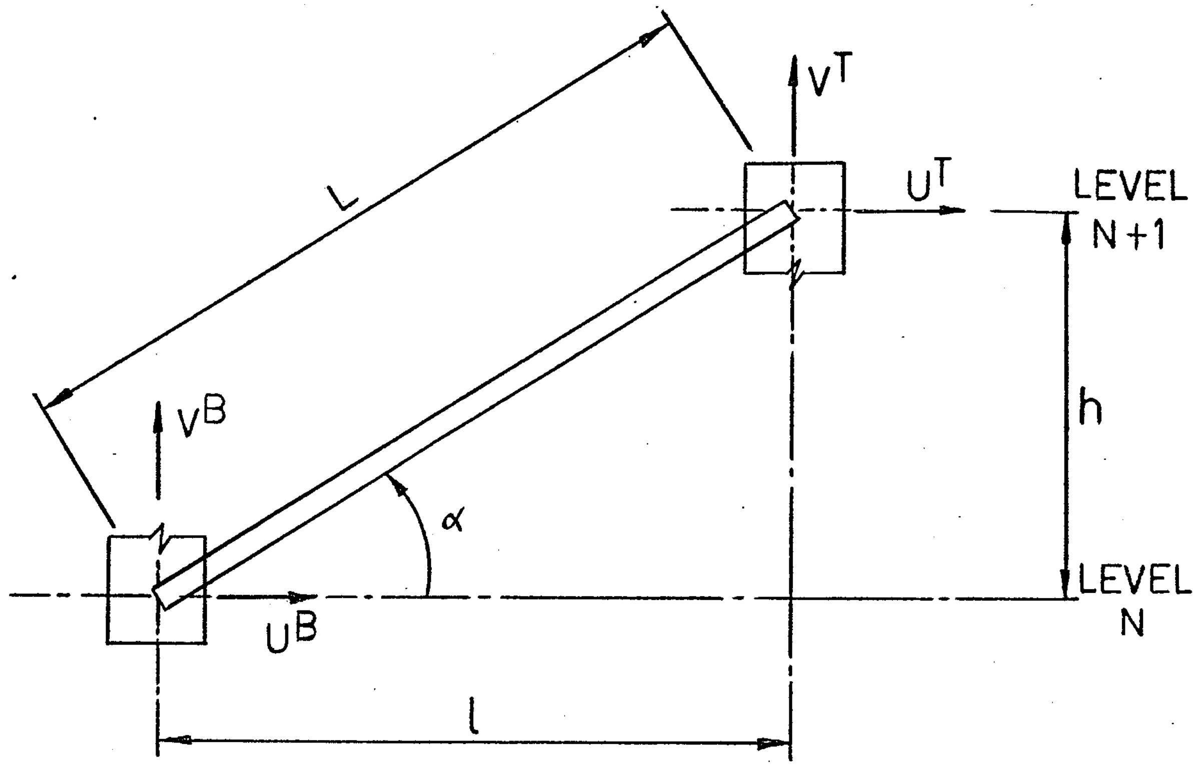


FIGURE 2·10 DIAGONAL DEFORMATIONS AND JOINT DISPLACEMENTS

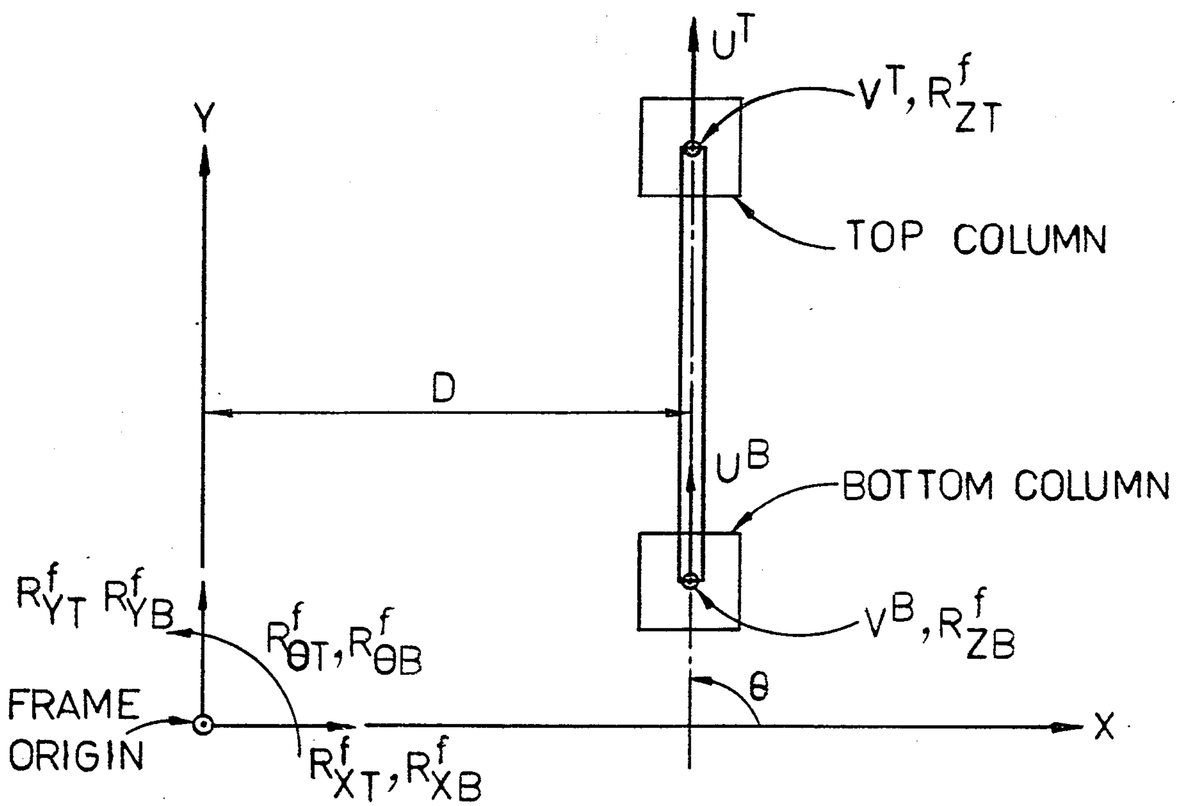


FIGURE 2·11 DIAGONAL JOINT AND FRAME DISPLACEMENTS

(b) Transformation at each level to frame displacement as shown in Figure 2.11.

$$\begin{Bmatrix} V^T \\ U^T \end{Bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ \cos\theta & \sin\theta & D & 0 \end{bmatrix} \begin{Bmatrix} R^f_{xT} \\ R^f_{yT} \\ R^f_{\theta T} \\ R^f_{zT} \end{Bmatrix} \qquad (2.26)$$

or

$$\underline{r}_D = \underline{b}_D \, \underline{r}_f$$

From Equations 2.24, 2.25, 2.26 the diagonal bracing stiffness matrix is given by

$$K_D = \underline{b}_D^T \, \underline{a}_D^T \, \underline{K}_D \, \underline{a}_D \, \underline{b}_D$$

CHAPTER III

BENDING STIFFNESS MATRIX OF A TYPICAL FLAT PLATE FLOOR

3.1 Introduction

In this chapter, a method is presented for automatically generating the bending stiffness matrix of a typical flat plate floor. Although the term "flat plate floor" is employed, beams can be incorporated along any column line. The bending stiffness of the floor system then includes contributions from both the "flat plate floor" and the floor beams (Figure 3.1). However, since the "flat plate floor" and the floor beams are considered as separate elements, no "tee beam" action is accounted for and the modelled structure is more flexible than the real one. The bending stiffness matrix for the "flat plate floor" is discussed in this chapter while that for the floor beams will be discussed in Chapter IV.

3.2 Aitken's Method of Condensation

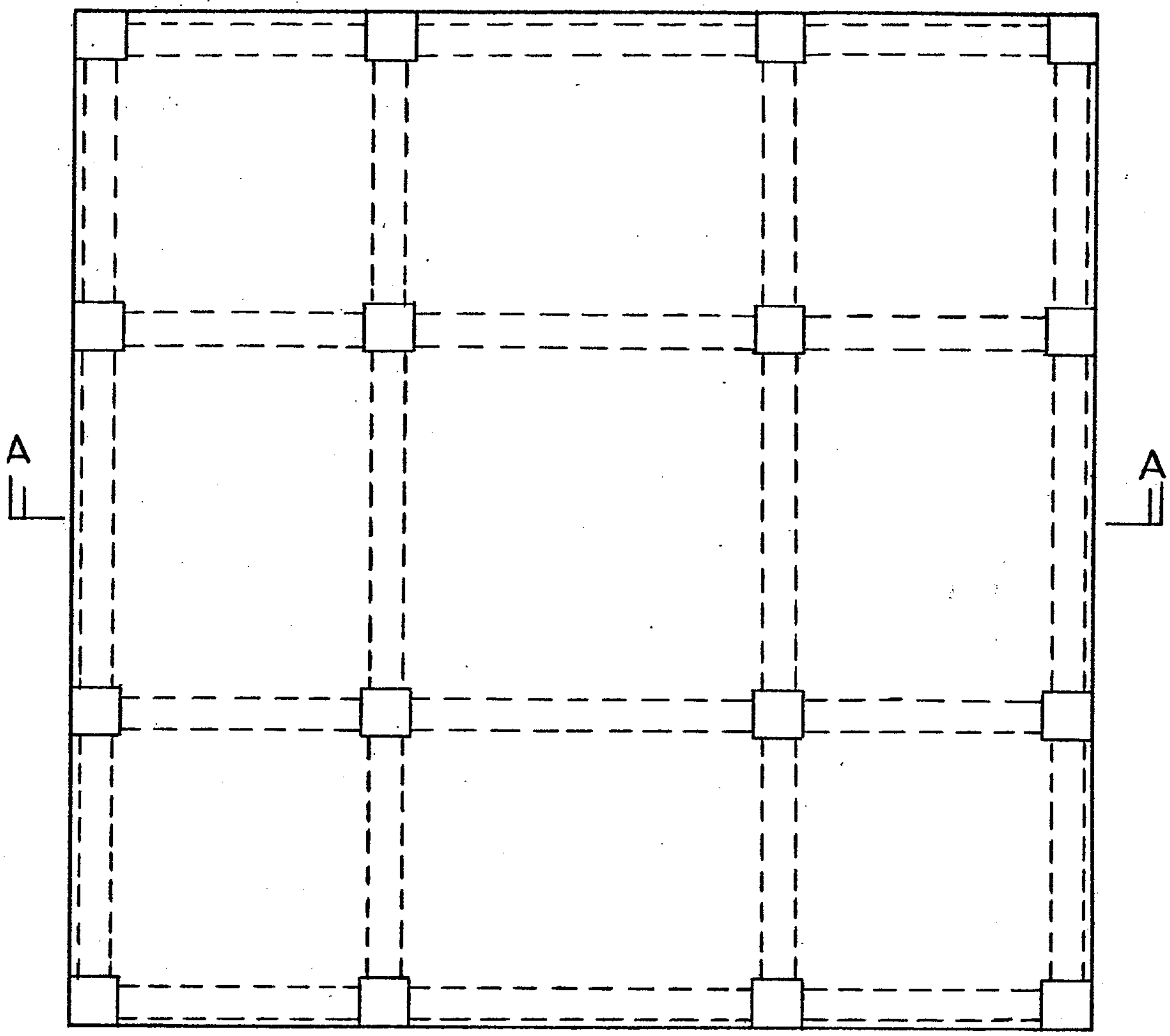
Figure 3.2 shows a model of a four-panel flat plate floor. All quantities associated with the corner nodes for all panels are designated E while those associated with all other finite element nodes are designated I. The force-displacement relationship for the floor slab can be expressed in partitioned form, as follows

$$\begin{Bmatrix} P_I \\ \hline P_E \end{Bmatrix} = \left[\begin{array}{c|c} K_{II} & K_{IE} \\ \hline K_{EI} & K_{EE} \end{array}\right] \begin{Bmatrix} D_I \\ \hline D_E \end{Bmatrix} \tag{3.1}$$

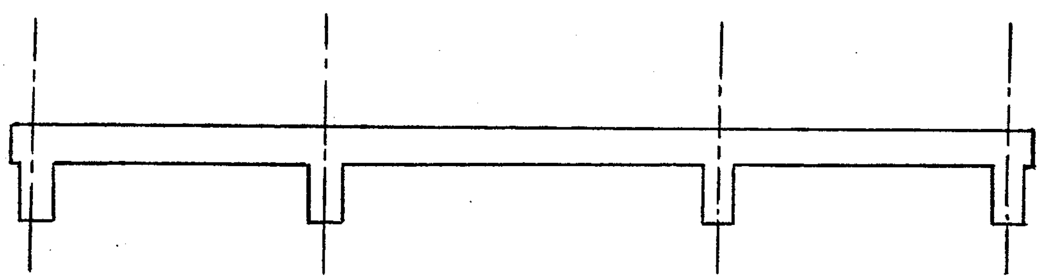
where P is the vector of forces applied to all nodes,

D is the nodal displacement vector,

and K is the stiffness matrix for the floor.



a) PLAN VIEW



b) SECTION A—A

FIGURE 3·1 TYPICAL FLOOR SYSTEM

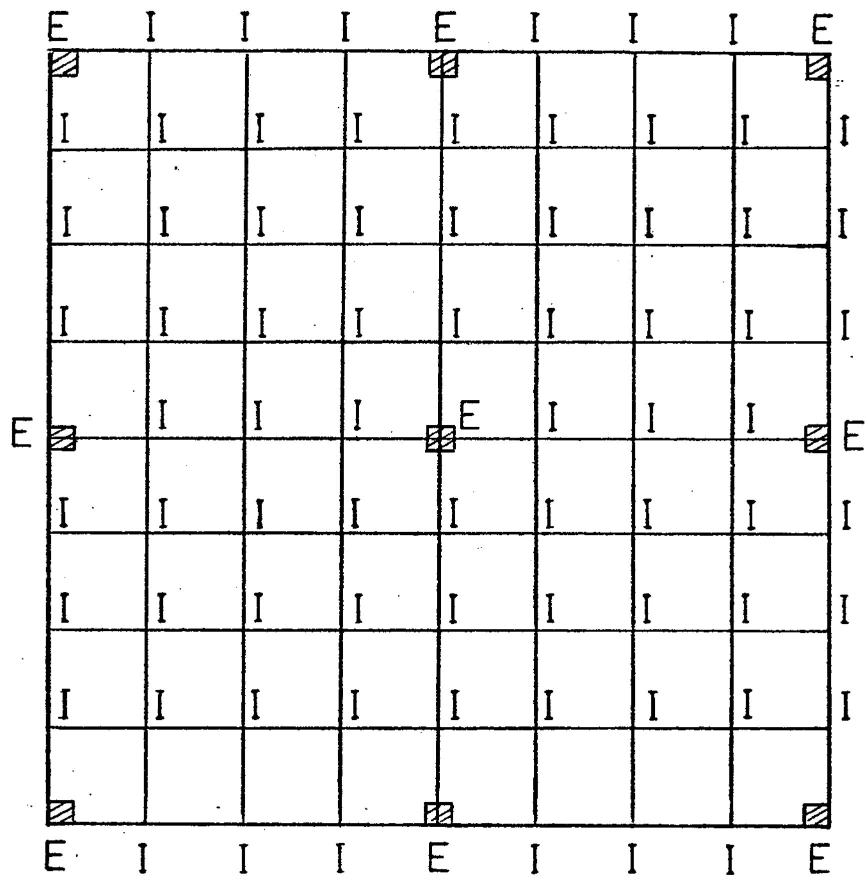


FIGURE 3·2 FOUR-PANEL FLAT PLATE MODEL

Normally, the matrix K is of very high order. If the assumption is made that all forces acting on nodes I are zero, and that the flat plate is connected to the framing system at the panel corners only, Equation (3.1) can be condensed to a smaller size, relating forces and displacements at nodes E only. This can be done as follows:

Equation (3.1) can be rewritten as

$$P_I = K_{II} D_I + K_{IE} D_E \tag{3.2}$$

$$P_E = K_{EI} D_I + K_{EE} D_E \tag{3.3}$$

Since P_I is {0}, equation (3.2) becomes

$$D_I = K_{II}^{-1} K_{IE} D_E \tag{3.4}$$

Substituting equation (3.4) into (3.3),

$$P_E = (K_{EE} - K_{EI} K_{II}^{-1} K_{IE}) D_E \tag{3.5}$$

In equations (3.5), P_E and D_E are respectively, the force and displacement vectors at nodes E. The quantity in the bracket is the condensed stiffness matrix, and its order corresponds to the number of nodes E.

The condensed stiffness matrix can be obtained by performing Gaussian Elimination on the original matrix K. The Gaussian Elimination can be expressed mathematically as

$$\left[\begin{array}{c|c} M & 0 \\ \hline L & I \end{array}\right] \left[\begin{array}{c|c} K_{II} & K_{IE} \\ \hline K_{EI} & K_{EE} \end{array}\right] = \left[\begin{array}{c|c} M\,K_{II} & M\,K_{IE} \\ \hline LK_{II} + K_{EI} & LK_{IE} + K_{EE} \end{array}\right] \tag{3.6}$$

where

0 = null matrix,

I = unit matrix,

M = a lower triangular matrix which performs a Gaussian Elimination on K_{II} and converts it to an upper triangular matrix $M K_{II}$,

L = a matrix which performs a Gaussian Elimination on K_{EI} and converts it to a null matrix.

Thus

$$LK_{II} + K_{EI} = 0,$$

and

$$L = -K_{EI} K_{II}^{-1}$$

Substituting the value of L into the lower right hand term of the product matrix in Equation 3.6, the term becomes

$$LK_{IE} + K_{EE} = (-K_{EI} K_{II}^{-1}) K_{IE} + K_{EE}$$

$$= K_{EE} - K_{EI} K_{II}^{-1} K_{IE}$$

Hence, during the Gaussian Elimination, the submatrix K_{EE} is automatically converted to the condensed stiffness matrix, K_c. The above process is known as Aitken's Method of Condensation [16].

Since Aitken's Method will be referred to later, it is convenient to express the process graphically as shown in Figure 3.3. In the figure, the arrow indicates that Gaussian Elimination, when performed on the matrix on the left, produces the matrix on the right, with zero coefficients as shown.

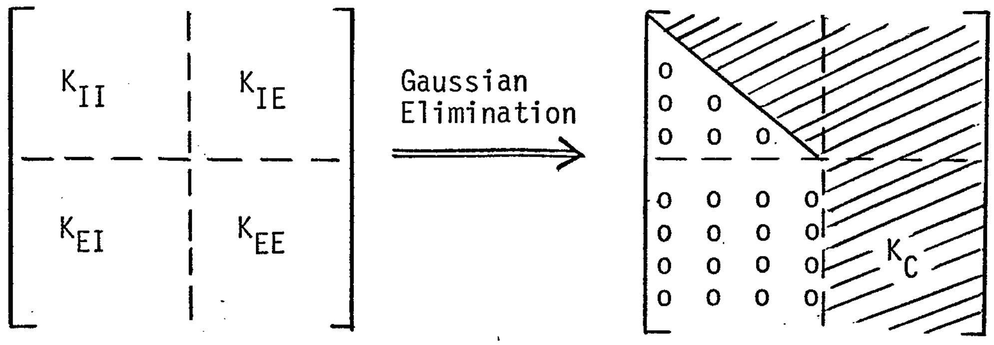


FIGURE 3.3 Graphical Representation of Aitken's Method of Condensation

3.3 Procedure for Assembling Bending Stiffness Matrix for Flat Plate Floor

As mentioned in the previous section, the following bending stiffness equation can be obtained for a flat plate floor

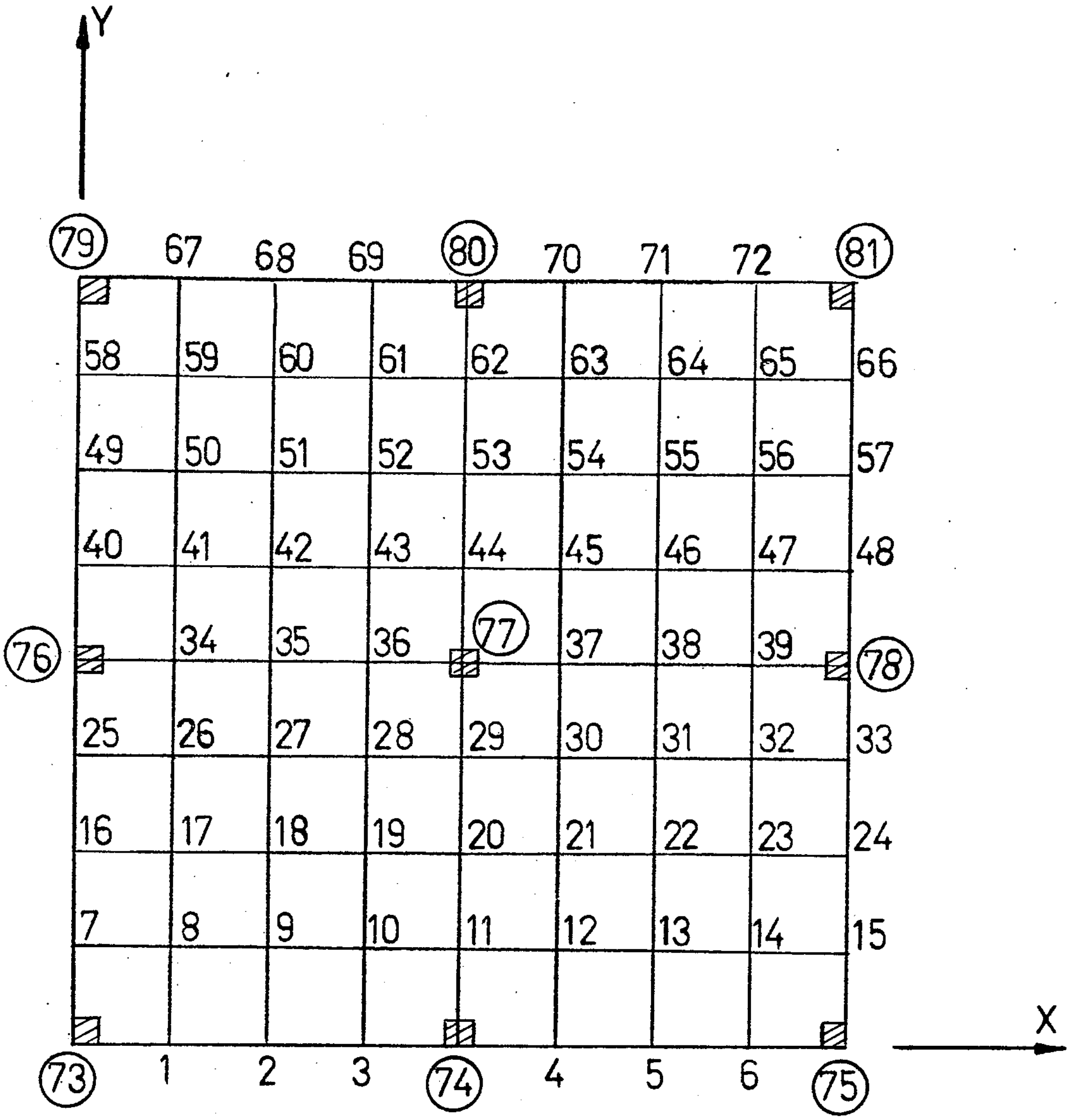
$$\left\{\begin{array}{c} P_I \\ \hline P_E \end{array}\right\} = \left[\begin{array}{c|c} K_{II} & K_{IE} \\ \hline K_{EI} & K_{EE} \end{array}\right] \left\{\begin{array}{c} D_I \\ \hline D_E \end{array}\right\} \qquad (3.7)$$

Gaussian Elimination can be performed on the stiffness matrix, K, in equation (3.7). Unfortunately, for almost every flat plate floor system, the order of matrix K is very large and computer core capacity precludes storing the entire matrix K for the Gaussian Elimination.

Consequently, a special technique was developed in this study, in which stiffness matrix generation and condensation are performed progressively.

To explain the procedure, consider the flat plate model shown in Figure 3.4. All nodes are automatically numbered sequentially by the computer, beginning with all "internal" nodes I, then all "external" (i.e. panel corner) nodes E. The numbers increase in the sequence of increasing X direction and then increasing Y direction, as shown. The floor stiffness matrix generation and condensation then involves repeated cycles of an operation in which a portion of the floor stiffness matrix relating to m nodes is assembled and condensation performed on the nodes, as indicated in Figure 3.5. In the figure, N is the total number of cycles required to complete the assembly and condensation procedure. The value of N being dependent upon the number of nodes present, including both "internal" nodes I and panel corner nodes E. Each cycle of the assembly-condensation operation involves m nodes except for the Nth (i.e. last) cycle, in which the number of nodes involved may be less than m, depending on the total number of nodes present. In Cycle 1, the assembly-condensation operation is performed for node 1 to node m, cycle 2 from node m + 1 to node 2m, cycle 3 from node 2m + 1 to node 3m and so on. For a typical cycle of the operation, say the nth cycle, the following steps are performed.

(a) Assemble the portion of the stiffness matrix for the m nodes concerned. The stiffness matrix is assembled one node (one row) at a time. The complete submatrices for the m nodes are shown in Figure 3.6. If the m nodes considered in the nth cycle are "internal" nodes, then the submatrices in Figure 3.6 are $\underline{K}_{II}$



NUMBERS 1–72 ARE "INTERNAL" NODES, DESIGNATED I
NUMBERS 73–81 ARE "EXTERNAL" NODES, DESIGNATED E

FIGURE 3·4 FLAT PLATE MODEL SHOWING COMPUTER GENERATED NUMBERING SCHEME

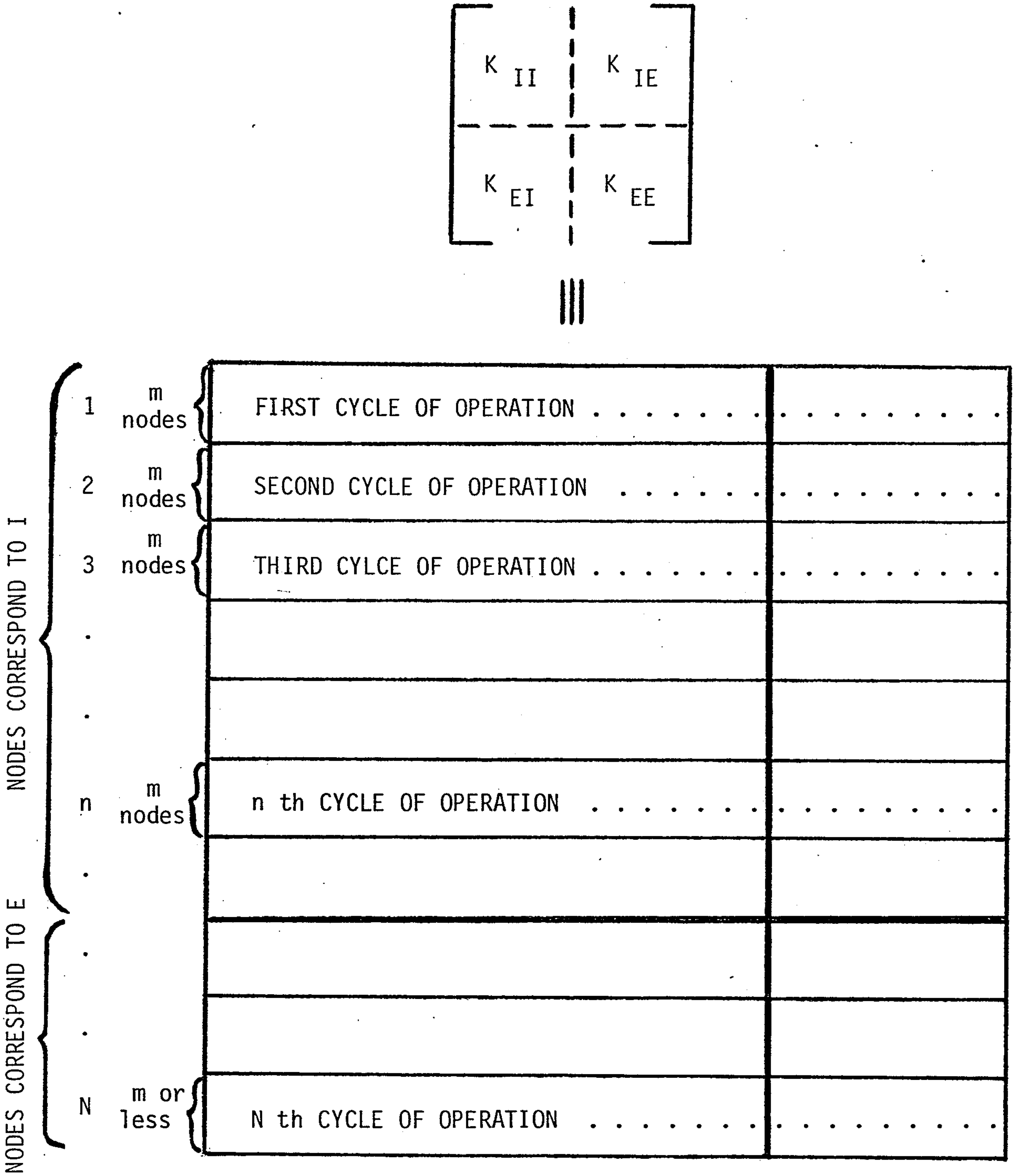


FIGURE 3.5 K MATRIX IN DETAILED FORM

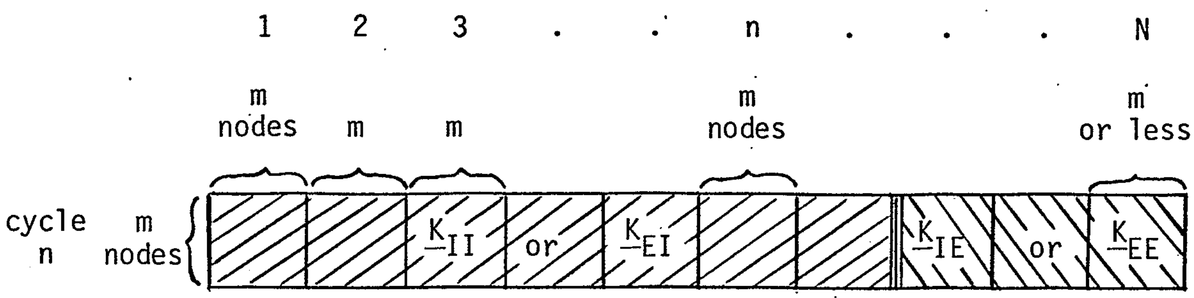


FIGURE 3.6 Submatrices for the nth Cycle of Operation

and $\underline{K}_{IE}$. $\underline{K}_{II}$ contains the forces applied at the m particular "internal" nodes due to displacements at all "internal" nodes which are directly connected to the m nodes considered. $\underline{K}_{IE}$ contains the forces at the m particular "internal" nodes corresponding to displacements at all panel corner nodes which are directly connected to the m particular nodes. On the other hand, if the m nodes considered are panel corner nodes, then the submetrices in Figure 3.6 are $\underline{K}_{EI}$ and $\underline{K}_{EE}$. $\underline{K}_{EI}$ contains the forces applied at the m particular panel corner nodes due to displacements at all "internal" nodes which are directly connected to the m nodes considered. $\underline{K}_{EE}$ contains the forces at the m particular panel corner nodes corresponding to displacements at all panel corner nodes which are directly connected to the m particular nodes.

(b) Perform Gaussian Elimination on the submatrices shown in Figure 3.6. If the m particular nodes of the Nth cycle are "internal" nodes, then condensation stops at the columns corresponding to the m nodes of the Nth cycle, forming an upper triangular submatrix at these column locations, as shown in Figure 3.7. In the figure, the prime of $\underline{K}'_{II}$ and $\underline{K}'_{IE}$ indicates that the submatrices have been modified by Gaussian Elimination. All blank submatrices are null matrices.

In performing the Gaussian Elimination, the previously saved submatrices from cycles 1 through n-1 are retrieved, one at a time, so that submatrices $\underline{K}_{II}$ and $\underline{K}_{IE}$ in Figure 3.7 can be modified to give the condensed submatrices $\underline{K}_{II}$ and $\underline{K}_{IE}$ as shown in the same figure.

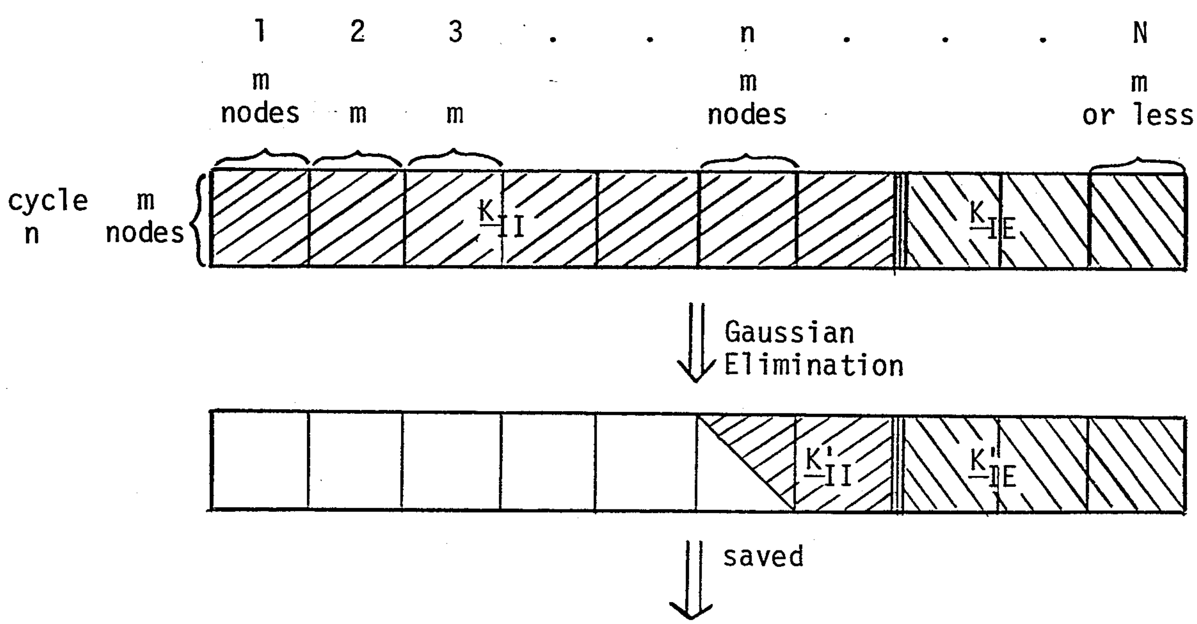


FIGURE 3.7 nth Cycle Condensation (For m "internal" nodes)

On the other hand, if the m particular nodes of the Nth cycle are panel corner nodes, the assembly-condensation procedure is similar to that for "internal" nodes except that condensation stops when $\underline{K}_{EI}$ becomes a null submatrix, as shown in Figure 3.8. Again, the prime of $\underline{K}'_{EE}$ indicates that the submatrix has been modified by Gaussian Elimination.

(c) If the m particular nodes of the Nth cycle are "internal" nodes, save the modified submatrices $\underline{K}'_{II}$ and $\underline{K}'_{IE}$ from step (b) as indicated in Figure 3.7. These submatrices are saved for subsequent cycles of the assembly-condensation operation and for subsequent backsubstitution for slab forces after the complete structure has been analyzed to determine the displacements at all panel corners.

(d) Repeat the above steps for the next cycle of the operation. Thus after N cycles of the operation, the assembly-condensation procedures is completed. The modified floor stiffness matrix has the form shown in Figure 3.9. In the figure, K_c is small compared to the entired stiffness matrix and it is stored in core. K'_{II} is an upper triangular matrix and K'_{IE} is a modified submatrix. Both are large matrices and they are stored in secondary storage, to be subsequently retrieved.

The force-displacement equations for the entire floor plate then have the form

$$\begin{Bmatrix} 0 \\ P_E \end{Bmatrix} = \left[\begin{array}{c|c} K'_{II} & K'_{IE} \\ \hline 0 & K_c \end{array}\right] \begin{Bmatrix} D_I \\ D_E \end{Bmatrix} \tag{3.8}$$

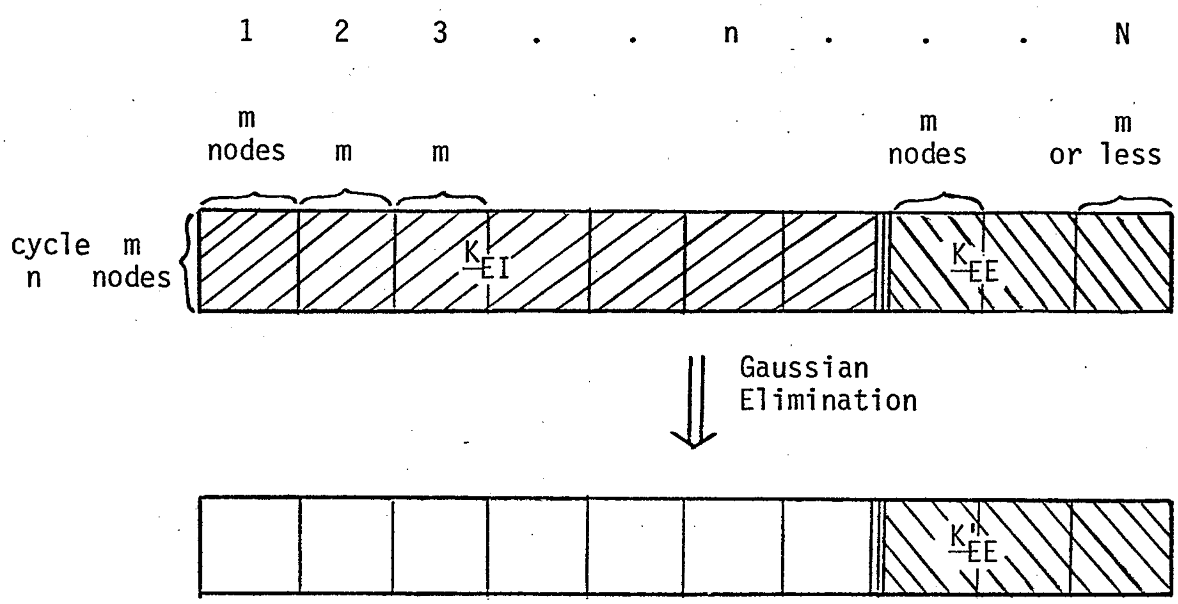
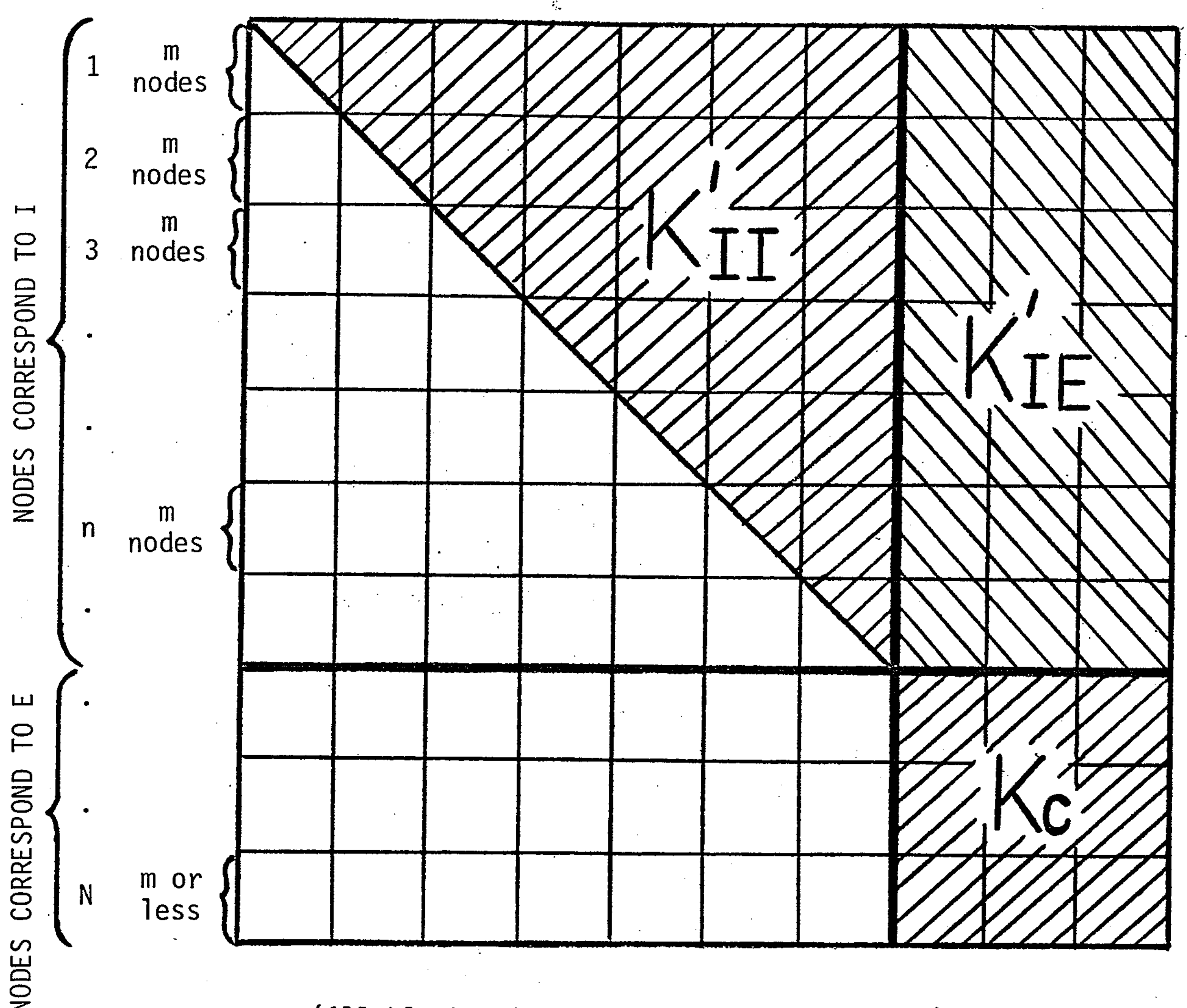


FIGURE 3.8 nth Cycle Condensation (For m Panel Corner Nodes)



(All blank submatrices are null matrices)

FIGURE 3.9 MODIFIED K MATRIX

where 0 stands for null matrix.

The method described above requires much less core storage than if the operations were performed on the entire stiffness matrix. Storage space is required for 2m stiffness submatrices plus the K_c matrix. In this study, an m value of 8 was used, because this value was found to be optimal in terms of computer storage, central processing time, I/O counts and other computer resources.

CHAPTER IV

THREE DIMENSIONAL FRAME ANALYSIS

4.1 Introduction

In this chapter, the assembly of the stiffness matrix for the entire structure is described. The condensed floor bending stiffness matrix K_C, described in the previous chapter, is combined with the stiffness matrices for the framing system, assembled by the direct stiffness method, to form the structure stiffness matrix. The floor system is assumed to be rigid in its own plane. Hence, as shown in Figure 4.1, all in-plane forces and displacements can be expressed in terms of three displacement components; i.e. translations X_R and Y_R and in-plane rotation θ_R, at some reference point on the floor plane. In this study, the origin of the global coordinate system at each floor level is taken as the reference point for the floor (for dynamic analysis, the floor reference point is taken at the center of mass of each floor). With the three in-plane degrees of freedom transformed to the floor reference point, each joint (column node or panel corner node) on the floor is left with three out-of-plane degrees of freedom, i.e. rotations about X and Y directions and translation in the Z direction. Thus the structure stiffness matrix corresponds to three degrees of freedom per joint at each floor, times the number of floors, plus three degrees of freedom per storey. The three out-of-plane degrees of freedom at every joint are eliminated by static condensation and the final structure stiffness matrix corresponds to three degrees of freedom per floor, i.e. horizontal displacements at all floor levels.

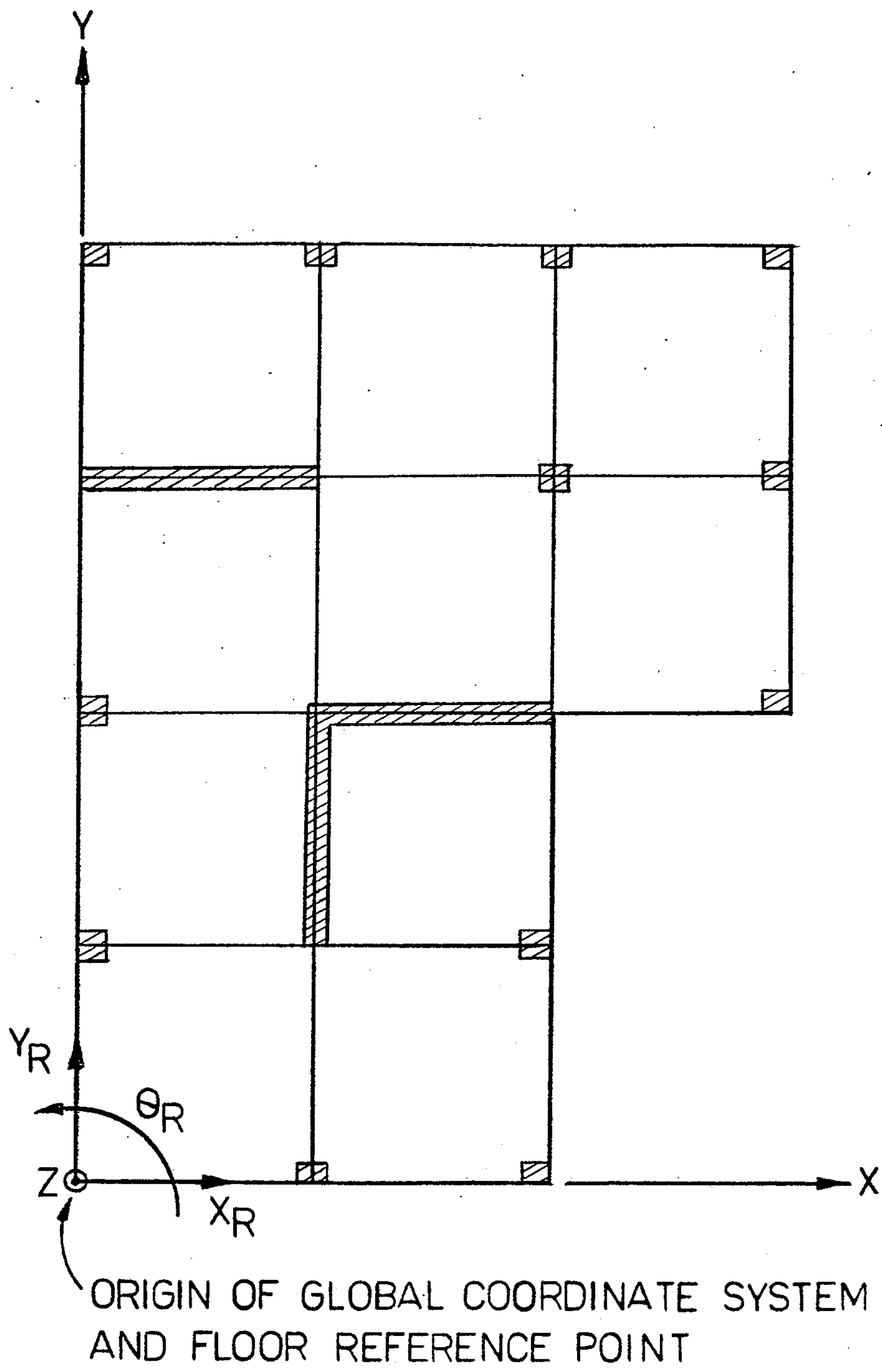


FIGURE 4·1 FLOOR PLAN SHOWING GLOBAL COORDINATE SYSTEM AND FLOOR REFERENCE POINT

4.2 Procedure for Assembling Structure Stiffness Matrix

With the structure degrees of freedom appropriately ordered, the equilibrium equations for the structure, obtained by assembling the stiffnesses of all structural elements, i.e. columns, beams, wall panels, braces, etc., have the tridiagonal form shown below:

$$
\left[\begin{array}{c} R_1 \\ R_2 \\ R_3 \\ \cdot \\ \cdot \\ R_n \\ R_{n+1} \\ \cdot \\ \cdot \\ R_{N-1} \\ R_N \\ \hline P_L \end{array}\right]
=
\left[\begin{array}{ccccccccccccc|c}
K_1 & C_1 & & & & & & & & & & & & E_1 \\
C_1^T & K_2 & C_2 & & & & & & & & & & & E_2 \\
 & C_2^T & K_3 & C_3 & & & & & & & & & & E_3 \\
 & & \cdot & \cdot & \cdot & & & & & & & & & \cdot \\
 & & & \cdot & \cdot & \cdot & & & & & & & & \cdot \\
 & & & & & C_{n-1}^T & K_n & C_n & & & & & & E_n \\
 & & & & & & C_n^T & K_{n+1} & C_{n+1} & & & & & E_{n+1} \\
 & & & & & & & \cdot & \cdot & \cdot & & & & \cdot \\
 & & & & & & & & \cdot & \cdot & \cdot & & & \cdot \\
 & & & & & & & & & & C_{N-2}^T & K_{N-1} & C_{N-1} & E_{N-1} \\
 & & & & & & & & & & & C_{N-1}^T & K_N & E_N \\ \hline
E_1^T & E_2^T & E_3^T & \cdot & & \cdot & E_n^T & E_{n+1}^T & \cdot & & \cdot & E_{N-1}^T & E_N^T & K_L
\end{array}\right]
\left[\begin{array}{c} r_1 \\ r_2 \\ r_3 \\ \cdot \\ \cdot \\ r_n \\ r_{n+1} \\ \cdot \\ \cdot \\ r_{N-1} \\ r_N \\ \hline r_L \end{array}\right]
\qquad (4.1)
$$

where

N is the number of storeys in the structure;

r_n is the vector of joint displacements at storey level n, it is of order 3 times the number of joints at the level. The 3 degrees of freedom considered at each joint are the vertical displacement and rotation about X and Y axis;

R_n is the vector of joints forces corresponding to joint displacements r_n. The joint forces are due to vertical loads on beams;

r_L is the vector of 3N lateral storey displacements components (in-plane floor displacements);

and P_L is the vector of storey lateral loads.

K_n in Equation (4.1) is the bending stiffness matrix for level n, consisting of two parts,

$$K_n = [K_f]_n + [K_c]_n$$

where

$[K_f]_n$ is the bending stiffness matrix of level n from all structural elements of the frame and $[K_c]_n$ is the "flat plate floor" bending stiffness matrix for level n as obtained previously.

Obviously, Gaussian Elimination may be performed on the full system until the equilibrium equations are condensed to a size corresponding to only the lateral degrees of freedom at each floor level, i.e. 3 degrees of freedom per floor. The condensed equations appear as shown below.

$$
\left[\begin{array}{c}
R'_1 \\ R'_2 \\ R'_3 \\ \cdot \\ \cdot \\ R'_n \\ R'_{n+1} \\ \cdot \\ \cdot \\ R'_{N-1} \\ R'_N \\ \hdashline P_L + R'_L
\end{array}\right]
=
\left[\begin{array}{cccccccccccc:c}
K'_1 & C'_1 & & & & & & & & & & & E'_1 \\
& K'_2 & C'_2 & & & & & & & & & & E'_2 \\
& & K'_3 & C'_3 & & & & & & & & & E'_3 \\
& & & \cdot & \cdot & & & & & & & & \cdot \\
& & & & \cdot & \cdot & & & & & & & \cdot \\
& & & & & K'_n & C'_n & & & & & & E'_n \\
& & & & & & K'_{n+1} & C'_{n+1} & & & & & E'_{n+1} \\
& & & & & & & \cdot & \cdot & & & & \cdot \\
& & & & & & & & \cdot & \cdot & \cdot & & \cdot \\
& & & & & & & & & K'_{N-1} & C'_{N-1} & & E'_{N-1} \\
& & & & & & & & & & K'_N & & E'_N \\
\hdashline
& & & & & & & & & & & & K'_L
\end{array}\right]
\left[\begin{array}{c}
r_1 \\ r_2 \\ r_3 \\ \cdot \\ \cdot \\ r_n \\ r_{n+1} \\ \cdot \\ \cdot \\ r_{N-1} \\ r_N \\ \hdashline r_L
\end{array}\right]
\qquad (4.2)
$$

In Equation (4.2), all blank submatrices are null matrices. The prime indicates that the coefficients are modified by the condensation process. K'_n is an upper triangular matrix and R'_n is the modified vector of joint loads. The vector R'_L indicates that the lateral load submatrix P_L is modified by the elimination process due to vertical loading on the structure. This term represent the sidesway effects of a non-symmetrical structure of non-symmetrical vertical loading. The matrix K_L' represents the condensed lateral stiffness matrix; i.e. the stiffness matrix of the structure in terms of only the lateral storey displacements.

As was the case in the assembly and condensation of the stiffness matrix for a single floor plate, the entire set of structure stiffness equations cannot be stored in core for one operation of Gaussian Elimination. Within the computer program however, the following approach is adopted in order to reduce core storage requirements. The assembly and reduction process is carried out systematically storey by storey from the top of the structure such that at any level, the following system is considered:

$$\begin{Bmatrix} R_n' \\ R_{n+1}' \\ P_L + R_L' \end{Bmatrix} = \begin{bmatrix} K_n' & C_n' & E_n' \\ C_n'^T & K_{n+1}' & E_{n+1}' \\ E_n'^T & E_{n+1}'^T & K_L' \end{bmatrix} \begin{Bmatrix} r_n \\ r_{n+1} \\ r_L \end{Bmatrix} \tag{4.3}$$

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

At each level the following steps are performed.

(a) Assemble the stiffnesses for the individual members of the framing system for level n. Combine the flat plate floor bending stiffness with the framing stiffness to form the equations shown in Equation (4.3). The framing system for level n is shown in Figure 4.2.

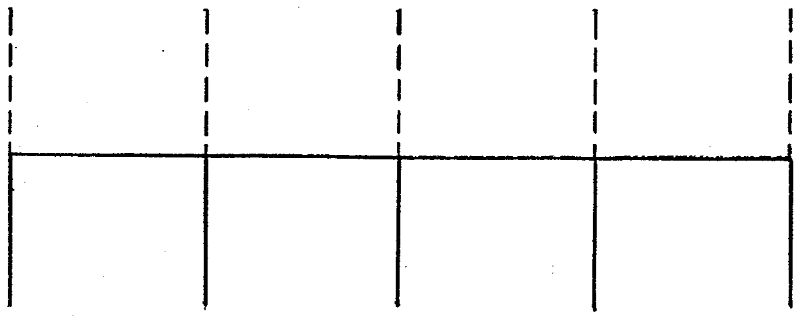


FIGURE 4.2 Framing System for Level n

(b) Perform the Gaussian Elimination on Equation (4.3) to give the following

$$\begin{Bmatrix} R'_n \\ R'_{n+1} \\ P_L+R'_L \end{Bmatrix} = \begin{bmatrix} K'_n & C'_n & E'_n \\ 0 & K'_{n+1} & E'_{n+1} \\ 0 & E'^T_{n+1} & K'_L \end{bmatrix} \begin{Bmatrix} r_n \\ r_{n+1} \\ r_L \end{Bmatrix} \qquad (4.4)$$

where K'_n is an upper triangular submatrix, and 0 denotes a null matrix.

(c) Save these reduced equations for subsequent backsubstitution for joint displacements r_n.

(d) Rearrange the submatrices in Equation (4.4) appropriately in order to proceed to the next level, i.e. level n+1. In Equation (4.4), the submatrices which have contribution to the stiffness of level n+1 are K'_{n+1}, E'_{n+1}, E'^T_{n+1} and K'_L. These submatrices are rearranged to occupy the positions shown below.

$$\begin{Bmatrix} R'_{n+1} \\ 0 \\ P_L+R'_L \end{Bmatrix} = \begin{bmatrix} K'_{n+1} & 0 & E'_{n+1} \\ 0 & 0 & 0 \\ E'^{T}_{n+1} & 0 & K'_L \end{bmatrix} \begin{Bmatrix} r_{n+1} \\ r_{n+2} \\ r_L \end{Bmatrix} \tag{4.5}$$

(e) Repeat the above steps for the next level, i.e. level n+1. This involves assembling stiffness submatrices C'_{n+1}, C'^{T}_{n+1}, K'_{n+2}, E'^{T}_{n+2} and augmenting submatrix K'_L to incorporate the contributions at the n+1 st level. These submatrices are then substituted into the stiffness matrix shown in Equation (4.5). The equation then takes the following form, which is equivalent to that of Equation (4.3).

$$\begin{Bmatrix} R'_{n+1} \\ R'_{n+2} \\ P_L+R'_L \end{Bmatrix} = \begin{bmatrix} K'_{n+1} & C'_{n+1} & E'_{n+1} \\ C'^{T}_{n+1} & K'_{n+2} & E'_{n+2} \\ E'^{T}_{n+1} & E'^{T}_{n+2} & K'_L \end{bmatrix} \begin{Bmatrix} r_{n+1} \\ r_{n+2} \\ r_L \end{Bmatrix} \tag{4.6}$$

Perform Gaussian Elimination on Equation (4.6). Thus after the elimination is completed for joint displacements at all storey levels, the submatrix K'_L left is the required lateral stiffness matrix for the structure.

4.3 Solution for Lateral and Gravity Loads

From Equation (4.2), the lateral stiffness matrix for the structure can be rewritten as

$$P_L+R'_L = K'_L\ r_L \tag{4.6}$$

where P_L is the lateral load vector and R_L' is the modification vector due to vertical loading. It may be noted that the global stiffness matrix K_L' is a full matrix, but it is of course very small compared to the total number of degrees of freedom associated with all joints in the structure. Equation (4.6) can be solved directly by Gaussian Elimination giving a vector of global lateral displacements, r_L. To complete the solution for the structure, the following system is considered.

$$R_n' = \begin{bmatrix} K_n' & C_n' & E_n' \end{bmatrix} \begin{Bmatrix} r_n \\ r_{n+1} \\ r_L \end{Bmatrix} \tag{4.7}$$

Note that these are the equations which were reduced, then saved at each level, n, of the structure (refer to Equation 4.4). R_n' is the modified vertical load vector and r_n is the vector of joint displacements at storey level n. K_n' is a triangularized submatrix. At any stage, n, r_{n+1} and r_L are known and so r_n is computed by back substitution. To start this sequence, it is noted that for n=N (the number of storeys in the structure), r_{n+1} represents the displacements at the foundation which are zero since columns are assumed rigidly connected to the foundation. Thus the joint displacements are computed successively storey by storey, starting from the bottom level. Individual member forces may be computed at the same time, as follow.

For column member forces, Equation (2.9b)

$$\underline{S}_c = \underline{K}_c \theta_c$$

is used. Where $\underset{\sim}{S}_c$ is the vector of column member forces, $\underset{\sim}{K}_c$ is the column element stiffness matrix and θ_c is the vector of column displacements which are known after r_n is computed.

Similarly, for beam forces, Equation (2.14)

$$\underset{\sim}{S}_b = \underset{\sim}{K}_b \, \underset{\sim}{\theta}_b$$

is used, where the symbol b denotes beam.

Equations (2.17b) and (2.24b) are used for wall panel forces and bracing element forces respectively. That is

$$\underset{\sim}{S}_p = \underset{\sim}{K}_p \, \underset{\sim}{\theta}_p$$

and

$$\underset{\sim}{S}_D = \underset{\sim}{K}_D \, \underset{\sim}{\theta}_D$$

where p and D designate panel and diagonal element (brace) respectively.

4.4 Solution for Slab Forces

Equation (3.8) is reproduced below

$$\begin{Bmatrix} 0 \\ P_E \end{Bmatrix} = \begin{bmatrix} K'_{II} & K'_{IE} \\ 0 & K_C \end{bmatrix} \begin{Bmatrix} D_I \\ D_E \end{Bmatrix}$$

where P_E and D_E are respectively, force and displacement vectors at all panel corner nodes (column nodes), denoted by E. D_I is the displacement vector at all internal slab nodes, denoted by I. K_C is the condensed flat plate floor stiffness matrix and K'_{II} is an upper triangular matrix which, together with K'_{IE} are previously saved. Note that r_n, the vector of joint displacements at leven n, as obtained from Equation (4.7), is the same as the vector D_E for level n. Thus for any storey level n, with P_E, K_C and D_E in Equation (4.8) known, the internal slab displacements D_I, can be computed by backsubstitution.

The slab forces for each finite element can then be calculated from Equation (2.8b)

$$\bar{M} = S\,\bar{u}$$

where $\bar{M}$ is the vector of plate moments at each node of the element, $\bar{u}$ is the vector of nodal displacement coordinates and S is the corresponding stress matrix. $\bar{u}$ are known quantities after D_I are solved. Slab moments can be calculated for any finite element required, using Equation (2.8b).

The slab moments at any node are calculated based on the average of the slab moments from all finite elements framing into the node.

CHAPTER V

ILLUSTRATIVE EXAMPLES

5.1 Introduction

Several examples are presented to demonstrate and check the validity and accuracy of the various aspects of the analysis procedure. Due to the unavailability of relevant experimental data, the results are compared with those obtained by other analysis methods, such as a special purpose analysis program [10] and a general structural analysis program [15].

In the first example, a planar frame is analysed and the results are compared with those obtained from a planar frame analysis program PFRAME [14]. This example verifies to some extent the "frame analysis" portion of the computer program TABSLAB developed in this study.

In the second example, a ten-storey flat plate building is analysed with each floor panel represented by a single finite element. The results are compared with those reported by French [10]. A detailed tabulation and comparison of results is presented for this example.

The convergence of the analysis procedure is examined in the last two examples. In the third example, the structure of Example 2 is analysed using different numbers of finite elements to represent the typical floor panel. The convergence pattern of the example is discussed. Since it is very expensive to analyse a 10-storey structure with a large number of finite elements per panel using a general analysis program, the results from the third example are given without comparison with other solutions.

In the last example, a two-panel, two-storey simple building is analysed. Different numbers of finite elements per panel are used and the results are compared with those obtained from a general structural analysis program SAP4 [15]. Convergence of results from the two analyses are then compared.

5.2 Frame Analysis

The purpose of the first example is to illustrate the correctness of the "frame analysis" part of the computer program TABSLAB. The example also demonstrates the capability of TABSLAB for analysing framed structures with no in-plane deformation.

Example 1:

The structure shown in Figure 5.1 is analysed for the loading shown in the figure. All members have cross-sectional areas of 1 sq. ft. and a modulus of elasticity of 576,000 kips/ft^2. This structure is analysed using TABSLAB and PFRAME. To eliminate the axial deformations of beams, all beam cross-sectional areas in the PFRAME analysis are assumed to be 1,000,000 square inches.

Lateral deflections at various level, as calculated by the two programs, are given in Table 5.1. The percentage differences for corresponding lateral deflections are all within 1.93%. It is noted that the deflections by TABSLAB analysis are slightly larger than those by PFRAME analysis. This may be explained by the fact that PFRAME neglects member shearing deformation. The inclusion of shearing deformation in TABSLAB slightly increases the flexibility. Table 5.2 lists the end moments for all members, as calculated by the two programs. The percentage differences for all end moments are within 0.5%.

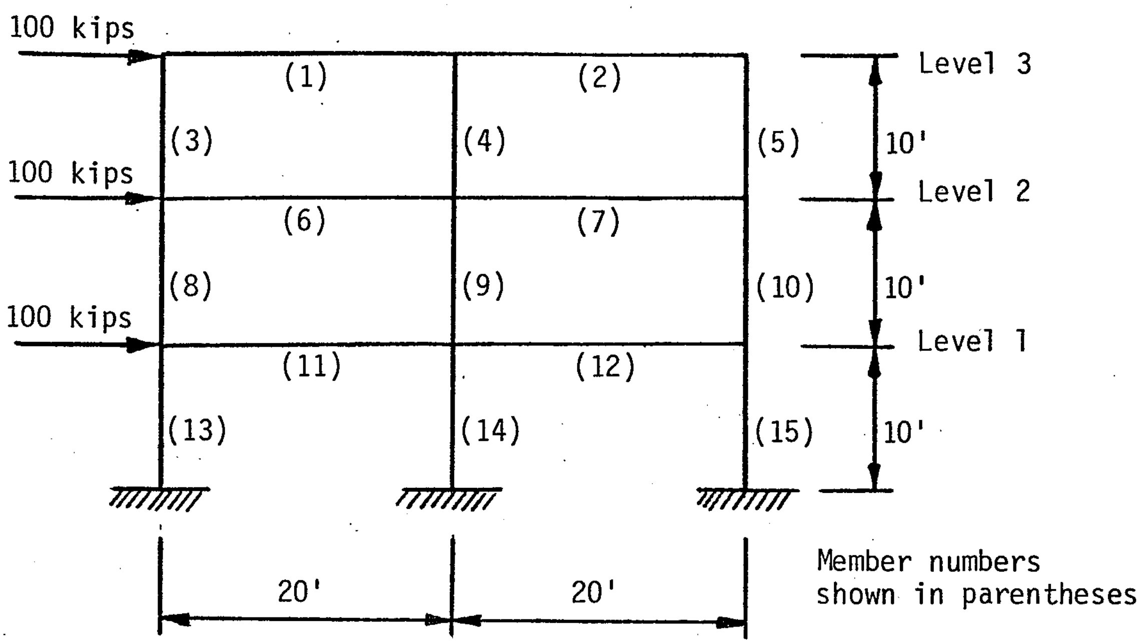


FIGURE 5.1 THREE-STOREY PLANAR FRAME

Level	TABSLAB	PFRAME	% differences
3	12.35	12.16	1.56
2	9.29	9.14	1.64
1	4.22	4.14	1.93

TABLE 5.1 PLANAR FRAME LATERAL DEFLECTIONS (INCHES)

Member	End Moments (left end or top)			End Moments (right end or bottom)		
	TABSLAB	PFRAME	% Differences	TABSLAB	PFRAME	% Differences
1	-183.539	-183.120	0.229	-160.470	-160.470	0.002
2	-160.470	-160.467	0.002	-183.539	-183.120	0.229
3	183.539	183.120	0.229	57.425	57.705	0.488
4	320.941	320.934	0.002	197.133	197.414	0.143
5	183.539	183.120	0.229	57.425	57.705	0.488
6	-370.656	-370.510	0.039	-346.435	-346.909	0.137
7	-346.435	-346.909	0.137	-370.656	-370.510	0.039
8	313.231	312.804	0.137	217.750	217.464	0.132
9	495.737	496.404	0.135	442.302	443.059	0.171
10	313.231	312.805	0.136	217.750	217.464	0.132
11	-492.847	-493.187	0.069	-447.603	-448.588	0.220
12	-447.603	-448.588	0.220	-492.847	-493.186	0.069
13	275.097	275.723	0.228	636.528	635.081	0.228
14	452.904	454.117	0.268	723.847	724.277	0.059
15	275.097	275.722	0.227	636.528	635.080	0.228

TABLE 5.2 Planar Frame End Moments (Kips-Ft.)

5.3 Analysis of Flat Plate Building Structures

The second example illustrates the capability of analysing flat plate building structures by TABSLAB. The program permits the analysis of flat plate structures, which incorporate shear walls, as well as columns. However, since published analysis results are not available for such structures, a flat plate structure with columns only, and no shear walls, is considered here.

Example 2

A 3 x 3 bay 10-storey flat plate building structure with square columns, is analysed with each floor panel represented by a single finite element. The results are compared with those obtained by French [10], who also used a single finite element to represent each floor panel. French's analysis program is subsequently referred to as SEP. Figure 5.2(a) shows the elevation of the structure with lateral loading acting on both sides of the frame. Figure 5.2(b) shows the typical plan view of the structure and the numbering scheme used for the column nodes. All flat plate floors have constant thicknesses of 230 mm. All columns are 760 mm square. All material has a modulus of elasticity of 20690 MPa and a Poisson's ratio of 0.1.

Figure 5.3 shows the deflected shape of the structure as predicted by TABSLAB and SEP analyses. The corresponding magnitudes of the lateral deflections are listed in Table 5.3, the percentage differences in deflection between the two analyses are also listed. It is noted that the percentage differences vary from 2.47 at the top floor to a negligible value at the bottom floor. This variation may be due to the fact that during the equation solving process, the backsubstitution for lateral displacements

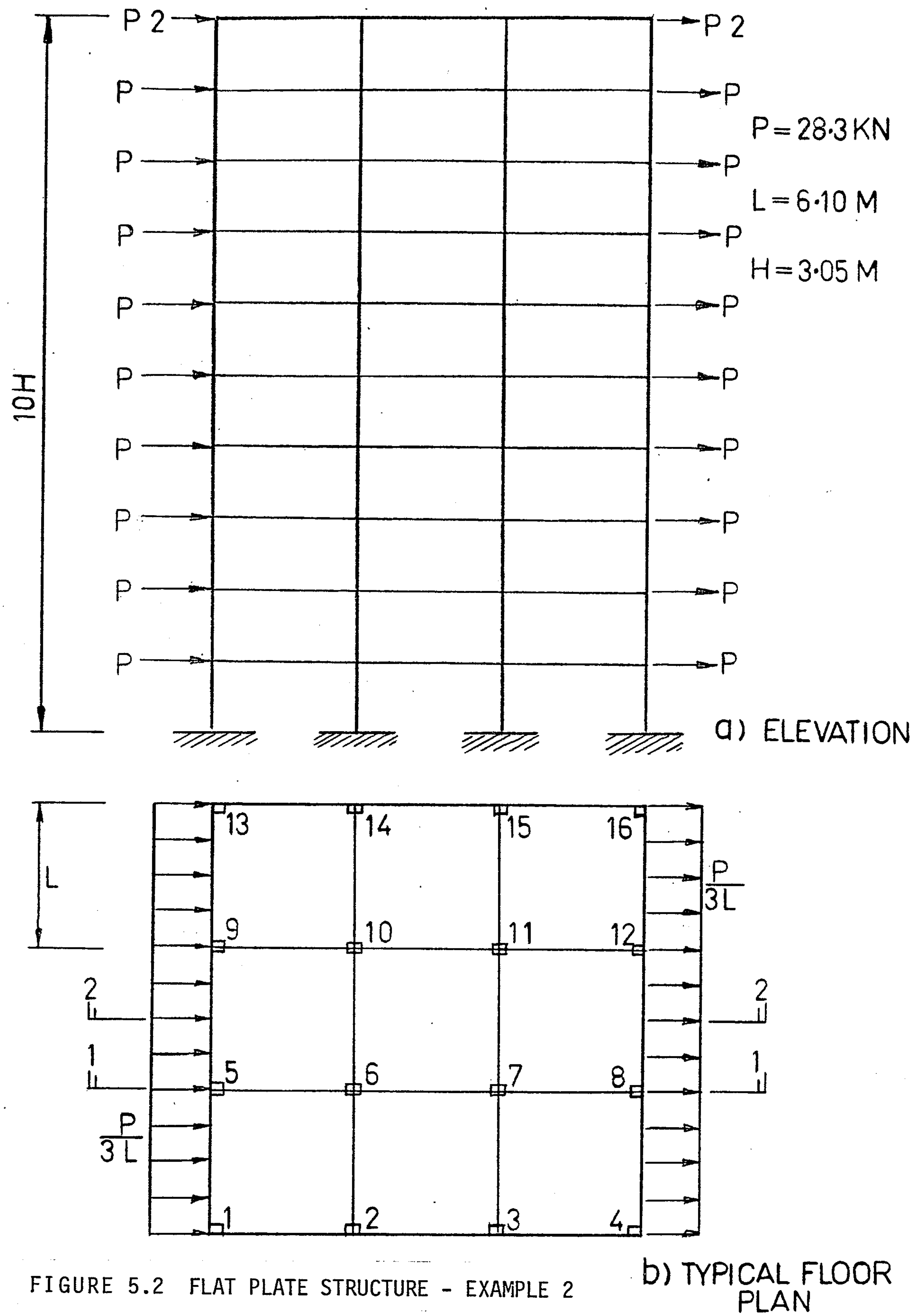


FIGURE 5.2 FLAT PLATE STRUCTURE - EXAMPLE 2

Floor Level	Lateral Deflection (mm)		
	SEP	TABSLAB	% DIFF.
10	10.34	10.09	2.47
9	10.02	9.78	2.45
8	9.52	9.30	2.37
7	8.82	8.62	2.32
6	7.88	7.71	2.20
5	6.70	6.57	1.98
4	5.32	5.23	1.72
3	3.76	3.71	1.35
2	2.14	2.13	0.46
1	0.70	0.70	0.00

TABLE 5.3 FLAT PLATE BUILDING LATERAL DEFLECTIONS

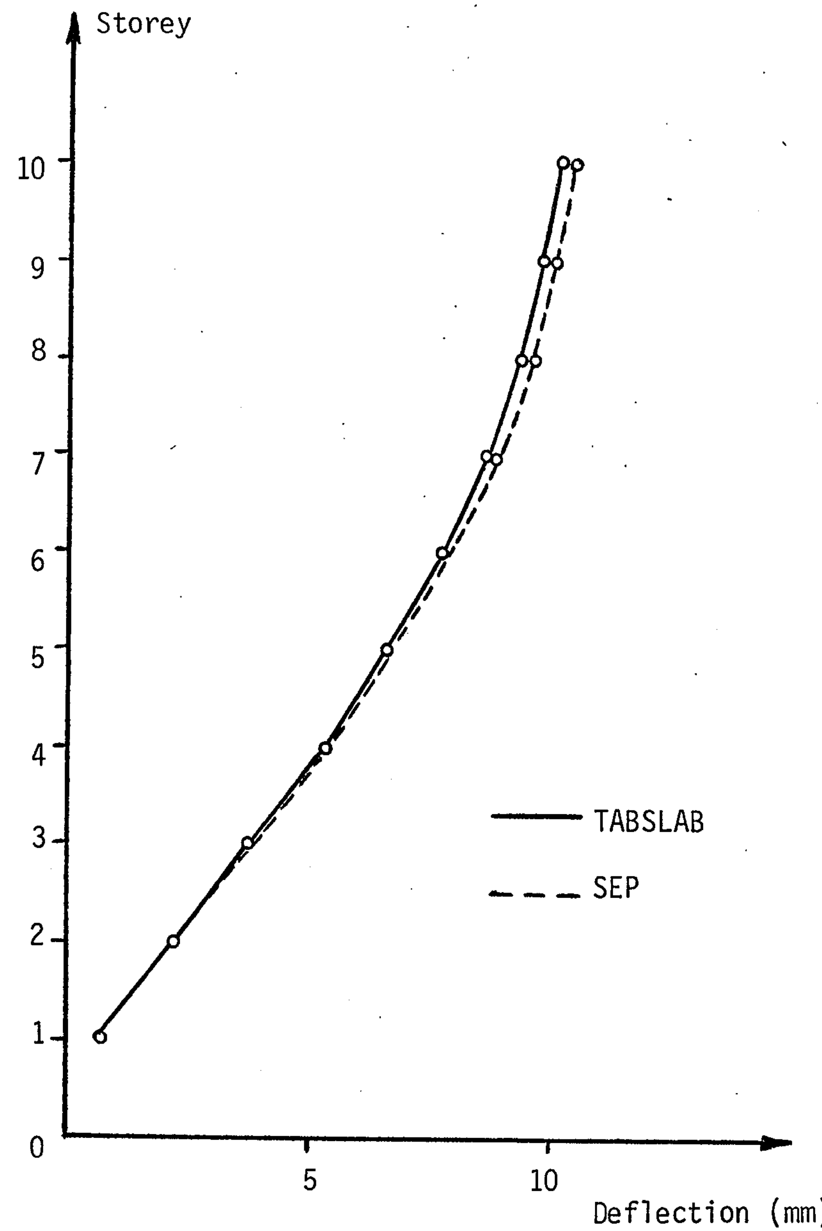


FIGURE 5.3 PLOTS OF LATERAL DEFLECTIONS

is performed starting from the bottom floor to the top. Thus, due to round-off error, the upper level displacements, which are obtained from the later stage of the backsubstitution, may have less accuracy than the lower level displacements. Since TABSLAB is in double precision, the round-off errors would be expected to be mainly in the results obtained from SEP, which is in single precision.

Table 5.4 gives the moments at the top and bottom ends of the various columns. With consideration of symmetry for both the structure and lateral loading, only moments at columns numbered 1, 2, 5 and 6 in Figure 5.2(b) are given. In the table, results from both SEP and TABSLAB analyses are given. The corresponding absolute differences between the two analyses are also given. The maximum absolute difference is 5.68 KN-m which, when compared with the maximum end moment 127.64 KN-m, is about 4.45%. Many of the differences are within 3 Kn-m, which is 2.35% of the maximum end moment. Figure 5.4 shows a plot of end moments for column 2. Plots of end moments for other columns are similar.

Table 5.5 gives the shear forces at the four columns mentioned. The maximum absolute difference between the results predicted by both analysis is 2.98 KN, which is 6.57% of the maximum column shear force (45.34 KN). Many of the differences are within 2 KN, which is about 4.4% of the maximum column shear forces. It is noted that shear forces on column 6 are larger than those on column 1, 2 and 5. Thus interior columns will carry a larger proportion of the lateral load than exterior columns. Figure 5.5 shows a plot of shear forces for column 2.

The above are results obtained from the analysis representing each slab panel by a single finite element. These results are in agreement with

Floor Level		COLUMN 1			COLUMN 2			COLUMN 5			COLUMN 6		
		SEP	TABSLAB	Abs. Diff.	SEP	TABSLAB	Abs. Diff.	SEP	TABSLAB	Abs. Diff.	SEP	TABSLAB	Abs. Diff.
10	top	5.26	4.48	0.78	10.83	9.32	1.51	7.61	7.87	0.26	15.90	16.71	0.81
	btm	-8.16	-8.25	0.09	-3.94	-4.57	0.63	-6.20	-5.49	0.71	0.31	1.51	1.2
9	top	15.96	14.99	0.97	20.02	18.61	1.41	18.15	18.03	0.12	24.61	25.13	0.52
	btm	-8.23	-8.46	0.23	-3.00	-3.74	0.74	-5.39	-4.45	0.94	2.75	4.61	1.86
8	top	20.39	19.03	1.36	27.47	25.24	2.23	24.30	24.44	0.14	35.25	36.28	1.03
	btm	-6.98	-7.40	0.42	0.78	0.47	0.31	-2.54	-1.24	1.30	9.44	12.01	2.57
7	top	24.10	22.36	1.74	33.39	30.59	2.8	29.55	29.83	0.28	43.92	45.58	1.66
	btm	-4.35	-5.10	0.75	5.79	3.98	1.81	1.64	3.22	1.58	17.32	20.59	3.27
6	top	26.65	24.62	2.03	38.43	35.07	3.36	33.70	34.32	0.62	51.92	54.28	2.36
	btm	-0.83	-1.97	1.14	11.74	9.28	2.46	6.79	8.54	1.75	26.19	30.06	3.87
5	top	28.15	25.93	2.22	42.22	38.44	3.78	36.75	37.72	0.97	58.48	61.63	3.15
	btm	4.13	2.51	1.62	18.90	15.75	3.15	13.21	15.05	1.84	36.02	40.34	4.32
4	top	27.61	25.39	2.22	43.60	39.64	3.96	37.50	38.99	1.49	62.20	66.23	4.03
	btm	11.91	9.73	2.18	28.42	24.52	3.9	22.20	23.89	1.69	47.64	52.13	4.49
3	top	22.77	20.85	1.92	39.72	36.04	3.68	33.38	35.47	2.09	59.52	64.48	4.96
	btm	26.20	23.37	2.83	43.07	38.52	4.55	36.81	38.00	1.19	62.83	66.95	4.12
2	top	8.01	6.93	1.08	24.00	21.35	2.65	18.11	20.93	2.82	42.74	48.42	5.68
	btm	54.23	50.86	3.37	69.44	64.43	5.01	63.80	64.06	0.26	87.18	89.85	2.67
1	top	-28.22	-27.68	0.54	-18.61	-18.80	0.19	-22.10	-19.02	3.08	-7.44	-2.20	5.24
	btm	117.25	114.58	2.67	122.06	118.58	3.48	120.20	118.48	1.72	127.64	126.06	1.58

TABLE 5.4 Column End Moments (Kilonewton-Meter)

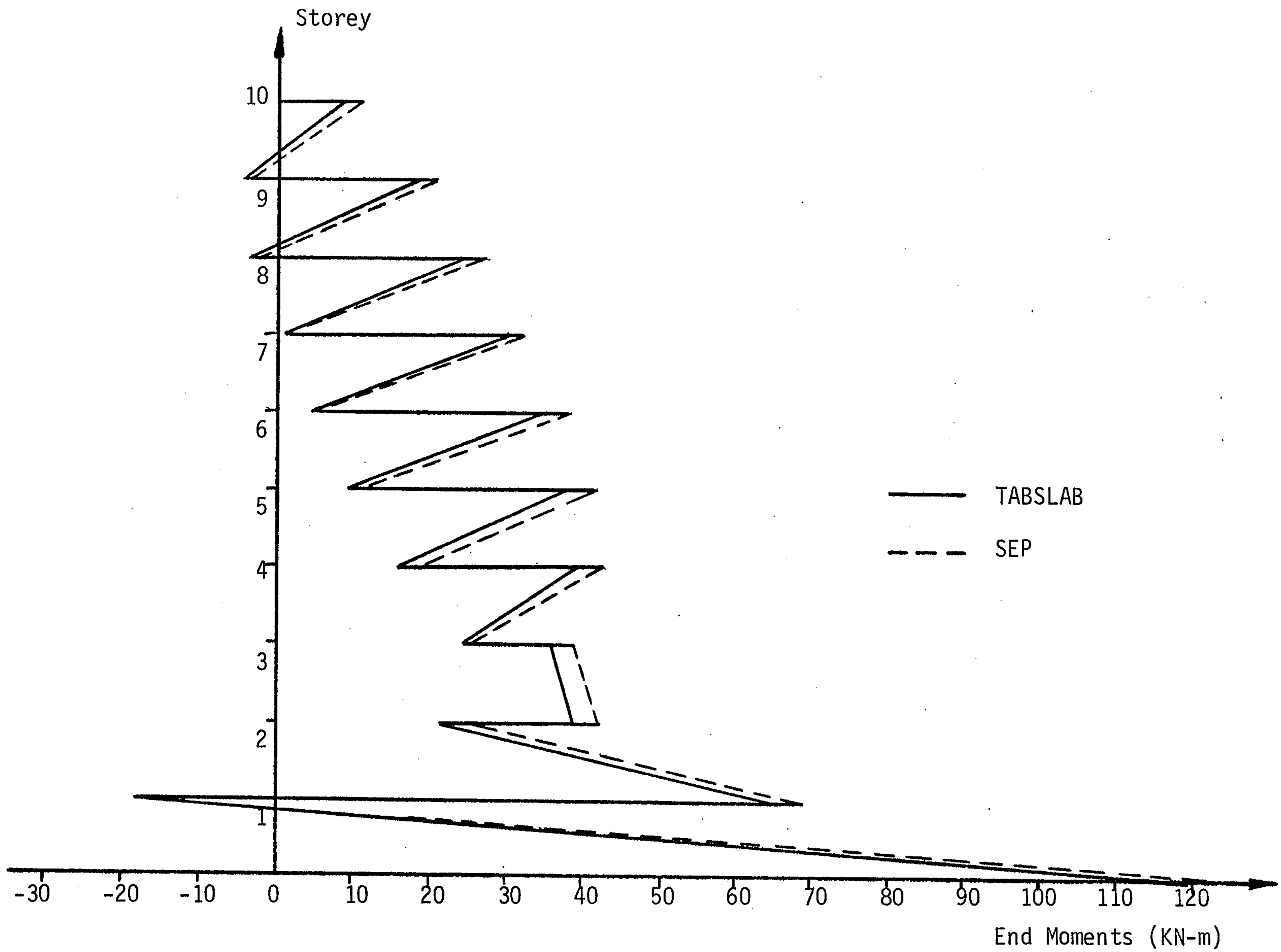


FIGURE 5.4 COLUMN 2 END MOMENTS -- EXAMPLE 2

Floor Level	COLUMN 1			COLUMN 2			COLUMN 5			COLUMN 6		
	SEP	TABSLAB	Abs. Diff.	SEP	TABSLAB	Abs. Diff.	SEP	TABSLAB	Abs. Diff.	SEP	TABSLAB	Abs. Diff.
10	-0.95	-1.24	0.29	2.25	1.56	0.69	0.46	0.78	0.32	5.32	5.97	0.65
9	2.53	2.14	0.39	5.58	4.88	0.70	4.18	4.45	0.27	8.97	9.75	0.78
8	4.39	3.81	0.58	9.26	8.12	1.14	7.13	7.6	0.47	14.65	15.83	1.18
7	6.47	5.66	0.81	12.84	11.33	1.51	10.23	10.84	0.61	20.08	21.70	1.62
6	8.46	7.43	1.03	16.45	14.54	1.91	13.28	14.05	0.77	25.61	27.65	2.04
5	10.58	9.33	1.25	20.04	17.76	2.28	16.38	17.30	0.92	30.99	33.43	2.44
4	12.95	11.52	1.43	23.61	21.04	2.57	19.57	20.62	1.05	36.01	38.81	2.8
3	16.05	14.50	1.55	27.14	24.44	2.70	23.01	24.09	1.08	40.11	43.09	2.98
2	20.40	18.95	1.45	30.64	28.12	2.52	26.86	27.87	1.01	42.60	45.34	2.74
1	29.19	28.49	0.70	33.92	32.71	1.21	32.16	32.61	0.45	39.41	40.61	1.2

TABLE 5.5 Column Shear Forces (Kilonewton)

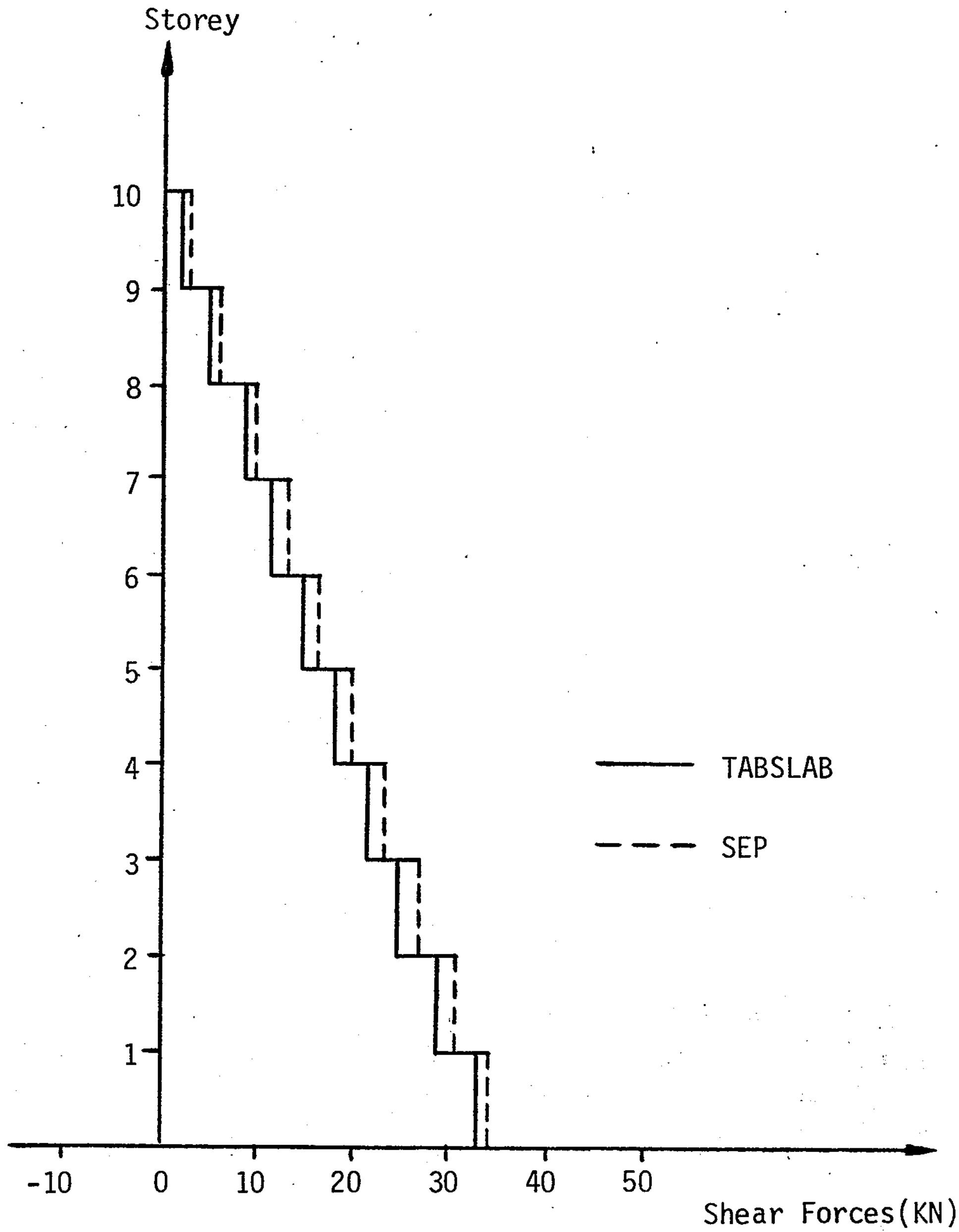


FIGURE 5.5 COLUMN 2 SHEAR FORCES -- EXAMPLE 2

the results predicted by SEP analysis. However, when the number of finite elements used to represent the panel increases, the structure appears to become more flexible. Problems regarding the flexibility and convergence of results will be discussed thoroughly in the next section.

5.4 Convergence of Results

The convergence aspect of the analysis procedure is illustrated in the two examples presented in this section.

Example 3:

The 10-storey flat plate structure in Example 2 of the previous section is analysed using different numbers of finite elements to represent each floor panel. Figure 5.6 shows the deflected shape of the structure as predicted by TABSLAB for the various cases. When n equals 1, i.e. one finite element per panel, the plot of the TABSLAB and SEP results (from Example 2) are in close agreement. When n equals 2 (2 x 2 elements per panel), the results predicted by TABSLAB indicate a more flexible structure, the lateral deflection at the top floor being about twice as large as that for the case of n equal 1 (19.78 mm versus 10.09 mm). With increasing values of n, increasing flexibility in the structure is observed. For the case n equals 8 (64 elements per panel), the lateral deflection at the top level is about 3 times that for the case where n equals 1 (29.31 mm versus 10.09 mm). Figure 5.7 shows the convergence of lateral deflections at the top level of the structure for different numbers of elements per panel. It is noted in Figure 5.7 that lateral deflection increases rapidly between the case n equals 1 and that where n equals 2. The rate of increase becomes less rapid for cases n equals 4 onward. For the case n equals 8, the curve becomes

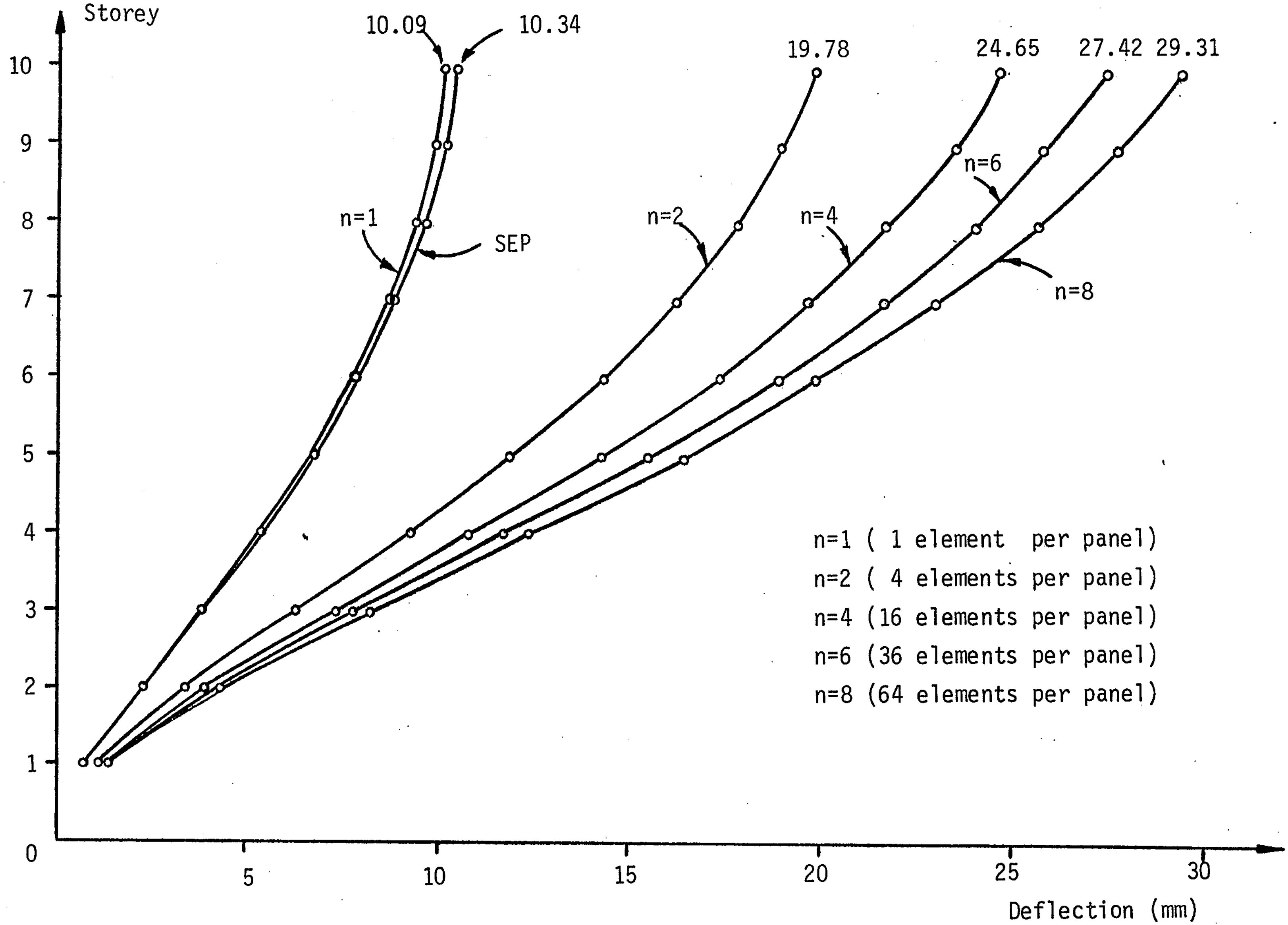


FIGURE 5.6 DEFLECTED SHAPES BY DIFFERENT NUMBERS OF FINITE ELEMENT REPRESENTATION

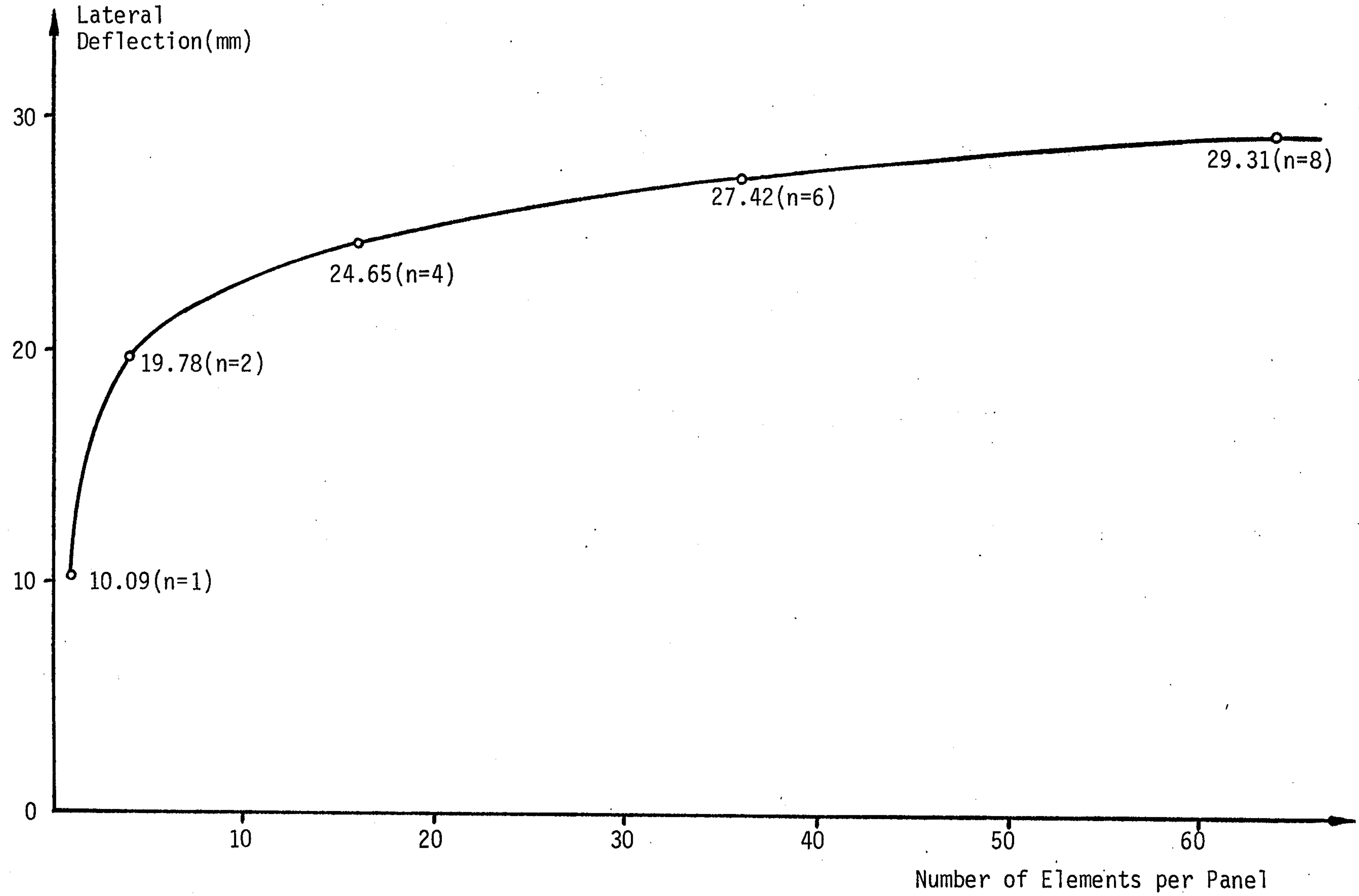


FIGURE 5.7 CONVERGENCE OF LATERAL DEFLECTION AT STOREY NUMBER 10

quite flat but it appears that further increase in the number of finite elements will still give some increase in lateral deflection.

It is of interest to know how column moments and shear forces vary for the two cases (n=1 and n=8). Since the column shear forces at each level should be in equilibrium with the applied lateral loads, which are constant, no significant variation in the shear forces in individual columns should be expected. Figure 5.8 shows shear forces for column 6 for the two cases n equals 1 and n equals 8. The curves for the two cases are quite close to each other. At the top floor level the shear force for the case n equals 1 is slightly less than that for the case n equals 8 while at the bottom floor level, the shear for the case n equals 1 is slightly larger. Figure 5.9 shows end moments for column 6 for the two cases. Unlike shear forces, the column end moments for the two cases vary significantly, especially at the lower levels of the structure. As shown in Figure 5.9, the top moment at storey 1 is -2.20 KN-m while the bottom moment is 126.06 KN-m for the case n equals 1. The corresponding results for the case n equals 8 is -83.21 KN-m and 198.68 KN-m. This wide spread in moments for the latter case represents an increase of 220 percent over the n equals 1 case, for the column at level 1, as indicated by the expression in Figure 5.9.

The information produced by SEP analysis (see Example 2) regarding plate moments is limited to the total moment transferred from the column into the plate at each junction. The SEP analysis therefore gives no information regarding the distribution of the bending moments within the floor slab. However, the TABSLAB analysis enables consideration of different numbers of elements per panel, thus it gives moment information at

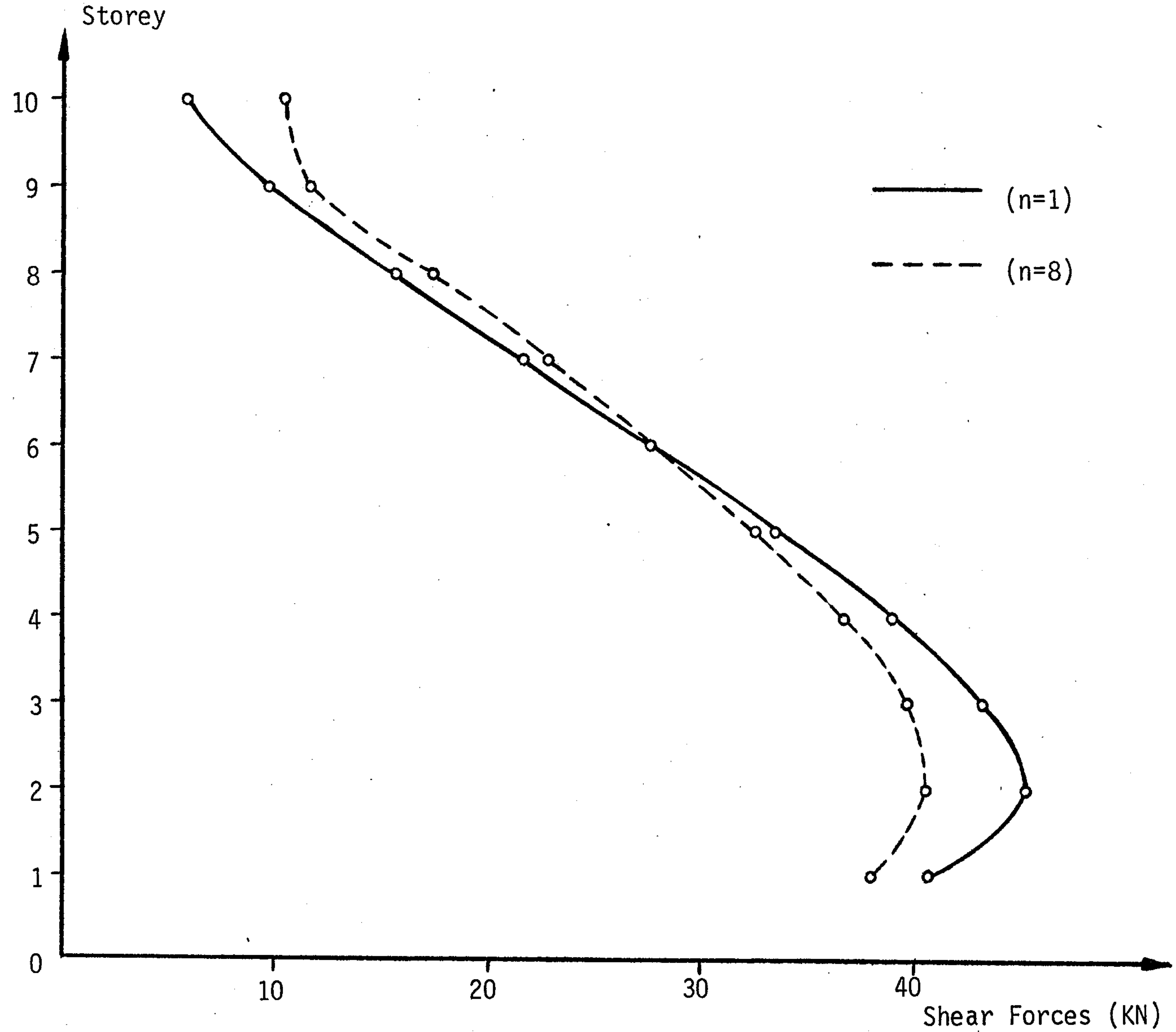


FIGURE 5.8 SHEAR FORCES FOR COLUMN 6 (n=1 and n=8)

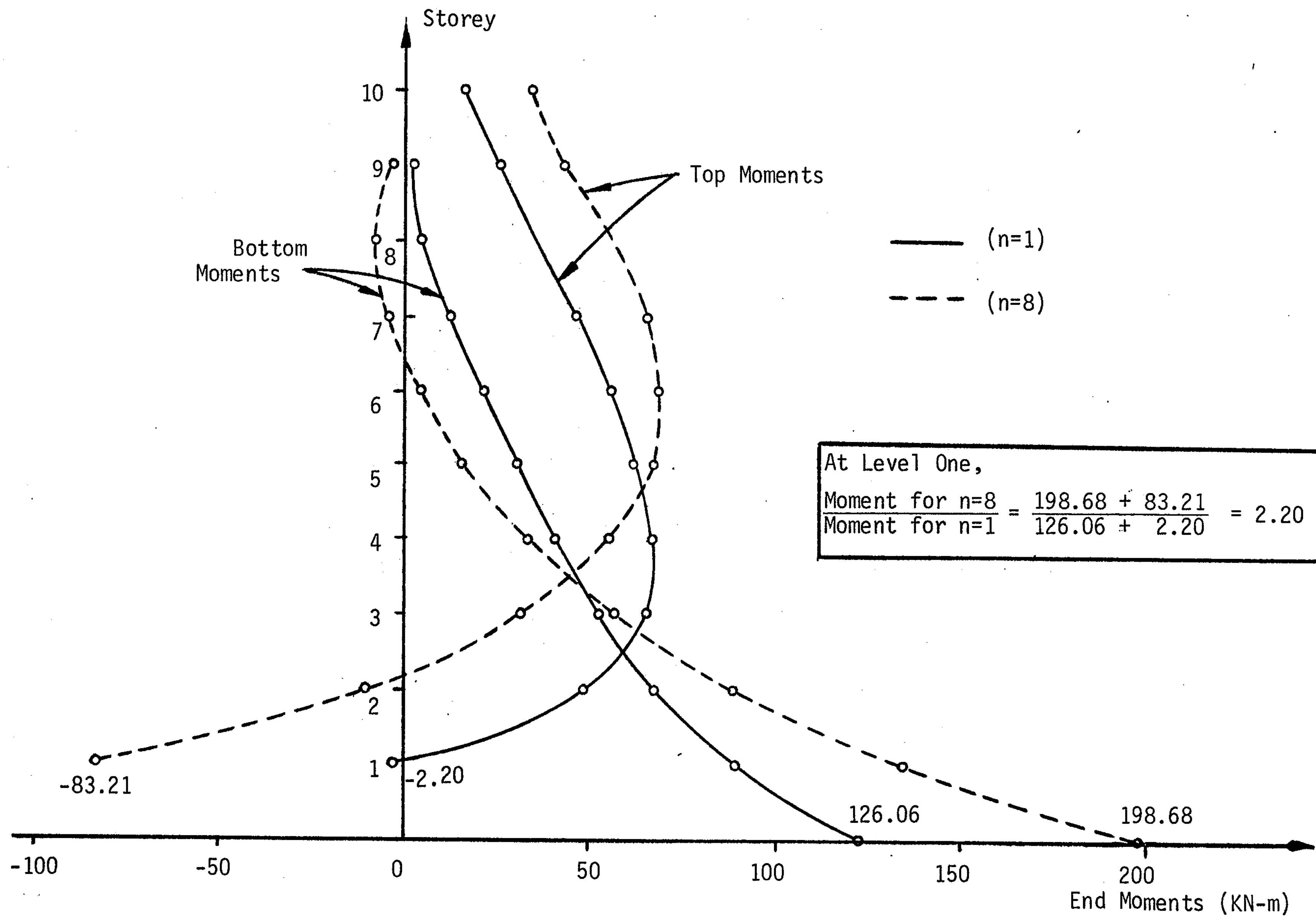


FIGURE 5.9 END MOMENTS FOR COLUMN 6 (n=1 and n=8)

any finite element nodes within the slab. Figures 5.10(a) and 5.10(b) show plots of the plate moments along section 1-1 and section 2-2 on the floor plan of Figure 5.2(b), for the case n equals 6. These two plots are given for reference purpose only, because plate moment solutions from other analyses are not available for comparison.

The very substantial difference in results yielded by the SEP analysis and the TABSLAB analysis, when more than one finite element per panel was used prompted an extensive, though not completely successful, investigation into the cause of the discrepancies.

Firstly, as illustrated in preceding Example 1, a framed structure was analysed using TABSLAB and the results were compared with those obtained using frame PFRAME. The close agreement between the results from the two analyses verified that the "frame portion" of TABSLAB is valid.

Secondly, the procedure for the assembly and condensation of the floor bending stiffness matrix was verified by two entirely different methods of stiffness matrix formation. The first method, which was described in Chapter 3, performed the assembly-condensation procedure progressively. Stiffness coefficients related to eight nodal points were assembled and condensed as a group. Cycles of assembly-condensation procedure were repeated for all nodal points on the floor, eight nodes at a time. The second method involved one complete assembly-condensation procedure for the whole floor. Identical results were obtained from both methods. Furthermore, it was found that in the first method, when different numbers of nodes were condensed as a group, identical results were obtained.

Thirdly, when the condensed floor stiffness matrices were printed for the various cases, with different numbers of finite elements per panel,

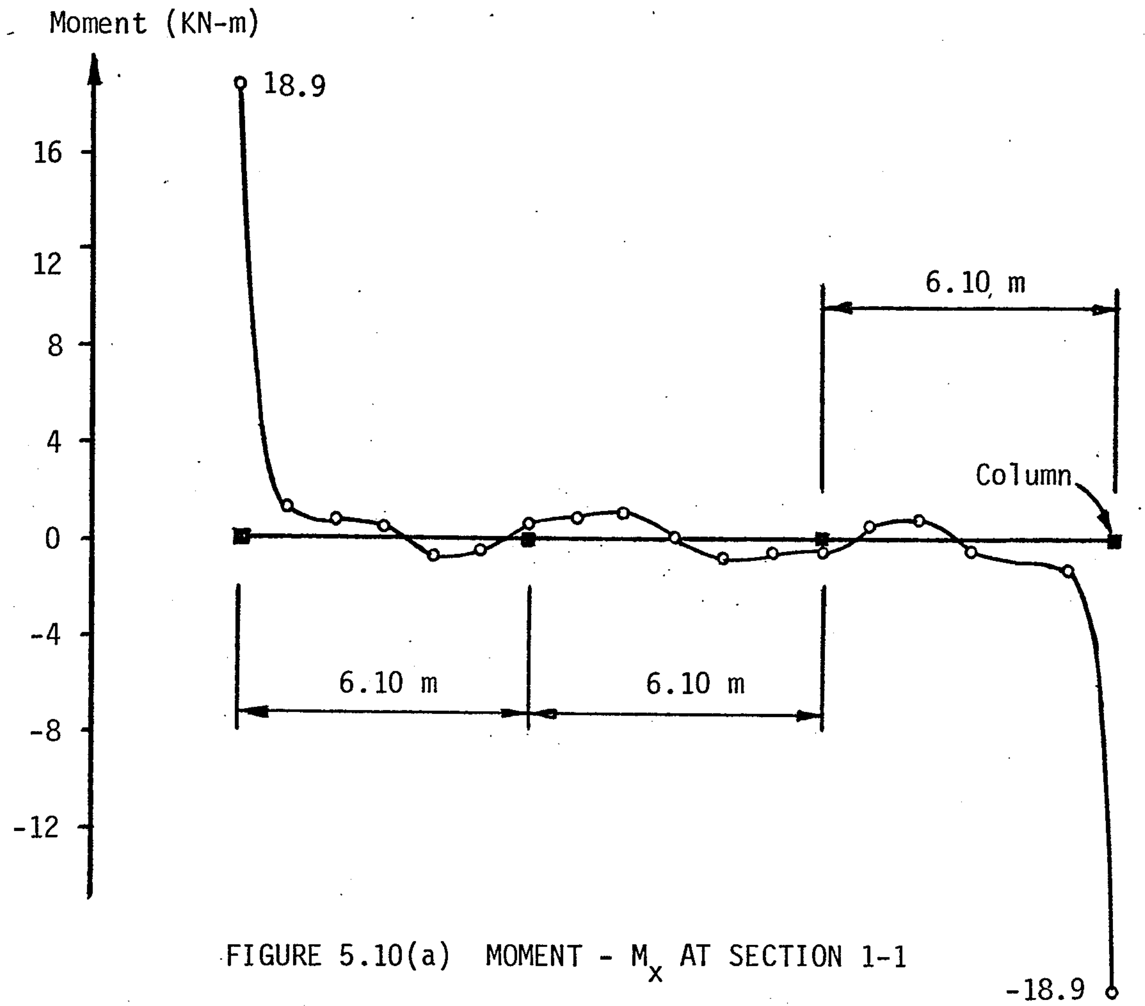


FIGURE 5.10(a) MOMENT - M_x AT SECTION 1-1

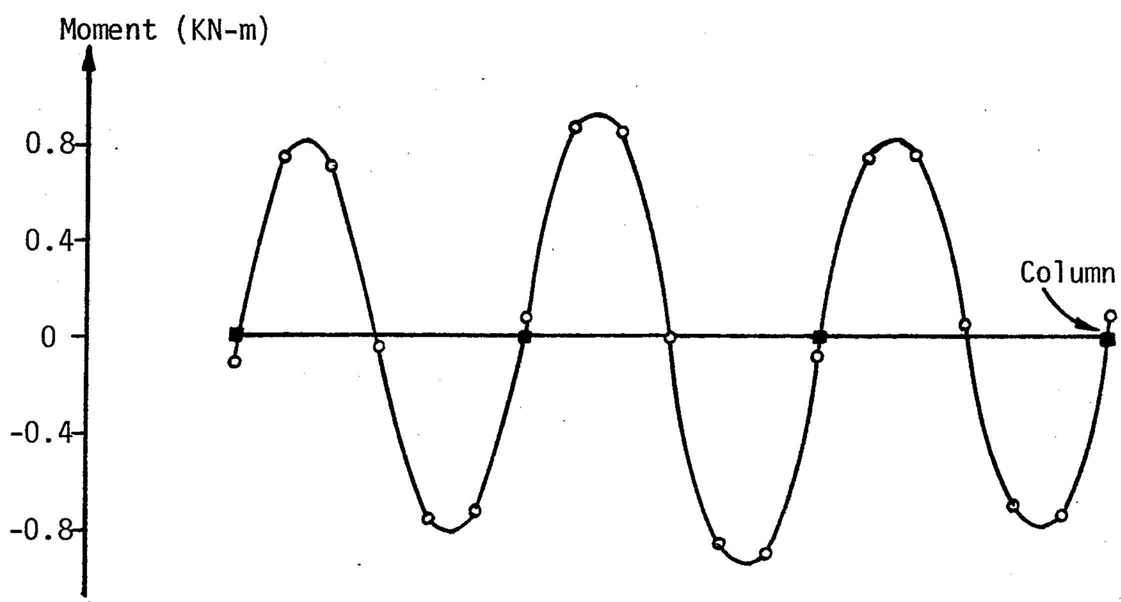


FIGURE 5.10(b) MOMENT - M_x AT SECTION 2-2

it was found that stiffness coefficients become smaller and smaller as the number of elements per panel increased. This is not difficult to visualize, because when more elements are used to model a structure, the modelled structure will become more flexible, with diminishing stiffness coefficients. Consequently, a convergence plot such as that in Figure 5.7, with increasing flexibility when the number of elements increases, is to be expected.

The plate model employed in this analysis is a non-conforming one. It was decided to compare results obtained from TABSLAB with those obtained using a general purpose structural analysis program that permitted modelling of the floor plates. Because of the relatively cumbersome input requirements and relatively large storage and computational requirements of the general purpose program (which does not employ condensation), the small, two-storey structure of Example 4 below was analysed. Several different numbers of elements per floor panel were considered and the rates of convergence studied for both programs.

Example 4:

A two-panel, two-storey flat plate building, supported by columns (equidistant) in both directions, is analysed by both TABSLAB and SAP4. Convergence of results from the two analyses is compared. Figure 5.11(a) shows the elevation of the building with lateral loading assumed acting on the frame as shown. Figure 5.11(b) shows the typical plan view of the building and the numbering scheme used for the column nodes. The flat plate floors have constant thicknesses of 0.5 ft. throughout. All columns are 1 ft. square. All material has a modulus of elasticity 576,000 kips/ft^2 and Poisson's ratio 0.2.

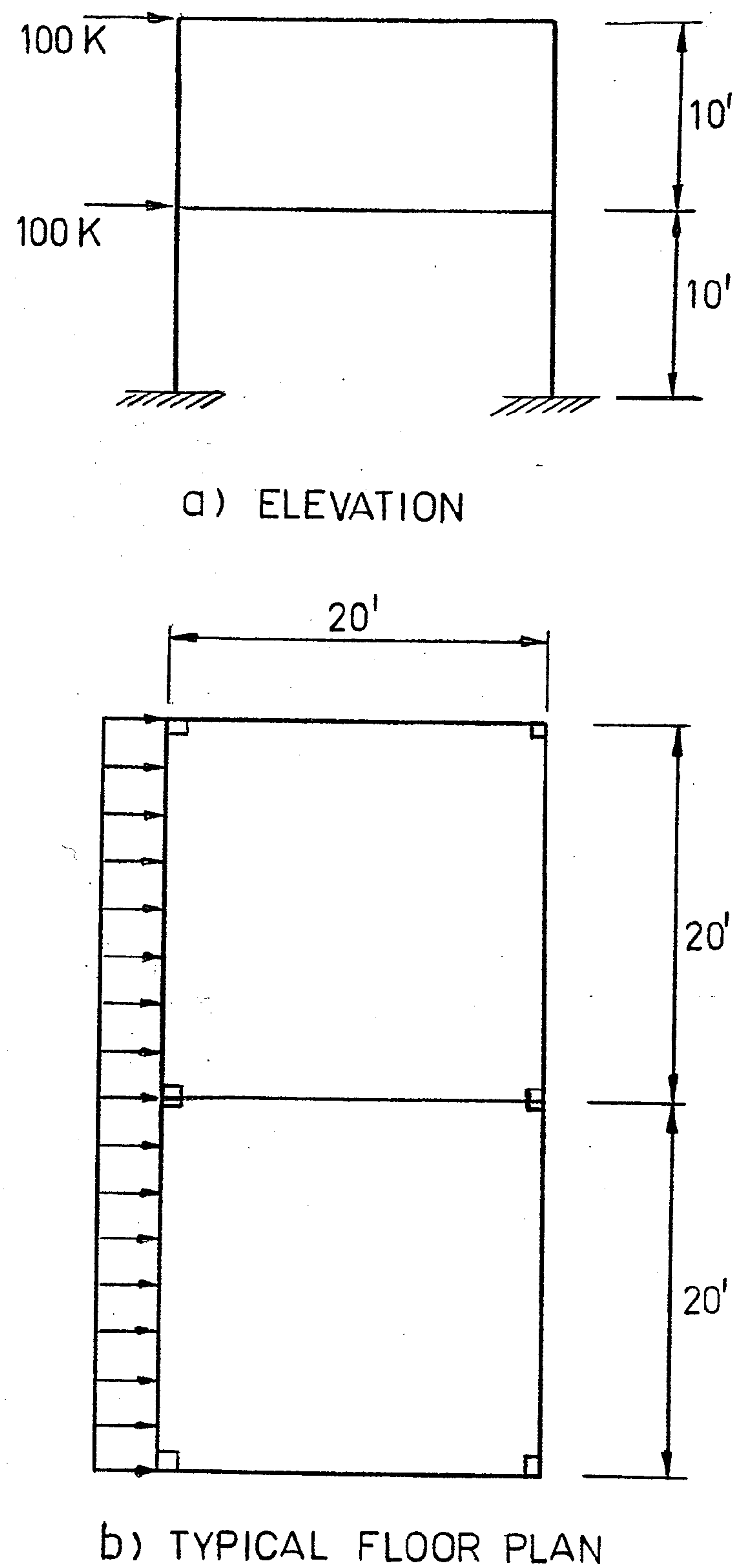


FIGURE 5.11 FLAT PLATE STRUCTURE - EXAMPLE 4

The structure is analysed using 1, 4, 16, 36, 64 and 100 finite elements per slab panel (i.e. cases n=1, 2, 4, 6, 8 and 10 respectively). Table 5.6 shows the lateral deflections as calculated by both analyses for the various cases. From the results shown, it can be seen that results predicted by SAP4 indicate a stiffer structure than do those by TABSLAB analysis. The largest difference in results is 21.7% while many are around 10%. One reason for this difference is the difference in the plate element stiffness matrices employed in the two analyses. The plate element available in the SAP4 program is a quadrilateral of arbitrary geometry formed from four compatible triangles. The element has six interior degrees of freedom which are eliminated by condensation prior to assembly. On the other hand, the plate element employed by TABSLAB analysis is a non-conforming one in which the roations of the element edges on a common boundary are not compatible. Therefore, the SAP4 representation can be expected to be stiffer, since compatibility is ensured along element boundaries.

Figure 5.12 shows the convergence of lateral displacements at the top level for both analyses. For the case n equals 10 (i.e. 100 elements per panel), the curves become quite flat but it appears that a further increase in the number of finite elements will still give some increase in lateral deflection. It is noted that the ratio of lateral deflection between the cases n equals 10 and n equals 1 is 2.16 for TABSLAB and 2.23 for SAP4. However, both analyses give a significant difference in results between the n equals 1 case and n equals 10 case.

Table 5.7(a) shows the comparison of moments for columns 1, 2, 3 and 4 in Figure 5.11(b). In Table 5.7(a), the column end moments for storey level 2 are in agreement for the two analyses, within 3.8%. While column

LEVEL		SAP4	TABSLAB	% DIFF.
2	n=1	0.172	0.192	11.6
	n=2	0.244	0.297	21.7
	n=4	0.303	0.349	11.5
	n=6	0.342	0.379	10.8
	n=8	0.369	0.400	8.4
	n=10	0.383	0.415	8.3
1	n=1	0.092	0.1	8.7
	n=2	0.118	0.136	15.2
	n=4	0.138	0.154	11.6
	n=6	0.152	0.163	7.2
	n=8	0.164	0.169	3.0
	n=10	0.171	0.174	1.8

TABLE 5.6 Comparison of Lateral Deflection (Ft.)

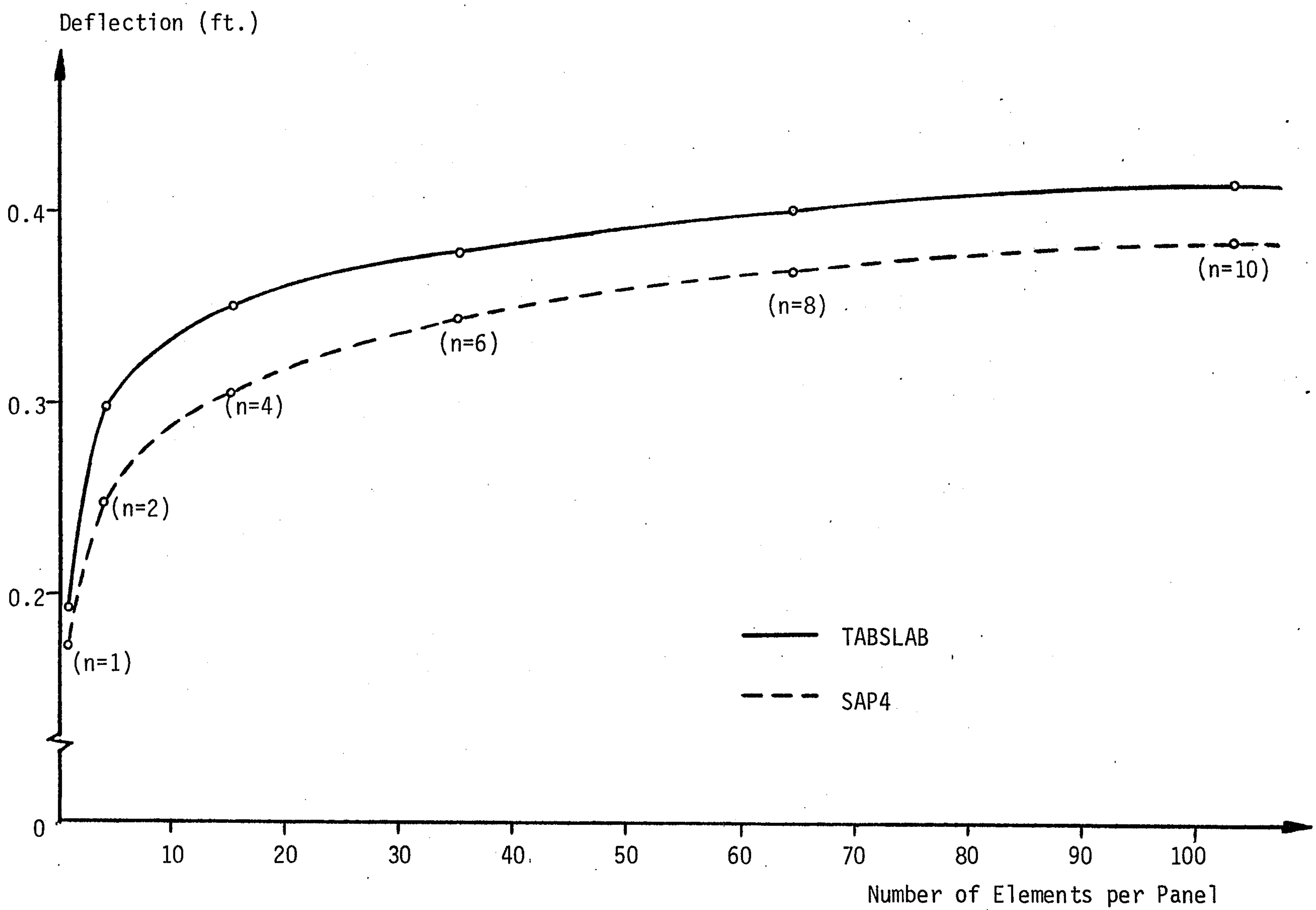


FIGURE 5.12 CONVERGENCE OF LATERAL DEFLECTION AT STOREY NUMBER 2

Floor Level		COLUMN 1			COLUMN 2			COLUMN 3			COLUMN 4		
		SEP	TAB-SLAB	% DIFF	SEP	TAB-SLAB	% DIFF	SEP	TAB-SLAB	% DIFF	SEP	TAB-SLAB	% DIFF
2	top	-84.67	-83.15	1.79	84.66	83.15	1.78	-147.2	-147.05	0.1	147.3	147.05	0.16
	btm	-40.89	-42.52	3.8	40.88	42.52	3.8	-101.4	-101.62	0.2	101.9	101.62	0.27
1	top	-76.1	-63.71	16.2	76.14	63.71	16.2	-119.6	-100.24	16.2	118.1	100.24	15.6
	btm	-235.3	-251.47	6.4	235.3	251.47	6.4	-258.5	-269.41	4.04	257.4	269.41	4.45

TABLE 5.7(a) Comparison of Colunn End Moments (n=4) Kip-Ft.

Floor Level	COLUMN 1			COLUMN 2			COLUMN 3			COLUMN 4		
	SEP	TAB-SLAB	% DIFF	SEP	TAB-SLAB	% DIFF	SEP	TAB-SLAB	% DIFF	SEP	TAB-SLAB	% DIFF
2	-12.56	-12.57	0.08	-12.55	-12.57	0.16	-24.86	-24.87	0.04	-24.92	-24.87	0.2
1	-31.14	-31.52	1.22	-31.15	-31.52	1.18	-37.81	-36.96	2.24	-37.61	-36.96	1.73

TABLE 5.7(b) Comparison of Column Shear Forces (n=4) Kip

end moments for storey level 1 vary by as much as 16.2%, this is partially due to the difference in the plate elements employed by the two analyses. Table 5.7(b) shows the comparison of column shear forces. The results predicted by the two analyses are in good agreement. The largest difference is 2.24%.

5.5 Comparison of Computer Requirements

The data preparation effort for the TABSLAB program is much less than that required by a general analysis program, such as SAP4. Example 4 is used for comparison purposes. Table 5.9(a) shows the number of data cards required for each analysis, for the various cases. SAP4 requires more data cards than TABSLAB program. The ratio varies from 2.61 for the case n equals 1 to 4.55 for the cases n equals 10. The SAP4 Program also requires much more computer time than does TABSLAB. As shown in Table 5.9(b), the ratio of central processing time for the two analyses ranges from 1.0 for n equals 1 to 13.64 for the case n equals 8. As for the core requirements, both programs require 320 K-btyes for Example 4.

No. of Elements Per Panel	TABSLAB	SAP4	RATIO
1 (n=1)	21	55	2.61
4 (n=2)	27	57	2.11
16 (n=4)	27	79	2.92
36 (n=6)	27	89	3.29
64 (n=8)	27	105	3.88
100 (n=10)	27	123	4.55

TABLE 5.9(a) DATA CARDS REQUIRED - EXMMPLE 4

No. of Elements Per Panel	TABSLAB	SAP4	RATIO
1 (n=1)	0.10	0.10	1.0
4 (n=2)	0.14	0.25	1.78
16 (n=4)	0.27	1.51	5.59
36 (n=6)	0.61	7.83	12.83
64 (n=8)	1.31	17.88	13.64
100 (n=10)	2.43	29.42	12.11

TABLE 5.9(b) Computer Central Processing Time in Minutes - EXAMPLE 4

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

A procedure has been presented and a computer program (TABSLAB) described for an efficient three-dimensional analysis of multistorey flat plate building structures subjected to lateral loadings (gravity or earthquake loading can be included if desired). The inclusion of the finite element representation of the floor slab in this analysis requires more computer time than some other structural frame analysis programs, such as TABS [4], in which no slab representation is included. However, additional computational effort is required for the generation of the loading stiffness matrix for a typical floor only. This analysis is still efficient and economical when compared to that carried out by general purpose structural analysis programs.

Four examples have been included to demonstrate the validity and accuracy of the various aspects of the analysis procedure. The first example demonstrated the capability of the TABSLAB program to analyse framed structures. The results given by TABSLAB agree within 2% with those given by PFRAME, a general frame analysis program. This verifies the validity of the "frame portion" of the TABSLAB program. The second example illustrated the lateral load analysis of a ten-storey flat plate building and the results were compared with SEP analysis [10]. When one element was used to represent each slab panel, the results predicted by TABSLAB agreed with those given by SEP, within approximately 2% for lateral deflection and 4.5% for column end moments and shear forces. The third example involved lateral load analyses of the structure of Example 2 using different numbers of finite elements to represent a floor panel. It was found that when the number of elements increase, TABSLAB predicted a more flexible structure. When 64 elements were used for the panel, the lateral deflection was about 3 times the deflection for the one-element panel case. It was found that the results converged slowly even when 64 elements (or more) per panel were used. The substantial difference in results between SEP analysis and TABSLAB analysis with more than one element per panel prompted an investigation into the cause of the discrepancies. It was found that when the number of elements per panel increased, floor stiffness coefficients diminished, and as a result, the modelled structure became less stiff. In Example 4, a two-storey flat plate building was analysed and the results were verified with SAP4 analysis, which employed a different type of plate model. Although it was found that results from SAP4 analysis are stiffer than those from TABSLAB, the results from both analyses converged almost in a parallel fashion. It was found in both analyses that, when 64 to 100 elements were used for the panel, the lateral deflection is 2.2 times the deflection for the one-element panel case.

Since no experimental results are available for comparison, it is not easy to justify which analysis will give the most realistic results. However, since TABSLAB and SAP4 give the same convergence pattern, it is reasonable to conclude that the analysis procedure, at least, is valid.

A recently available report [17] suggests that ignoring finite column dimensions when formulating slab stiffness matrix significantly affects the calculated slab stiffness. According to the report, dimensionless stiffness coefficients of the slab panel increased by an average of 64% when column size to slab size ratio increase from 1/20 to 1/8. Hence, while the analysis procedure described in this study appears to be valid, the ignoring of the transverse column dimensions in formulating slab stiffness matrix appears to lead to a significantly overly flexible mathematical model, and one whose convergence with increasing numbers of slab elements is relatively slow.

6.2 Recommendations for Further Study

Because of the lack of an "exact" analytical procedure, it is recommended that an experimental study of frame and flat plate structures be undertaken to correlate with the analytical procedure generated in this study.

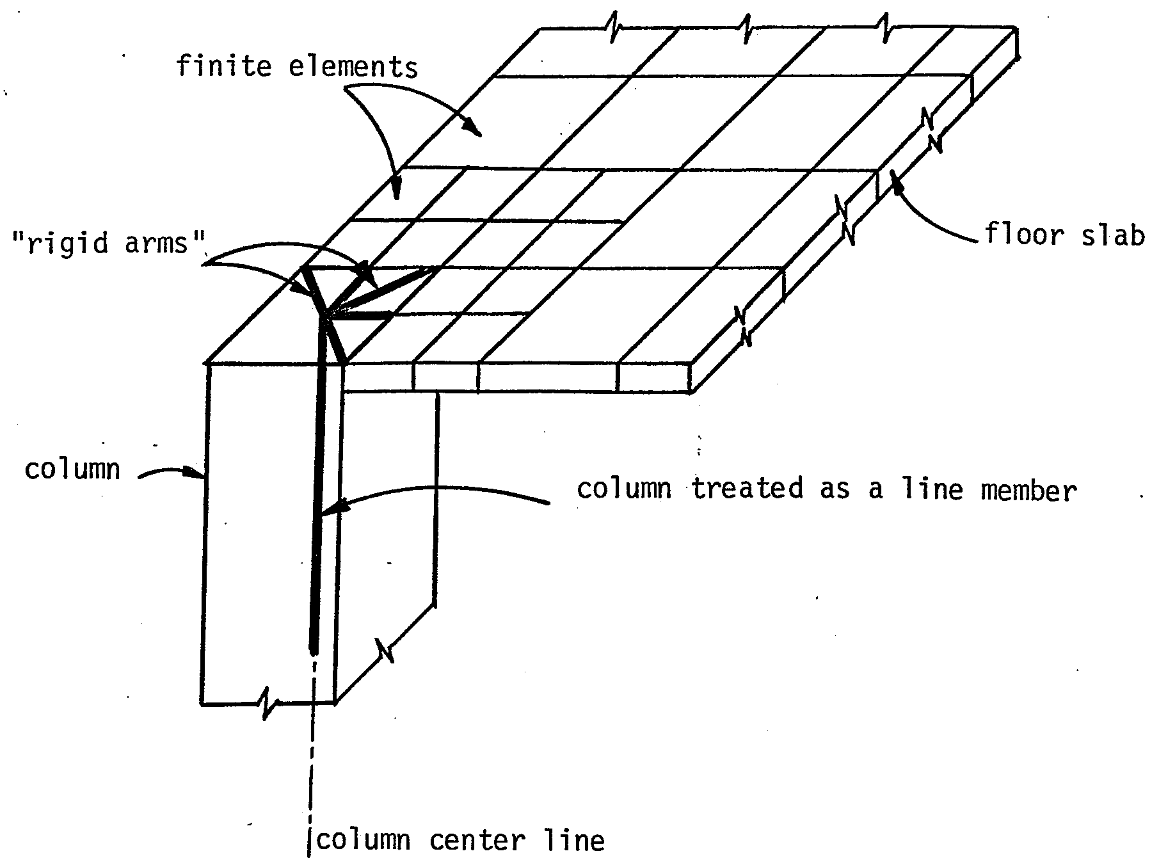
In this analysis, only rectangular slab panels were considered. In some cases, structures may not be rectangular in plan. It is therefore recommended that further study should be performed to incorporate the capability of analyzing structures with non-rectangular floor plans.

The TABSLAB program requires all floors to be identical and slab panels in a given floor to have the same properties and thickness. In many cases, floor and slab panels of particular thickness and properties are desired, such as a particular floor for a warehouse. It is therefore recommended that different floor and slab thicknesses and properties be provided for in further work.

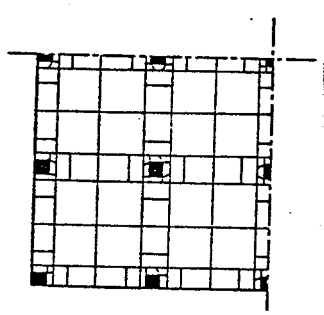
For gravity load analysis, TABSLAB requires that loading be assumed to act on beam elements only. This is a rough approximation because gravity loads such as the self weight of the slab and live load on the slab do not act directly on the beams. Therefore, the capability of analyzing structures with gravity loading on slabs should be built into the program.

The plate element employed in this analysis is a non-conforming element, i.e. compatibility of rotations is not ensured along common element boundaries. It is suggested that a higher order compatible element be incorporated. For example, a quadilateral element formed from four compatible triangles, having an internal node within the element, could be used.

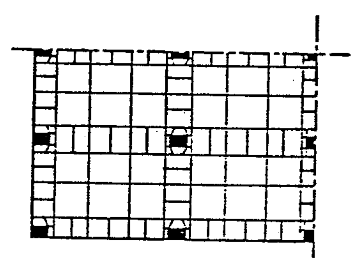
As suggested by French's report [17], size of column has a very significant effect on the slab stiffness. Further works on this area should include the finite column dimensions in the formulation of slab stiffness matrix. One way of doing this is to treat the column as essentially a line member, which is connected to the abutting elements by "rigid arm", as shown in Figure 6.1(a). To provide accurate modelling in the regions of high curvature (near the locations of columns), the mesh grading technique of Somervaille [18], such as those shown in Figure 6.1(b) and (c), can be used.



(a) Column-slab Idealization



(b) Square Plate Mesh



(c) Rectangular Plate Mesh

FIGURE 6.1 COLUMN IDEALIZATION AND SLAB PANEL GRADING

LIST OF REFERENCES

1. ACI Committee 442 Report, "Response of Buildings to Lateral Forces", ACI Journal, Proceedings, V. 68, No. 2, February 1971.

2. Wilson, E.L. and Dovey, H.H., "Static and Earthquake Analysis of Three-Dimensional Frame and Shear Wall Building", Report No. EERC 72-2, University of California, Berkeley, May 1972.

3. Weaver, W. and Nelson, M.F., "Three-Dimensional Analysis of Tier Buildings", Journal of the Structural Division, ASCE, Vol. 92, No. ST6, Proc. Paper 5019, December 1966.

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

5. Carpenter, J., "Flexural Characteristic of Flat Plate Floors in Buildings Subjected to Lateral Loads", Ph.D. Thesis of Civil Engineering, Purdue University, August 1965.

6. Goyal, U.C. and Morris, G.A., "Interaction Between Flat Plate Floors and Shear Walls", E.I.C. Transactions, Engineering Journal, December 1973.

7. Clough, R.W., King, I.P. and Wilson, E.L., "Structural Analysis of Multistorey Buildings", Journal of the Structural Division, ASCE, Vol. 90, No.ST3, Proc. Paper 9325, June 1964.

8. Viswanathan, G., "Analysis of Shear Wall-Frame Structures Subjected to Lateral Loads", M.Sc. Thesis, University of Manitoba, September 1972.

9. Nair, R. Shankaran, "Linear Structural Analysis of Multistorey Buildings", Journal of the Structural Division, ASCE, Vol. 101, No. ST3, Proc. Paper 11163, March 1975.

10. French, S., Kabaila, A.P. and Pulmano, V.A., "Single Element Panel for Flat Plate Structures", Journal of the Structural Division, ASCE, Vol. 101, No. ST9, Proc. Paper 11553, September 1975.

11. Pulmano, V.A., Black, D.C. and Kabaila, A.P., "Substructure Analysis of Multistorey Flat Slab Buildings", Proceedings of the ASCE-IABSE Regional Conference on Tall Buildings, Bangkok, Thailand, 1974.

12. Przemieniecki, J.S., "Theory of Matrix Structural Analysis", McGraw-Hill Book Company, 1968.

13. Zienkiewicz, O.C., "The Finite Element Method in Engineering Science", McGraw-Hill Book Company, 1971.

14. Morris, G.A., "PFRAME: Linear Analysis of Planar Frames", Civil Engineering Program Library, University of Manitoba, June 1972.

15. Bathe, K.J., Wilson, E.L. and Peterson, F.E., "A Structural Analysis Program for Static and Dynamic Response of Linear Systems", Report No. EERC 73-11, University of California, Berkeley, June 1973.

16. Fox, L., "An Introduction to Numerical Linear Algebra", Oxford University Press, 1965.

17. French, S., Black, D.C., Pulmano, V.A. and Kabaila, A.P., "Panel Method for Multistorey Flat Plate Structures", International Association for Bridge and Structural Engineering, Volume 35-II, pp. 65-82, Zurich, 1975.

18. Somervaille, I.J., "A General Procedure for Grading the Mesh in Arrays of Finite Elements", UNICIV Report No. R-85, The University of New South Wales, Kensington, Australia.

A P P E N D I X   A

USER'S MANUAL

A.1 Program Identification

TABSLAB - This program performs a linear elastic analysis of a three-dimensional building frame with flat plate floors. It performs either a static analysis or a dynamic analysis or a combination of both static and dynamic analyses. Up to two lateral loading cases, designated A and B and three gravity loading cases, designated I, II and III, are permitted.

A.2 Numerical Definition of the Structure

In this analysis, the overall structure is considered as a single three-dimensional frame. Figure A.1 shows the plan view and three-dimensional view of a typical shear wall-frame structure. Each typical floor is modelled as series of slab panels with columns (or dummy columns) at each panel corner. A global coordinate system, whose origin is conveniently assumed at one corner of the structure, as illustrated in Figure A.1, is used in describing the structural geometry. For maximum computing efficiency, the shorter direction of the structure should be alligned parallel to the global X axis. Floor levels and column centerlines are taken as basic reference lines in the frame description. All lateral loads are assumed to be applied at floor level.

Each slab panel, in turn, is modelled as a rectangular finite element array. Shear walls are considered as series of vertical wall panels connected together at floor levels as shown in Figure A.1(b). Since plane transverse sections of the shear wall are assumed to remain plane after deformation, stiff beams (beams with large flexural inertia) may be added to each floor level between wall panels to give a better representation of the shear wall. Both beams and columns may be omitted from any location by specifying zero member type for the member. Bending stiffnesses of beams

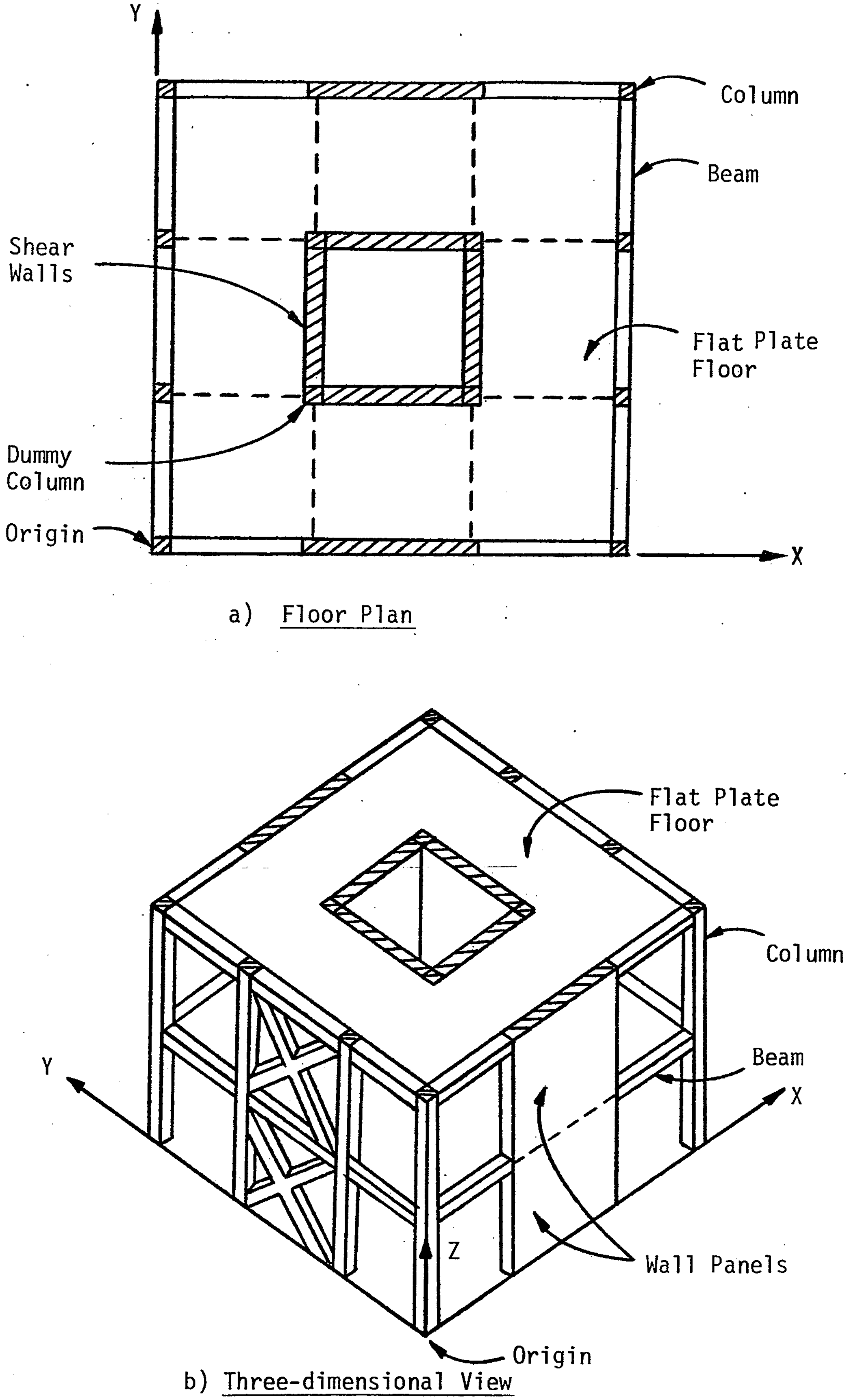


FIGURE A1 TYPICAL SHEAR WALL - FRAME BUILDING may be neglected also. Beam cross-sectional areas do not enter the stiffness calculations because the floor is assumed to be rigid in its own plane.

Deformations within joints are neglected. The effective length of a beam is reduced by the rigid zone specified on either end. The height of a column is reduced by the rigid end zones specified on either end of the column and is assumed identical about both axes. Columns must be prismatic; however, both shearing and axial deformations are included. Beams need not be prismatic but must be symmetrical about their vertical mid-planes.

A.3 Input Data

For convenience, the input data description is subdivided into two parts; Part I, which deals with statical load analysis only (the mode in which it would be most often used) and Part II, which deals with dynamic analysis. To avoid duplications, the statical analysis input preparation is fully described in Part I, while only the additional input needed for the dynamic analysis is described in Part II. The corresponding data cards are marked with an asterisk (*) in Part I. Each data card is divided into eight 10-column fields unless otherwise indicated. Information may appear anywhere within a field.

PART I - STATICAL LOAD ANALYSIS

I JOB TITLE AND CONTROL INFORMATION CARDS

FIRST CARD

Field

1-8 Job title or building identification information to be printed with the output.

SECOND CARD*

Field

1 Total number of storeys.

2 Number of loading Combinations. These are combinations of the five basic loading cases defined on LOADING COMBINATION CARDS.

3 Number of storeys where slab forces (internal moments) are required.

4 Number of columns (dummy columns included) on a typical floor.

5 Allowable storey degrees of freedom:

blank ; X, Y translation + storey rotations

X ; X translation only) for symmetrical buildings
Y ; Y translation only)

6 Plate element type for slab:

blank : Non-conforming element

1 : Conforming element

THIRD CARD

Field

1 Number of slab panels in X direction (maximum = 16).

2 Number of slab panels in Y direction (maximum = 16).

3 Total number of slab panels in a typical floor.

4 Number of different slab panel sizes (slab panel with different dimensions).

5 Number of boundary lines common to two panels on a given floor.

6 Number of "interior" columns, i.e. columns with four slab panels framing into them.

7 Number of finite element divisions per slab panel. If blank, it is assumed to be 4 (i.e. 4 x 4 finite element divisions).

8 Number of joint stiffnesses to be assembled and condensed per cycle. If blank, it is assumed to be 8. This field is normally left blank.

II STORY DATA AND LATERAL LOADING INFORMATION

Prepare two (2) cards per storey level; data are entered in sequence from top to bottom of structure.

FIRST CARD*

Field

1 Storey number.

2 Storey height.

SECOND CARD

Field

Lateral Load Case A

1 Concentrated load in x-direction, Fx

2 Concentrated load in y-direction, Fy

3 X-ordinate of point of load application, x

4 Y-ordinate of point of load application, y

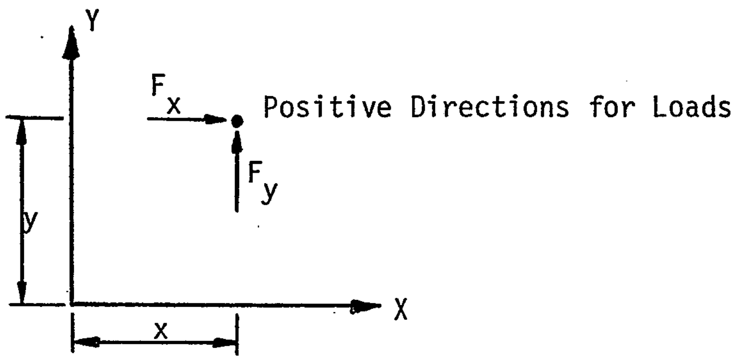
Lateral Load Case B

5 Concentrated load in x-direction, Fx

6 Concentrated load in y-direction, Fy

7 X-coordinate of point of load application, x

8 Y-coordinate of point of load application, y



THIRD CARD (used only if information for floor(s) below to be automatically generated, same as for floor considered)

Field

1 The letter G (for Generate)

2 Number of storeys in sequence below to be generated with the same properties and lateral loading information as this storey.

III TYPICAL FLOOR INFORMATION

a. Slab Panel Dimension Cards

x-direction

Field

1 Length of first panel, x_1

2 Length of second panel, x_2

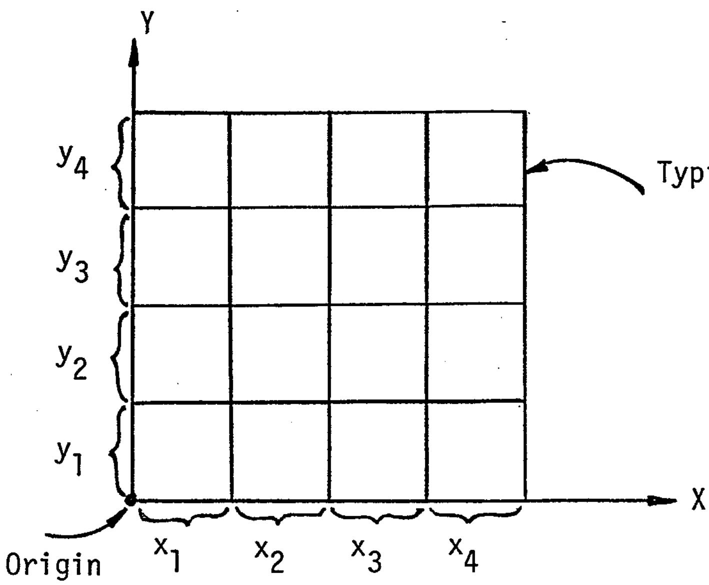
3 Length of third panel, x_3

. .

. .

. .

Enter as many numbers as required by the number of panels present. Up to 16 panels (2 cards) in the x-direction permitted.



Y-direction

Field

1 Length of first panel, y_1

2 Length of second panel, y_2

3 Length of third panel, y_3

. .

. .

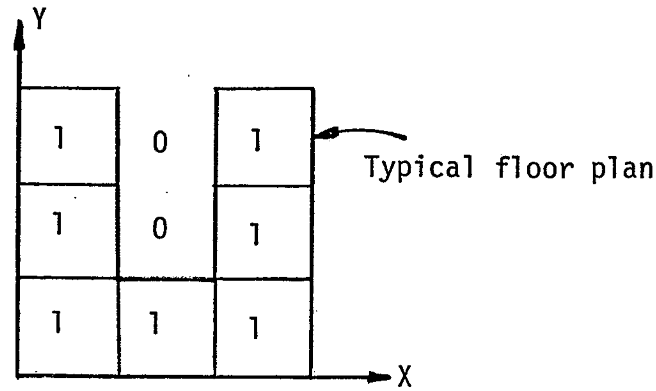
. .

Enter as many numbers as required by the number of panels present. Up to 16 panels (2 cards) in the y-direction permitted.

b. Slab Panel Location Cards

Enter 1 (one) for existing panel and 0 (zero) for missing panel starting from the origin in the sequence of increasing x-direction and then increasing y-direction. Eight numbers per card. Use as many cards as required.

Example:



1	1	1	1	0	1	1	0
1							

c. Slab Properties Card

All slabs must have same thickness, modulus of elasticity and Poison's ratio.

Field

1 Slab thickness

2 Modulus of elasticity

3 Poison's ratio

IV FRAME DATA

The complete structure is considered as a single three-dimensional frame. Input data include properties, locations and vertical loading information, for all structural elements other than the floor slabs.

a. Frame Control Information Card

Field

1 Number of bays where beam elements occur. If blank, flat plate with no beam, assumed.

2 Number of column types, i.e. columns with different sectional or material properties.

3 Number of beam types, i.e. beams with different sectional or material properties.

4 Number of sets of different fixed end moments and shears to be applied as vertical loadings to beams (girders). If blank, no vertical loading assumed.

5 Number of bays where wall panels (shear walls) occur. If blank, no wall panel assumed.

6 Number of wall panel types, i.e. wall panels with different sectional or material properties. If blank, no wall panel assumed.

7 Number of bracing elements in the building. If blank, no bracing elements assumed.

b. Column Properties Cards

Two (2) cards must be supplied for each different column type.

FIRST CARD

Field

1 Identification number for this column type (must be numbered sequentially).

2 Modulus of elasticity.

3 Cross-sectional area.

4 Shear area associated with shear forces in Y direction. If blank, shearing deformation ignored.

5 Shear area associated with shear forces in X direction. If blank, shearing deformation ignored.

6 Torsional moment of inertia.

7 Flexural moment of inertia for bending about X axis.

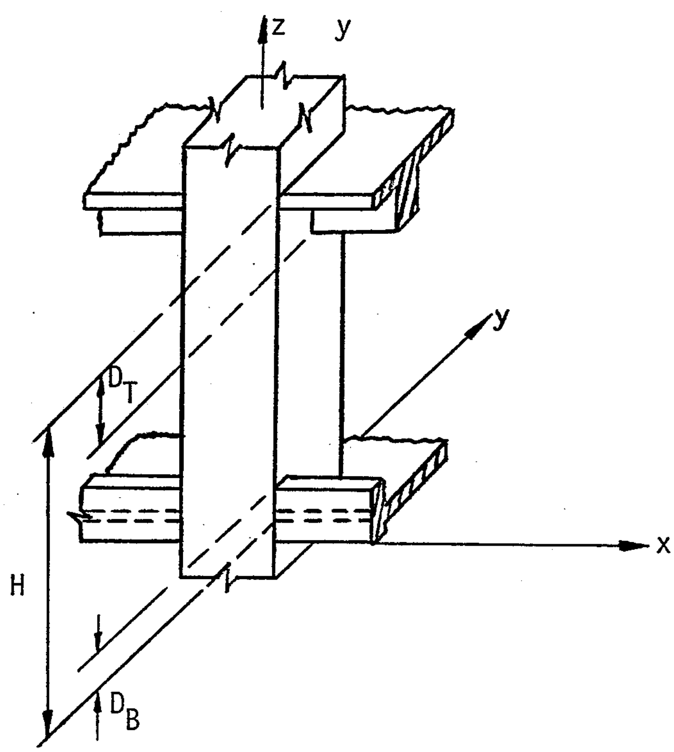
8 Flexural moment of inertia for bending about Y axis.

SECOND CARD

Field

1 Depth of finite joint at top of column, DT (for both axes). Finite joint reduces effective length of column about both axes.

2 Depth of finite joint at bottom of column, DB. Usually zero unless beam extends above floor level.



c. Beam Properties Cards

Two (2) cards must be supplied for each different beam type. Omit these cards if no beam elements in the structure.

FIRST CARD

Field

1 Identification number for this beam type (must be numbered sequentially).

2 Modulus of elasticity.

3 Shear area. If blank, shearing deformation ignored.

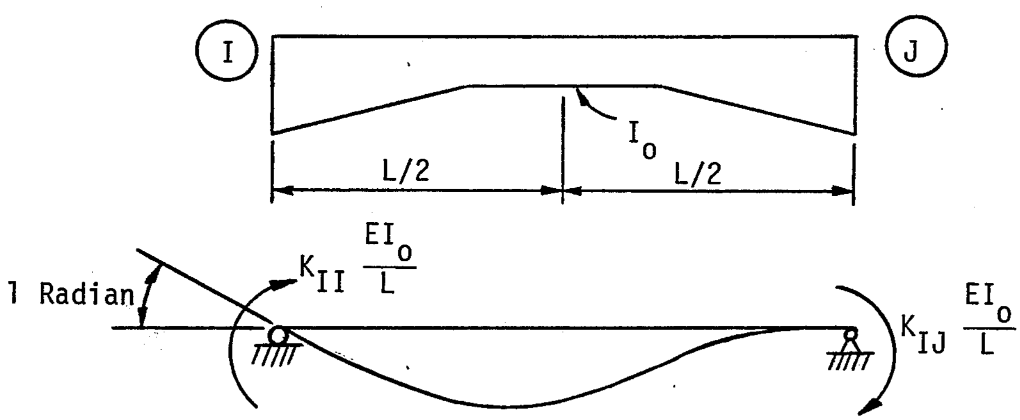
4 Torsional moment of inertia.

5 Flexural moment of inertia.

6 Stiffness factor K_{II}. If blank, it is assumed to be 4.

7 Stiffness factor K_{JJ}. If blank, it is assumed to be 4.

8 Stiffness factor K_{IJ}. If blank, it is assumed to be 2.



SECOND CARD

Field

1 Width of finite joint at end I, W_I

2 Width of finite joint at end J, W_J

d. Wall Panel Properties Cards

Prepare one card for each different wall type. Omit these cards if no wall panels in the structure.

Field

1 Identification number for this wall panel type (must be numbered sequentially).

2 Modulus of elasticity.

3 Cross-sectional area.

4 Moment of inertia. If blank, pure shear deformation panel model selected.

5 Effective shear area.

6 Shearing modulus.

e. Beam Gravity Loading Information Cards

Prepare one card for each different beam gravity loading type. Omit these cards if there is no gravity loading.

Field

1 Identification number for this beam gravity loading type. (must be numbered sequentially).

2 Input Code:

Blank ; Fixed-end-forces are applied at column faces.

C : Fixed-end-forces are applied at column center lines.

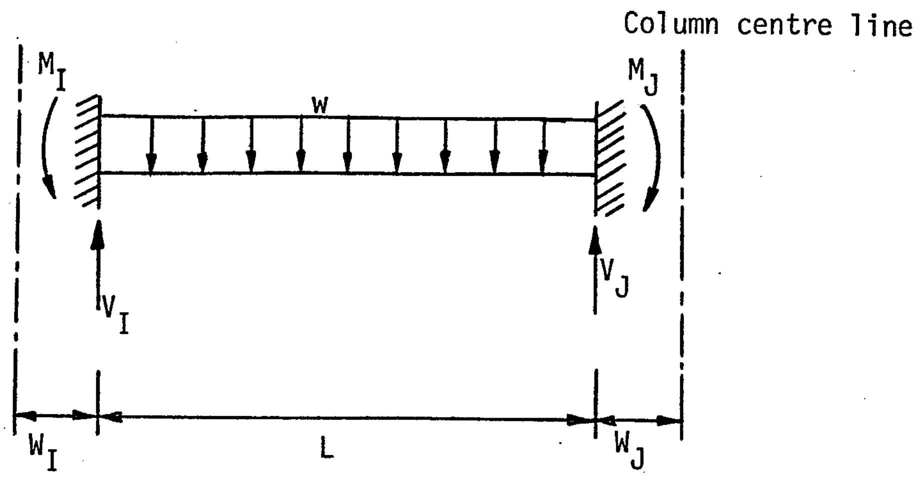
3 Fixed-end reaction, M_I

4 Fixed-end reaction, V_I

5 Fixed-end reaction, M_J

6 Fixed-end reaction, V_J

7 Uniform force per unit length, w, acting downward to be added to fixed-end reactions.



f. Beam Location Cards

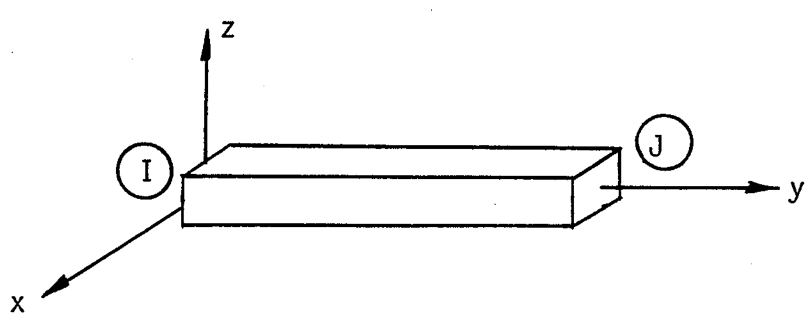
One card per beam (girder) must be input from top to bottom from bay to bay in the structure (unless the data generation option is used).

Field

1 Bay identification number for this beam.

2 Column number at end I.

3 Column number at end J. Position of I and J ends defines local co-ordinate axis with local "y" positives from I to J and local "z" positive vertically upward. A right-hand screw rule sign convention applies.



4 Beam type for this girder. If blank, no beam assumed.

5 Number of beams in sequence below to be generated with the same properties and gravity loading as this beam.

6 Gravity loading set identification number for gravity loading case I. Three independent gravity loading distributions (I, II, III) are allowed, and these distributions are combined with the lateral load case (A and B) to form load case combination for the structure.

7 Gravity loading set identification number for gravity loading case II.

8 Gravity loading set identification number for gravity loading case III.

g. Column Location Cards

One card per column must be input from top to bottom and from column to column of the structure (unless the data generation option is used).

Field

1 Column identification number of this column.

2 Column type for this column. If blank, missing column assumed.

3 Number of columns in sequence below to be generated having the same properties as this column.

h. Wall Panel Location Card

One card per wall panel must be input from top to bottom and from bay to bay of the structure (unless the data generation option is used).

Field

1 Bay identification number for this wall panel.

2 Column number at end I.

3 Column number at end J.

4 Wall panel type for this panel. If blank, missing wall panel assumed.

5 Number of panels in sequence below to be generated having the same properties as this panel.

i. Bracing Element Cards

Enter one card per brace in order; no generation is allowed.

Field

1 Level identification number at the top of this brace.

2 Column number at upper end of this brace.

3 Column number at lower end of this brace.

4 Modulus of elasticity.

5 Cross-sectional area.

V LOADING COMBINATION CARDS*

Load case for the complete building are defined as combinations of gravity loading cases (I, II, III) and lateral loading cases (A and B). One card must be entered in this section for each different combination. The total number of loading combinations is controlled by the entry in field 2 of the second CONTROL INFORMATION CARD in section I.

Field

1 Multiplier for gravity loading case I.

2 Multiplier for gravity loading case II.

3 Multiplier for gravity loading case III.

4 Multiplier for lateral loading case A.

5 Multiplier for lateral loading case B.

VI LOCATIONS WHERE SLAB FORCES REQUIRED

One set of cards must be supplied for each storey where slab forces are required, from bottom floor to the top floor. The number of sets of cards is controlled by the entry in field 3 of the second CONTROL INFORMATION CARD, in section I. Omit this section if no slab forces required.

a. Control Information Card

Field

1 Level number where slab forces are required.

2 Number of slab lines along X direction where slab forces are required. (maximum = 16).

3 Number of slab lines along Y direction where slab forces are required.

b. Slab Line Card - X Direction

Field

1 Y ordinate of first slab line.

2 Y ordinate of second slab line.

3 Y ordinate of third slab line.

. .

. .

. .

Enter as many data as required. The number of data is controlled by the entry in field 2 of the first card in this section. Up to 16 data permitted (2 cards).

c. Slab Line Card - Y Direction

Field

1 X ordinate of first slab line.

2 X ordinate of second slab line

3 X ordinate of third slab line

. .

. .

. .

Enter as many data as required. The number of data is controlled in field 3 of the first card in this section. Up to 16 data permitted (2 cards).

PART II - DYNAMIC ANALYSIS

The center of mass for each storey level must be calculated and supplied by the user. Structure displacements for dynamic analyses are calculated at the location of the center of mass. The input preparation for static and dynamic analyses is mostly similar, therefore, only the additional input needed for the dynamic analysis are described in Part II. Thus in preparing input data for dynamic analysis the user is required to refer to the statical load analysis manual as given in Part I of this section.

The following are data cards with additional input. They are cards with asterisk (*) in Part I - Statical Load Analysis.

SECTION I, SECOND CARD

Field

7 Analysis type code:

- Eq. 1 : Mode shapes and frequencies analysis only.
- Eq. 2 : Static load analysis plus mode shape and frequencies analysis.
- Eq. 3 : Lateral earthquake spectrum in addition to analysis type 2, above.
- Eq. 4 : Lateral earthquake response in addition to analysis type 2, above.

8 Number of frequencies to be calculated. It must be less than the number of storey times the number of degrees of freedom per storey.

SECTION II, FIRST CARD

Field

3 Translational mass. It has units of force divided by acceleration.

4 Rotation mass moment of inertia about a vertical axis through the center of mass. If the allowable storey degrees of freedom do not include rotation, omit this entry.

5 X-distance to the center of mass measured from the origin of the global coordinate system.

6 Y-distance to the center of mass measured from the origin of the global coordinate system.

7 External storey stiffness in the x-direction. It acts on line through the center of mass.

8 External storey stiffness in the y-direction.

The following cards are to be added after SECTION IV and before SECTION V in Part I - Statical Load Analysis.

A EARTHQUAKE ACCELERATION SPECTRUM CARDS

These data cards are required only if analysis type code was set equal to three (3), see Section I, above.

a. Control Card

Field

1 Total number of period cards used to define acceleration spectrum.

2 The number of modes, in sequence, starting with the lowest, to be printed separately. Do not exceed the number of frequencies specified on CONTROL INFORMATION CARD.

3 Acceleration, units/sec/sec.

4 Direction of earthquake input, in degrees and decimals.

b. Period Cards

Field

1 Period entered in increasing sequence.

2 Spectrum acceleration.

B TIME HISTORY CARDS

These data cards are required only if the analysis type code was set equal to four (4), see Section I, second card.

a. Control Card

Field

1 Number of acceleration cards.

2 Number of time steps to be used in the analysis.

3 Scale factor for accelerations.

4 Direction of earthquake input, angle between the global Y direction and the line of action of the earthquake direction.

5 Time increment Δt, for print of results (see Field 2 above).

b. Damping Cards

One card must be supplied for each frequency in the analysis.

Field

1 Mode number (in ascending order).

2 Damping ratio: Modal Damping/Critical Damping.

c. Acceleration Cards

One card must be supplied for each time point, at which ground acceleration is specified, in increasing time order. The time span must be greater than the number of time steps times Δt.

Field

1 Time.

2 Ground acceleration.

The following is card with additional input. This card is with asterisk (*) in Part I.

SECTION V

Field

6 Multiplier for spectrum - 1 loading

7 Multiplier for spectrum - 2 loading

8 Multiplier for earthquake response

A.4 Output From the Program

In addition to a print-out of the input data, the following is output.

a) Storey displacements for load cases I, II, III, A and B.

b) Structural mode shapes and periods (for dynamic and earthquake analysis only).

c) Frame displacements for each loading combination.

d) Member forces for each loading combination.

Positive sign convention of all members is shown in Figure A2.

For interpretation of output, all static load conditions must satisfy statics. Equilibrium should be checked at selected joints taking into consideration the effects of finite member sizes and rigid in plane floor diaphragms. Results from a dynamic spectrum analysis will not satisfy statics since the method involves summation of absolute values of individual analysis.

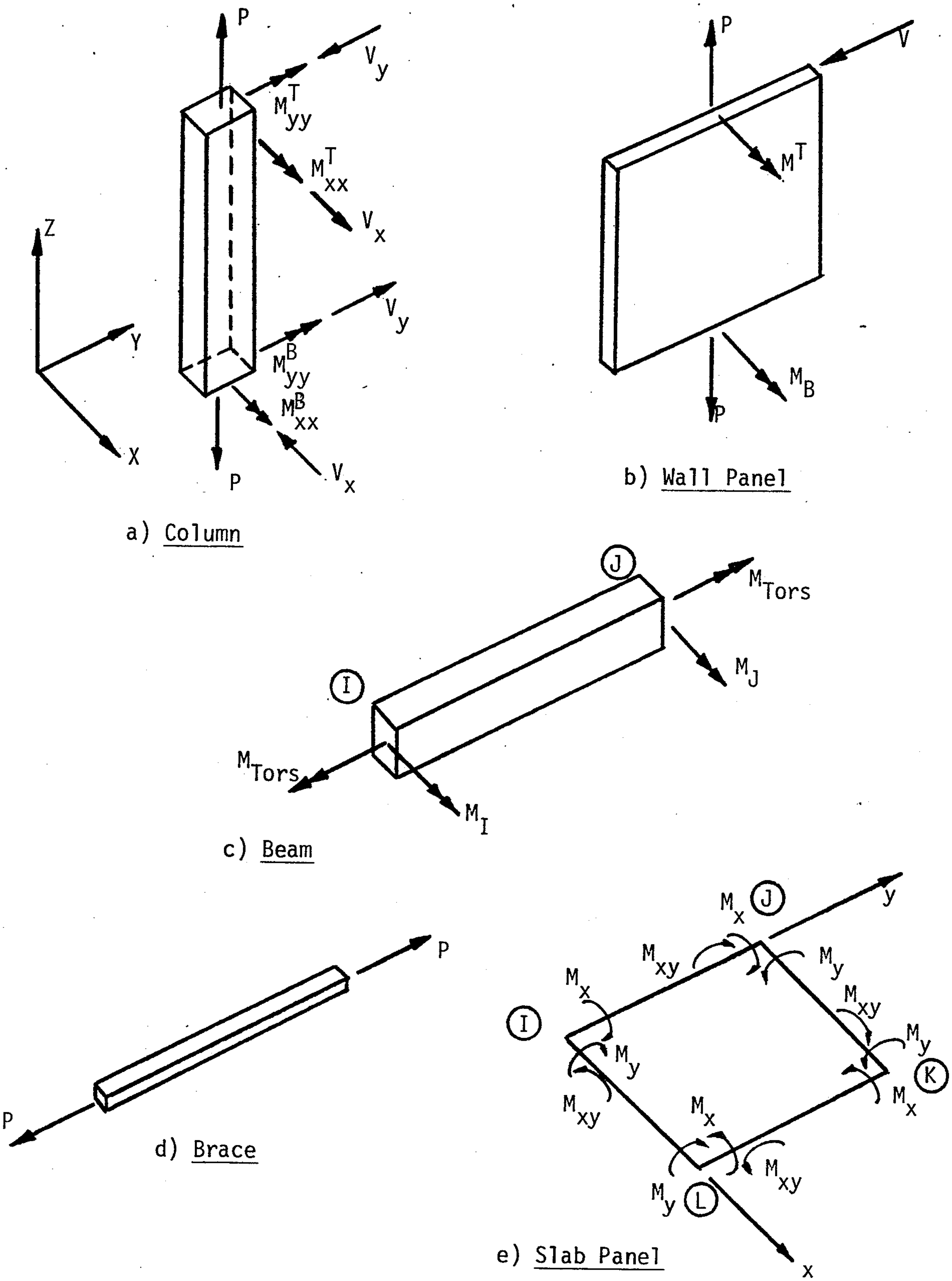


FIGURE A2 POSITIVE SIGN CONVENTION FOR MEMBER FORCES

A.5 Program Capacity

The program is written in FORTRAN IV with dynamic storage allocation for major arrays in blank COMMON. Thus the amount of high speed storage required for a particular problem may be changed by altering the following two cards at the beginning of the main program:

```
COMMON A(n)
MTOT = n
```

For a given building the value of n required is computed as follows:

1. For the floor analysis, calculate

```
NT1 = 3*NJ*(3*NJ+1)/2
NT2 = NSP*(NN+1)**2-NJ-NCS*(NN+1)+NIC
NT3 = (M*NN+1)*(N*NN+1)+M*N+2*NJ+5*NSP+(M+1)*(N+1)+146*NDP+18*NST
NODE = MAXO(NDM,M+1)
NT4 = 3*(M*NN+1)+2*NODE+3+2*NJ
```

where

NST = total number of storeys in building.

NJ = number of columns (dummy columns included) on a typical floor.

M = number of slab panels in X direction.

N = number of slab panels in Y direction.

NSP = total number of slab panels in a typical floor.

NDP = number of different slab panel sizes.

NN = number of finite element divisions per slab panel.

NCS = number of boundary lines common to two panels on a given floor.

NIC = Number of "interior" columns, i.e. columns with four slab panels framing into them.

NDM = Number of joint stiffnesses to be assembled and condensed per cycle.

The storage required for the floor analysis is

$$N_1 = NT1+2*NT2+NT3+9*NODE*NT4$$

2. For the formation of the building lateral stiffness matrix, calculate

LE = 6*NJ*(3*NST+6)

LC = 9*NJ*NJ

NNM = 3*NST+3

LSL = NNM*(NNM+1)/2+3*NNM

L1 = MAX0(NT1,LE)

L2 = MAX0(LSL,L1+LC)

L3 = 68*NJ+21*NST+9*(NCP+NBP+2)+NST*(3*NPB+6*NB+NJ)+8*NFEF+5*(NPP+1)+5*NTRU

where

NB = number of bays where beam elements occur.

NPB = number of bays where wall panels occur.

NCP = number of column types.

NBP = number of beam types.

NPP = number of wall panel types.

NFEF = number of sets of different fixed-end beam forces.

NTRU = number of bracing elements in the building.

The required storage for this step is given by

$$N_2 = NT1+L1+L2+L3$$

3. For the solution of static load case, calculate

N_3 = 20*NST+NSS*(NSS+5)+3*NST(3*NST+3)

where

NSS = 3*NST if three degrees of freedom per storey are allowed in the analysis.

= NST if only one degree of freedom per storey is allowed in the analysis.

4. If slab forces are required from the analysis, calculate

MLD = MAXO(5+NFQ,5+NTIME)

K_1 = 2*NST+8*NLD+146NDP+N*M+NSP+(M*NN+1)*(N*NN+1)

K_2 = 9*NODE*(NJ+NODE+M*NN+2)

K_3 = 35*NSF+2*NT2

K_4 = MLD*(3*NT2+3*NJ)+6*(NLD+NMD)+3*NFQ

where

NSF = number of storeys where slab forces are required.

NLD = number of loading combinations.

NFQ = number of frequencies to be calculated.

NTIME = number of times that output is required from the response analysis.

NMD = number of modes that output is required from the spectrum analysis.

NFQ = NTIME = NMD = 0 for static load case.

The required storage for this step is given by

$N_4 = K_1+K_2+K_3+K_4$

5. If a dynamic response analysis is required, calculate

$$N_5 = 8*NST+(6+NTIME)*NSS+2*NPC+NTIME$$

where

NPC = number of time points at which ground acceleration is specified.

The minimum value required for n is the maximum of the set of values of N_1, N_2, N_3, N_4 and N_5.

For typical buildings, N_1 will usually govern if fine element mesh is used to represent the slab panel. N_2 and N_3 may be critical if the building is relatively tall and three degrees of freedom per floor are allowed. N_4 may be critical if slab forces at many locations are required. N_5 may be critical if a large number of output times are required.

APPENDIX B

INTERNAL ORGANIZATION OF COMPUTER PROGRAM

The program is divided into the following seven major parts (subroutine organization is shown in Figure B1):

1. The first operation performed by the program is to read the basic control information and data associated with the complete building (MAIN program, subroutine INPUTT).
2. The next operation involves the formation of the bending stiffness matrix of a typical flat plate floor. Backsubstitution equations are stored on tapes for subsequent calculations of slab forces (subroutine FLSLAB).
3. Formation of frame lateral stiffness matrix for the building. The lateral stiffness matrix and backsubstitution equations are stored sequentially on tapes (subroutine FORM).
4. Transformation of frame lateral stiffness matrix to the building reference point. For static load case, the reference point is taken at the origin of the global coordinate system, therefore no transformation required. For dynamics analysis, the reference point is taken at the center of mass for the storey considered (subroutine LAT).
5. The system is solved for one or two of the following conditions:
   (a) The static lateral loads A and B and the vertical loads I, II, and III applied (subroutine SOLVE).
   (b) The three-dimensional mode shapes and frequencies are evaluated. The earthquake acceleration spectra is read and the maximum storey displacements associated with each mode shapes are calculated (subroutine EARTH).
   (c) Earthquake ground motion is read and the structure displacements are computed for each time step (subroutine DYN).

6. The loading combination cards are read and the total storey displacements are evaluated. Joint displacements are calculated by backsubstituting for each static load case and each modal spectral case or response time increment. As these joint displacements are determined the member forces are also evaluated and are combined according to the loading combination cards (subroutine DISP, FRAME).
7. Read the storeys and slab lines where slab forces are required. Joint displacements of all the slab nodes are calculated by backsubstitution for each static load case and each modal spectral case or response time increment. The slab forces are then calculated and are combined according to the loading combination cards. Slab forces at a joint are the average of joint forces from all elements framing into the joint (subroutine SLAB).

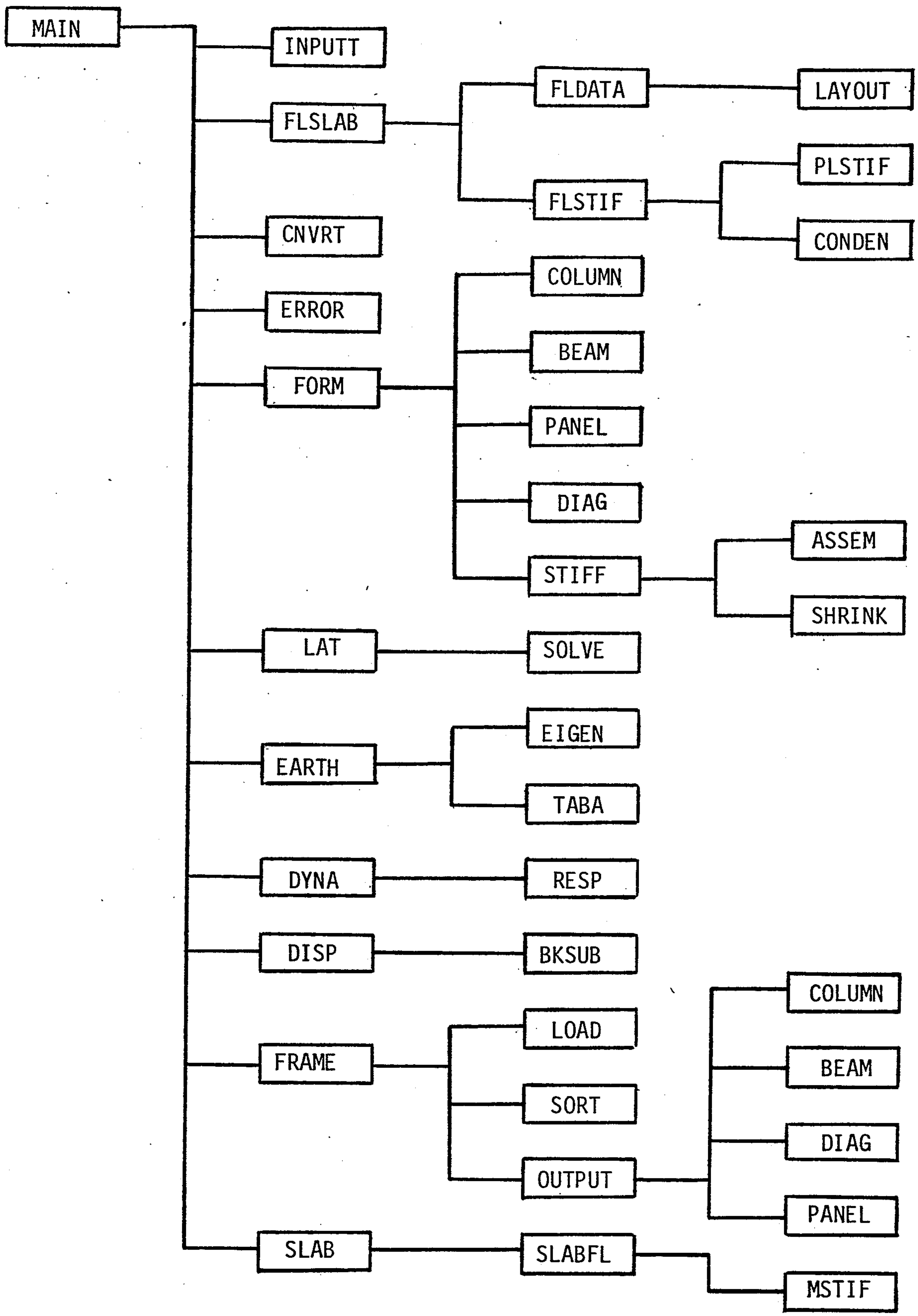


FIGURE B1 PROGRAM ORGANIZATION

APPENDIX C

PROGRAM LISTING

```
C    *******************************************************************
C    *
C    *               ANALYSIS OF THREE-DIMENSIONAL FRAMES
C    *
C    *                            WITH
C    *
C    *                      FLAT  PLATE FLOOR
C    *                      (PROGRAM TABSLAB)
C    *
C    *                   PROGRAMMED BY K. H. MA
C    *
C    *               DEPARTMENT OF CIVIL ENGINEERING
C    *
C    *                  UNIVERSITY OF MANITOBA
C    *
C    *                     WINNIPEG  CANADA
C    *
C    *                       OCTOBER 1976
C    *
C    *
C    *  TAPE 1 TAPE 2 TAPE 3 TAPE 9 TAPE 11 TAPE 12 TAPE 13 TAPE 14
C    *            TAPE 5 = INPUT  TAPE 6 = OUTPUT
C    *
C    *******************************************************************
C
      INTEGER RD, WT
      INTEGER*2 INPT(80), B(2)/'X', 'Y'/
      COMMON /JPW/RD, WT
      COMMON /INPUT/INPT, FN(8)
      COMMON /GEN/NST, NSP, NLD, NAT, NFQ, NSD, NSS, RLAB(3), IS, I3,
     &BHED(20)
      INTEGER MTOT1/'MTOT'/
      DATA B1/'  X '/, B2/'  Y '/, B3/'POTN'/
      COMMON /JUNK/JK(3), MM, MN, JUK(270)
      COMMON /STIF/STF(336)
      COMMON /DYN/NTIME, NPC, DT, DAMP
      COMMON /LSLAB/M, N, NSP, NDP, NJ, NN, NODE
      COMMON /TEV/T, E, V/MDIM/NT1, NT2, NT3, NT4, NT5, NT6, MT4, MT5,
     &NODE3, ITE
C
C     PROGRAM CAPACITY CONTROLLED BY THE FOLLOWING TWO STATEMENTS
C
      COMMON A(16500)
      MTOT = 16500
C
C     INITIALIZES COMMON BLOCK DATA
C
      RD = 5
      WT = 6
      RLAB(1) = B1
      RLAB(2) = B2
      RLAB(3) = B3
C
C     READ JOB TITLE AND GENERAL INFORMATION
C
      WRITE (WT,210)
      READ (RD,190) BHED
      WRITE (WT,220) BHED
      READ (RD,200) INPT
      CALL CNVRT(1, 7)
      NST = FN(1)
      NLD = FN(2)
      NSP = FN(3)
      NJ = FN(4)
      NSD = FN(5)
      ITE = FN(6)
      NAT = FN(7)
      NFQ = FN(8)
      DO 20 I = 41, 50
      IF (INPT(I) .NE. B(1)) GO TO 10
      NSD = 1
      GO TO 30
   10 IF (INPT(I) .NE. B(2)) GO TO 20
      NSD = 2
      GO TO 30
   20 CONTINUE
   30 CONTINUE
      READ (RD,200) INPT
      CALL CNVRT(1, 8)
      M = FN(1)
      N = FN(2)
      NSP = FN(3)
      NDP = FN(4)
      NCS = FN(5)
      NIC = FN(6)
      NN = FN(7)
      NDM = FN(8)
      IF (NN .EQ. 0) NN = 4
      IF (NDM .EQ. 0) NDM = 8
      NAT1 = NAT + 1
      GO TO (40, 50, 60, 70, 80), NAT1
   40 WRITE (WT,230)
      GO TO 90
   50 WRITE (WT,240)
      GO TO 90
   60 WRITE (WT,250)
      GO TO 90
   70 WRITE (WT,260)
      GO TO 90
   80 WRITE (WT,270)
   90 I3 = 3
      IS = 0
      IF (NSD .EQ. 0) GO TO 100
      I3 = 1
      IS = NSD - 1
  100 NSS = NST*I3
      NFS = NST*3
      IF (NFQ .GT. NSS) NFQ = NSS
      IF (NAT .EQ. 0) NFQ = 0
      REWIND 1
      REWIND 2
      REWIND 3
C
C     READ(OR GENERATE)AND PRINT OF STORY HEIGHT AND LATERAL LOAD
C     INFORMATION
C
      N1 = 1
      N2 = N1 + 18*NST
      CALL INPUTT(A(N1), NST, NAT)
C
C     READ AND PRINT (OR GENERATE) FLOOR INFORMATION; FORM FLOOR
C     BENDING STIFFNESS
C
      NT1 = NJ*3*(NJ*3 + 1)/2
      NT2 = NSP*(NN + 1)**2 - NJ - NCS*(NN + 1) + NIC
      NT3 = M*NN + 1
      NT4 = N*NN + 1
      MT4 = M + 1
      MT5 = N + 1
      NODE = MAX0(NDM, MT4)
      NODE3 = 3*NODE
      NT5 = 2*NT3 + NODE + 2
```

```
      NT6 = NT3 + NODE + 1
      N4 = N2 + NJ*2
      N5 = N4 + NT1
      N6 = N5 + M*N
      N7 = N6 + MT4*MT5
      N8 = N7 + NDP
      N9 = N8 + NDP
      N10 = N9 + NT2
      N11 = N10 + NSP
      N12 = N11 + NSP
      N13 = N12 + NSP
      N14 = N13 + NSP
      N15 = N14 + NSP
      N16 = N15 + NT3*NT4
      N17 = N16 + NT2
      N18 = N17 + 3*NODE3*NT5
      N19 = N18 + 3*NODE3*NJ
      N20 = N19 + 3*NODE3*NT6
      N21 = N20 + 3*NODE3*NJ
      N22 = N21 + 144*NDP
      IF (N22 .LE. MTOT) GO TO 110
      MTOT = N22
      CALL ERROR(MTOT1, MTOT, WT)
  110 CALL PLSLAB(A(N2), 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), A(N20), A(N21))
C
C     READ FRAME PROPERTIES AND FORM LATERAL STIFNESSES
C
      NC = NJ
      READ (RD,200) INPT
      CALL CNVRT(1, 7)
      NB = FN(1)
      NCP = FN(2)
      NBP = FN(3)
      NFEF = FN(4)
      NPB = FN(5)
      NPP = FN(6)
      NTRU = FN(7)
C
      NA = 0
      MM = 3*NC
      NCP = NCP + 1
      NBP = NBP + 1
      NPP = NPP + 1
      NL = 6*NC + 3*NST + 3
      N6 = N5 + 9*NCP
      N9 = N6 + NST*NPB*3
      N10 = N9 + 9*NBP
      N11 = N10 + 7*NFEF
      N12 = N11 + 3*NB*NST
      N13 = N12 + 3*NB*NST
      N15 = N13 + NC*NST
C
C     COMPUTE STIFFNESS FOR REDUCED STORGE
C
      CALL STIFF(NA, S, R, NL)
C
      N16 = N15 + NA
      N17 = N16
      N18 = N17 + NFEF
      N19 = N18 + 5*NPP
      N20 = N19 + 3*NTRU
      N21 = N20 + 2*NTRU
      IF (N21 .LE. MTOT) GO TO 120
      MTOT = N21
      CALL ERROR(MTOT1, MTOT, WT)
  120 CALL FORM(A(N1), A(N2), A(N4), A(N5), A(N6), A(N9), A(N10), A(N11)
     &, A(N12), A(N13), A(N15), A(N16), NST, NB, NC, NCP, NBP, NFEF, NPB
     &, A(N17), A(N18), NL, A(N19), A(N20), NTRU, NPP)
C
C     SOLVE STATIC LOAD CASES
C
      N3 = N2 + NSS*NSS
      N4 = N3 + 5*NSS
      N5 = N4 + NFS*NFS
      N6 = N5 + 3*NFS
      N7 = N6 + NST*2
      IF (N7 .LE. MTOT) GO TO 130
      MTOT = N7
      CALL ERROR(MTOT1, MTOT, WT)
  130 CALL LAT(A(N1), A(N2), A(N3), A(N4), A(N5), A(N6), NSS, NFS, NST)
C
C     MODE SHAPES AND FREQUENCIES AND SPECTRUM RESPONSE
C
      N2 = N1 + 8*NST
      N3 = N2 + NSS*NSS
      N4 = N3 + NSS
      N5 = N4 + NSS
      N6 = N5 + NSS
      N7 = N6 + NSS*NSS
      N8 = N2 + (NFQ + 5)*NSS
      NMD = 0
      IF (N8 .GT. N7) N7 = N8
      IF (N7 .LE. MTOT) GO TO 140
      MTOT = N7
      CALL ERROR(MTOT1, MTOT, WT)
  140 CALL EARTH(A(N1), A(N2), A(N3), A(N4), A(N5), A(N6), A(N6), NSS,
     &NST, NMD)
C
C     COMPUTE DYNAMIC RESPONSE
C
      IF (NAT .NE. 4) GO TO 160
      MLD = 5 + NTIME
      N3 = N2 + MLD*NSS
      N4 = N3 + 2*NPC
      N5 = N4 + NTIME
      N6 = N5 + NSS
      IF (N6 .LE. MTOT) GO TO 150
      MTOT = N6
      CALL ERROR(MTOT1, MTOT, WT)
  150 CALL DYNA(A(N2), A(N3), A(N4), A(N5), NSS)
  160 CONTINUE
C
C     COMPUTE FRAME DISPLACEMENTS
C
      IF (NAT .LE. 2) NFO = 0
      IF (NAT .EQ. 1) GO TO 180
      MLD = 5 + NFO
      IF (NAT .EQ. 4) MLD = 5 + NTIME
      N3 = N2 + MLD*NSS
      N4 = N3 + NST*2
      N5 = N4 + NFS*MLD
      IF (N5 .LE. MTOT) GO TO 170
      MTOT = N5
      CALL ERROR(MTOT1, MTOT, WT)
  170 CALL DISP(A(N1), A(N2), A(N3), A(N4), NSS, MLD, NST, NFS)
  180 CONTINUE
C
C     CALCULATE AND PRINT DISPLACEMENTS AND STRESSES
C
      IF (NAT .EQ. 1) STOP
```

```
      CALL FRAME(MTOT, NMD, NFS)
      IF (NSF .EQ. 0) STOP
      CALL SLAB(MTOT, MTOT1, NMD)
      STOP
C
  190 FORMAT (20A4)
  200 FORMAT (80A1)
  210 FORMAT ('1', 20X, 'LATERAL AND/OR GRAVITY LOAD ANALYSIS OF SHEAR W
     &ALL-FRAME STRUCTURE WITH FLAT PLATE FLOORS'///)
  220 FORMAT (' ', 20X, 'JOB TITLE : ', 20A4///)
  230 FORMAT (' ', 20X, 'TYPE OF ANALYSIS : STATIC ANALYSIS ONLY'////)
  240 FORMAT (' ', 20X, 'TYPE OF ANALYSIS : DYNAMIC ANALYSIS, MODE SHAPE
     &S AND FREQUENCIES ONLY, PRINTED'////)
  250 FORMAT (' ', 20X, 'TYPE OF ANALYSIS : STATIC ANALYSIS PLUS MODE SH
     &APES AND FREQUENCIES'////)
  260 FORMAT (' ', 20X, 'TYPE OF ANALYSIS : STATIC ANALYSIS PLUS MODE SH
     &APES, FREQUENCIES AND SEISMIC SPECTRUM ANALYSIS'////)
  270 FORMAT (' ', 20X, 'TYPE OF ANALYSIS : STATIC ANALYSIS PLUS MODE SH
     &APES, FREQUENCIES AND SEISMIC RESPONSE ANALYSIS'////)
C
      END
C     *****************************************************************
C     * THIS SUBROUTINE READS(OR GENERATES)AND PRINTS STOREY HEIGHTS  *
C     *                AND LATERAL LOAD INFORMATION                   *
C     *****************************************************************
      SUBROUTINE INPUTT(SD, NST, NAT)
      DIMENSION SD(NST, 18)
      INTEGER RD, WT
      INTEGER*2 INPT(80), G/'G'/
      COMMON /JRW/RD, WT
      COMMON /INPUT/INPT, FN(8)
C
C     READ (OR GENERATE) AND PRINT OF STORY DATA AND LATERAL LOAD
C     INFORMATION.
C
      WRITE (WT,140)
      IF (NAT .EQ. 0) WRITE (WT,160)
      IF (NAT .NE. 0) WRITE (WT,150)
      M = 0
      KT = 0
      DO 100 N = 1, NST
      LN = NST + 1 - N
      IF (M .NE. 0) GO TO 60
      IF (KT .EQ. 1) GO TO 10
      READ (RD,110) INPT
   10 CALL CNVRT(1, 8)
      KT = 0
      DO 20 I = 2, 8
   20 SD(N, I) = FN(I)
      READ (RD,110) INPT
      CALL CNVRT(1, 8)
      SD(N, 9) = FN(1)
      SD(N, 10) = FN(2)
      SD(N, 15) = FN(3)
      SD(N, 16) = FN(4)
      SD(N, 12) = FN(5)
      SD(N, 13) = FN(6)
      SD(N, 17) = FN(7)
      SD(N, 18) = FN(8)
      IF (N .EQ. NST) GO TO 50
      READ (RD,110) INPT
      DO 30 J = 1, 10
      IF (INPT(J) .EQ. G) GO TO 40
   30 CONTINUE
      KT = 1
      GO TO 50
   40 CALL CNVRT(1, 2)
      M = FN(2)
   50 SD(N, 11) = - SD(N, 9)*(SD(N, 16) - SD(N, 6)) + SD(N, 10)*(SD(N,
     &15) - SD(N, 5))
      SD(N, 14) = - SD(N, 12)*(SD(N, 18) - SD(N, 6)) + SD(N, 13)*(SD(N,
     &17) - SD(N, 5))
      GO TO 80
   60 M = M - 1
      DO 70 I = 2, 18
      SD(N, I) = SD(N - 1, I)
   70 CONTINUE
   80 IF (NAT .NE. 0) GO TO 90
      WRITE (WT,130) LN, SD(N, 2)
      GO TO 100
   90 WRITE (WT,120) LN, (SD(N, I), I = 2, 8)
  100 CONTINUE
C
      RETURN
C
  110 FORMAT (80A1)
  120 FORMAT (I6, 7F15.2)
  130 FORMAT (I6, F15.2)
  140 FORMAT (' ', 43X, '........ DESCRIPTION OF STRUCTURE ........', //
     &//' STORY DATA', /)
  150 FORMAT (' ', 14X, ' STORY', 9X, ' STORY', 4X, 'ROT INERTIA', 5X, '
     &COORDINATE OF MASS CENTRE', 7X, 'EXT STIF', 7X, 'EXT STIF', /' LEV
     &EL NO', 6X, 'HEIGHT', 8X, 'MASS(M)', 9X, 'M*R**2', 11X, 'X(M)',
     &11X, 'Y(M)', 12X, 'K-X', 12X, 'K-Y')
  160 FORMAT (' ', 14X, ' STORY', /' LEVEL NO', 6X, 'HEIGHT')
      END
C     *****************************************************************
C     * THIS SUBROUTINE READS AND PRINTS FLAT PLATE FLOOR INFORMATION *
C     *****************************************************************
      SUBROUTINE FLSLAB(CLN, KK, NP, IM1, A1, B1, LH, MMI, X, Y, T1, T2,
     &IM, KH, SS, SC, S, C, K)
      INTEGER RD, WT
      INTEGER*2 INPT(80)
      COMMON /JRW/RD, WT
      COMMON /INPUT/INPT, FN(8)
      COMMON /LSLAB/M, N, NSP, NDP, NJ, NN, NODE
      COMMON /TEV/T, E, V/MDIM/NT1, NT2, NT3, NT4, NT5, NT6, MT4, MT5,
     &NODE3, ITF
      COMMON /JUNK/XL(20), YL(20), IR(235)
      REAL*4 K(12, 12, NDP), KK(NT1)
      DIMENSION NP(N, M), A1(NDP), B1(NDP), MMI(NSP), LH(NT2), KH(NT2),
     &X(NSP), Y(NSP), CLN(NJ, 2), T1(NSP), T2(NSP), IM(NT4, NT3), SS(
     &NODE3, 3, NT5), SC(NODE3, 3, NJ), S(NODE3, 3, NT6), C(NODE3, 3, NJ
     &), IM1(MT5, MT4)
C
C     READ XL
C
      READ (RD,100) INPT
      CALL CNVRT(1, 8)
      M1 = 8
      IF (M .LE. 8) M1 = M
      DO 10 I = 1, M1
   10 XL(I) = FN(I)
      IF (M .LE. 8) GO TO 30
      READ (RD,100) INPT
      CALL CNVRT(1, 8)
      DO 20 I = 9, M
   20 XL(I) = FN(I - 8)
C
C     READ YL
C
   30 READ (RD,100) INPT
```

```
      CALL CNVRT(1, 8)
      M1 = 8
      IF (N .LE. 8) M1 = N
      DO 40 I = 1, M1
   40 YL(I) = FN(I)
      IF (N .LE. 8) GO TO 60
      READ (RD,100) INPT
      CALL CNVRT(1, 8)
      DO 50 I = 9, N
   50 YL(I) = FN(I - 8)
C
C     READ SLAB PANEL LOCATIONS
C
   60 NT = N*M/8
      NREM = N*M - NT*8
      IF (NREM .NE. 0) NT = NT + 1
      DO 80 I = 1, NT
      READ (RD,100) INPT
      CALL CNVRT(1, 8)
      M2 = (I - 1)*8
      DO 70 J = 1, 8
   70 IR(M2 + J) = FN(J)
   80 CONTINUE
      DO 90 I = 1, N
      M2 = (I - 1)*M
      DO 90 J = 1, M
   90 NP(I, J) = IR(M2 + J)
C
C     READ SLAB PANEL PROPERTIES
C
      READ (RD,100) INPT
      CALL CNVRT(1, 3)
      T = FN(1)
      E = FN(2)
      V = FN(3)
      CALL FLDATA(NP, MMI, X, Y, CLN, T1, T2, A1, B1, IM, IM1)
      CALL FLSTIF(A1, B1, LH, KH, MMI, IM, SS, SC, S, C, K, KK, NP)
      RETURN
  100 FORMAT (80A1)
      END
C     ******************************************************************
C     * THIS SUBROUTINE GENERATES INFORMATION FOR THE FLAT PLATE FLOOR *
C     ******************************************************************
      SUBROUTINE FLDATA(NP, MMI, X, Y, CLN, T1, T2, A1, B1, IM, IM1)
      INTEGER RD, WT
      COMMON /JFW/RD, WT
      INTEGER NSP1/'NSP'/
      INTEGER NJ1/'NJ'/
      INTEGER NDP1/'NDP'/, NT21/'NT2'/
      COMMON /LSLAB/M, N, NSP, NDP, NJ, NN, NODE
      COMMON /MDIM/NT1, NT2, NT3, NT4, NT5, NT6, MT4, MT5, NODE3, ITE/
     &TEV/T, E, V
      COMMON /JUNK/XL(20), YL(20), DX(20), DY(20)
      DIMENSION NP(N, M), MMI(NSP), X(NSP), Y(NSP), CLN(NJ, 2), T1(NSP),
     &T2(NSP), A1(NDP), B1(NDP), IM(NT4, NT3), IM1(MT5, MT4)
C
C     ASSIGN PANEL NUMBERS TO EACH SLAB PANEL
C
      J1 = 1
      DO 10 I = 1, N
      DO 10 J = 1, M
      IF (NP(I, J) .EQ. 0) GO TO 10
      NP(I, J) = J1
      MMI(J1) = (I - 1)*M + J
      J1 = J1 + 1
   10 CONTINUE
      IF (NSP .EQ. J1 - 1) GO TO 20
      NSP = J1 - 1
      CALL ERROR(NSP1, NSP, WT)
   20 CONTINUE
C
C     ASSIGN NODAL NUMBERS TO COLUMN LINES AND FINITE ELEMENT NODES-IM
C
      NN1 = NN + 1
      JC = 1
      JM = 1
      DO 30 I = 1, NT4
      DO 30 J = 1, NT3
   30 IM(I, J) = 0
      DO 130 I = 1, N
      K3 = NN
      IF (I .EQ. N) K3 = NN1
      DO 120 I1 = 1, K3
      LPP = NN*I + I1 - NN
      DO 110 J = 1, M
      IF (NP(I, J) .EQ. 0) GO TO 40
      K4 = NN
      IF (J .EQ. M) K4 = NN1
      GO TO 70
   40 IF (I .EQ. 1 .AND. J .EQ. 1) GO TO 110
      IF (J .EQ. 1) GO TO 60
      IF (NP(I, J - 1) .EQ. 0) GO TO 50
      K4 = 1
      IF (I .EQ. 1) GO TO 70
      IF (NP(I - 1, J) .EQ. 0) GO TO 70
      IF (I1 .GT. 1) GO TO 70
      K4 = NN
      IF (J .EQ. M) K4 = NN1
      GO TO 70
   50 IF (I .EQ. 1) GO TO 110
   60 IF (NP(I - 1, J) .EQ. 0) GO TO 110
      IF (I1 .GT. 1) GO TO 110
      K4 = NN
      IF (J .EQ. M) K4 = NN1
   70 DO 100 I2 = 1, K4
      LT = NN*J + I2 - NN
      IF (I2 .NE. 1 .AND. I2 .NE. NN1) GO TO 80
      IF (I1 .EQ. 1 .OR. I1 .EQ. NN1) GO TO 90
   80 IM(LPP, LT) = JC
      JC = JC + 1
      GO TO 100
   90 IM(LPP, LT) = 10000 + JM
      JM = JM + 1
  100 CONTINUE
  110 CONTINUE
  120 CONTINUE
  130 CONTINUE
      IF (NT2 .EQ. JC - 1) GO TO 140
      NT2 = JC - 1
      CALL ERROR(NT21, NT2, WT)
  140 CONTINUE
      IF (NJ .EQ. JM - 1) GO TO 150
      NJ = JM - 1
      CALL ERROR(NJ1, NJ, WT)
C
C     OBTAIN DIMENSIONS OF DIFFERENT TYPES OF PANEL
C
  150 CONTINUE
      DO 180 I = 1, NSP
      I3 = MMI(I)
      I4 = I3/M
      IF (I4*M .EQ. I3) GO TO 160
```

```
      I1 = I3/M + 1
      GO TO 170
  160 I1 = I3/M
  170 I2 = I3 - (I1 - 1)*M
      X(I) = XL(I2)
      Y(I) = YL(I1)
      T1(I) = XL(I2)
      T2(I) = YL(I1)
  180 CONTINUE
      J1 = 1
      DO 220 I = 1, NSP
      IF ((T1(I) .EQ. 0.) .AND. (T2(I) .EQ. 0.)) GO TO 220
      IJ = I + 1
      IF (IJ .GT. NSP) GO TO 210
      DO 200 J = IJ, NSP
      IF ((T1(I) .EQ. T1(J)) .AND. (T2(I) .EQ. T2(J))) GO TO 190
      GO TO 200
  190 T1(J) = 0.
      T2(J) = 0.
      MMI(J) = J1
  200 CONTINUE
  210 A1(J1) = T1(I)
      B1(J1) = T2(I)
      T1(I) = 0.
      T2(I) = 0.
      MMI(I) = J1
      J1 = J1 + 1
  220 CONTINUE
      IF (NDP .EQ. J1 - 1) GO TO 230
      NDP = J1 - 1
      CALL ERROR(NDP1, NDP, WT)
  230 CONTINUE
      DO 240 I = 1, MT5
      I1 = (I - 1)*NN + 1
      DO 240 J = 1, MT4
      J1 = (J - 1)*NN + 1
      IM1(I, J) = IM(I1, J1) - 10000
  240 CONTINUE
C
C     GENERATE COLUMN LINE CORDINATES FOR THE FLOOR
      DX(1) = 0.
      DO 250 I = 2, MT4
      DX(I) = DX(I - 1) + XL(I - 1)
  250 CONTINUE
      DY(1) = 0.
      DO 260 I = 2, MT5
      DY(I) = DY(I - 1) + YL(I - 1)
  260 CONTINUE
      IC = 1
      DO 280 I = 1, MT5
      DO 270 J = 1, MT4
      IF (IM1(I, J) .EQ. 0) GO TO 270
      CLN(IC, 1) = DX(J)
      CLN(IC, 2) = DY(I)
      IC = IC + 1
  270 CONTINUE
  280 CONTINUE
C
C     PLOT LAYOUT OF FLOOR PLAN
C
      WRITE (WT,290) T, E, V
      CALL LAYOUT(M, N, XL, YL, NP, IM1, MT4, MT5)
      RETURN
  290 FORMAT (///' TYPICAL FLOOR PLAN', /' (SHOWING FLOOR PANEL AND COLU
     &MN NUMBERING)', //' PANEL THICKNESS T =', F14.2/' ELASTIC MODULUS
     &E =', F14.2/' POSION RATIO     V =', F14.2/' COLUMNS .............
     &.......... ----5----', //)
      END
C     ******************************************************************
C     * THIS SUBROUTINE PLOTS LAYOUT OF A TYPICAL FLAT PLATE FLOOR     *
C     ******************************************************************
      SUBROUTINE LAYOUT(M, N, XL, YL, NP, IM, MM1, NN1)
      INTEGER*2 XW, X1/' '/, X2/'X'/, X3/'-'/, X4/'|'/, X5/'Y'/
      INTEGER*2 B(10)/'1', '2', '3', '4', '5', '6', '7', '8', '9', '0'/
      INTEGER*2 A(120)
      INTEGER RD, WT
      COMMON /JRW/RD, WT
      DIMENSION XL(M), YL(N), NP(N, M), IM(NN1, MM1)
      NS = 110/M
      NS = MINO(NS, 18)
      ND = NS*5/6
      MD1 = M*NS + 3
      MD2 = MD1 + 5
      MD3 = MD2 + 1
      WRITE (WT,400) X5
      DO 10 I = 1, 5
   10 WRITE (WT,400) X4
      DO 390 I = 1, NN1
      NN = NN1 - I + 1
      NNS = ND
      IF (I .EQ. NN1) NNS = 1
      DO 380 K = 1, NNS
      IF (K .NE. 1) GO TO 220
      DO 20 L1 = 1, 113
   20 A(L1) = X1
      DO 190 J = 1, MM1
      IF (IM(NN, J)) 190, 190, 30
   30 N2 = 10000
      N3 = 10000
      IF (IM(NN, J) - 100) 40, 70, 70
   40 IF (IM(NN, J) - 10) 50, 60, 60
   50 N1 = IM(NN, J)
      GO TO 80
   60 N1 = IM(NN, J)/10
      N2 = IM(NN, J) - 10*N1
      GO TO 80
   70 N1 = IM(NN, J)/100
      N2 = (IM(NN, J) - N1*100)/10
      N3 = IM(NN, J) - N1*100 - N2*10
   80 K1 = (J - 1)*NS
      A(K1 + 1) = B(N1)
      IF (N2 .EQ. 10000) GO TO 120
      IF (N2 .EQ. 0) GO TO 90
      A(K1 + 2) = B(N2)
      GO TO 100
   90 A(K1 + 2) = B(10)
  100 IF (N3 .EQ. 10000) GO TO 120
      IF (N3 .EQ. 0) GO TO 110
      A(K1 + 3) = B(N3)
      GO TO 120
  110 A(K1 + 3) = B(10)
  120 IF (J .EQ. MM1) GO TO 190
  130 IF (NN .EQ. 1) GO TO 150
      IF (NP(NN - 1, J)) 140, 140, 160
  140 IF (NN .EQ. NN1) GO TO 190
  150 IF (NP(NN, J) .EQ. 0) GO TO 190
  160 XW = X3
  170 DO 180 L2 = 1, NS
      IF (A(K1 + L2) .NE. X1) GO TO 180
      A(K1 + L2) = XW
  180 CONTINUE
  190 CONTINUE
```

```
      IF (I .NE. NN1) GO TO 210
      DO 200 L1 = MD1, MD2
  200 A(L1) = X3
      A(MD3) = X2
      WRITE (WT,400) (A(IL), IL = 1, MD3)
      GO TO 380
  210 WRITE (WT,400) (A(IL), IL = 1, 113)
      GO TO 380
  220 IF (K .NE. 2) GO TO 300
      DO 230 L1 = 1, 113
  230 A(L1) = X1
      DO 290 J = 1, MM1
      IF (J .EQ. MM1) GO TO 260
  240 IF (NP(NN - 1, J)) 250, 250, 270
  250 IF (J .EQ. 1) GO TO 290
  260 IF (NP(NN - 1, J - 1) .EQ. 0) GO TO 290
  270 XW = X4
  280 K1 = (J - 1)*NS
      A(K1 + 1) = XW
  290 CONTINUE
      GO TO 370
  300 IF (K .NE. ND/2 + 1) GO TO 370
      DO 350 J = 1, M
      IF (NP(NN - 1, J) .EQ. 0) GO TO 350
      N2 = 10000
      IF (NP(NN - 1, J) - 10) 310, 320, 320
  310 N1 = NP(NN - 1, J)
      GO TO 330
  320 N1 = NP(NN - 1, J)/10
      N2 = NP(NN - 1, J) - N1*10
  330 K1 = (J - 1)*NS + NS/2
      A(K1) = B(N1)
      IF (N2 .EQ. 10000) GO TO 350
      IF (N2 .EQ. 0) GO TO 340
      A(K1 + 1) = B(N2)
      GO TO 350
  340 A(K1 + 1) = B(10)
  350 CONTINUE
      WRITE (WT,410) YL(NN - 1), (A(IJ), IJ = 1, 113)
      DO 360 J = 1, M
      K1 = (J - 1)*NS + NS/2
      A(K1) = X1
      A(K1 + 1) = X1
  360 CONTINUE
      GO TO 380
  370 WRITE (WT,400) (A(IJ), IJ = 1, 113)
  380 CONTINUE
  390 CONTINUE
      IF (NS .EQ. 11) WRITE (WT,420) (XL(I), I = 1, M)
      IF (NS .EQ. 12) WRITE (WT,430) (XL(I), I = 1, M)
      IF (NS .EQ. 13) WRITE (WT,440) (XL(I), I = 1, M)
      IF (NS .EQ. 15) WRITE (WT,450) (XL(I), I = 1, M)
      IF (NS .EQ. 18) WRITE (WT,460) (XL(I), I = 1, M)
      RETURN
  400 FORMAT (' ', 10X, 120A1)
  410 FORMAT (' ', 1X, F7.2, 2X, 114A1)
  420 FORMAT (//9X, 10F11.2)
  430 FORMAT (//8X, 9F12.2)
  440 FORMAT (//8X, 8F13.2)
  450 FORMAT (//7X, 7F15.2)
  460 FORMAT (//5X, 6F18.2)
      END
C     ****************************************************************
C     *  THIS SUBROUTINE GENERATES BENDING STIFFNESS MATRIX OF A FLAT  *
C     *                       PLATE FLOOR                              *
C     ****************************************************************
      SUBROUTINE PLSTIF(A1, B1, LH, KH, MMI, IM, SS, SC, S, C, K, KK, NF
     &)
      INTEGER RD, WT
      COMMON /JRW/RD, WT
      COMMON /JUNK/XL(20), YL(20), JK(5), LL(4), JJ(15), II(4), MP(4),
     &MQ(4)
      COMMON /LSLAB/M, N, NSP, NDP, NJ, NN, NODE
      COMMON /TEV/T, E, V/MDIM/NT1, NT2, NT3, NT4, NT5, NT6, MT4, MT5,
     &NODE3, ITE
      REAL*4 K(12, 12, NDP), KK(NT1)
      INTEGER*2 MI(4, 200)
      INTEGER NJ1/'NJ'/
      DIMENSION A1(NDP), B1(NDP), MMI(NSP), LH(NT2), KH(NT2), IM(NT4,
     &NT3), SS(NODE3, 3, NT5), SC(NODE3, 3, NJ), S(NODE3, 3, NT6), C(
     &NODE3, 3, NJ), NP(N, M)
C
C     PLATE ELEMENT STIFFNESS MATRICES
C
      DO 10 I = 1, NDP
      A = A1(I)/NN
      B = B1(I)/NN
      L = I
      IF (ITE .EQ. 0) CALL PLSTIF(A, B, L, K)
      IF (ITE .EQ. 1) CALL PLSTIC(A, B, L, K)
   10 CONTINUE
      NN1 = NN + 1
C
C     FORMULATION OF FLOOR STIFFNESS MATRIX.
C
      DO 20 J = 1, NT1
      KK(J) = 0.
   20 CONTINUE
      IF (NN .EQ. 1) GO TO 530
      DO 30 I1 = 1, NT5
      DO 30 I2 = 1, NODE3
      DO 30 I3 = 1, 3
   30 SS(I2, I3, I1) = 0.
      DO 40 I1 = 1, NJ
      DO 40 I2 = 1, NODE3
      DO 40 I3 = 1, 3
   40 SC(I2, I3, I1) = 0.
      REWIND 14
      IC = 0
C
C     ASSEMBLE STIFFNESS RELATING TO NODAL POINTS ON SLAB PANELS
C
      DO 520 I = 1, N
      K5 = NN
      IF (I .EQ. N) K5 = NN1
      DO 510 IA = 1, K5
      LP = NN*I + IA - NN
      DO 500 J = 1, M
      K4 = NN
      IF (J .EQ. M) K4 = NN1
      DO 490 IB = 1, K4
      LT = NN*J + IB - NN
      NC = IM(LP, LT)
      IF (NC .EQ. 0) GO TO 490
      MP(1) = - 1 + LP
      MP(2) = - 1 + LP
      MP(3) = 1 + LP
      MP(4) = 1 + LP
      MQ(1) = - 1 + LT
      MQ(2) = 1 + LT
      MQ(3) = - 1 + LT
      MQ(4) = 1 + LT
```

```
      IF (NC .GT. 10000) GO TO 410
      IC = IC + 1
      K1 = 3*IC - 2
      K2 = 3*IC
      LU = 1
      KMAX = 0
      LMAX = 0
      DO 210 L1 = 1, 4
      IF (MP(L1) .LT. 1 .OR. MP(L1) .GT. NT4) GO TO 210
      IF (MO(L1) .LT. 1 .OR. MQ(L1) .GT. NT3) GO TO 210
      LL(1) = NC
      LL(2) = IM(MP(L1), LT)
      LL(3) = IM(MP(L1), MO(L1))
      LL(4) = IM(LP, MO(L1))
      IF (LL(2) .EQ. 0 .OR. LL(3) .EQ. 0 .OR. LL(4) .EQ. 0) GO TO 210
      IT = MMI(NP(I, J))
      GO TO (50, 60, 70, 80), L1
   50 JJ(1) = 6
      JJ(2) = 9
      JJ(3) = 0
      JJ(4) = 3
      IF (IA .EQ. 1) IT = MMI(NP(I - 1, J))
      IF (IB .EQ. 1) IT = MMI(NP(I, J - 1))
      GO TO 90
   60 JJ(1) = 3
      JJ(2) = 0
      JJ(3) = 9
      JJ(4) = 6
      IF (IA .EQ. 1) IT = MMI(NP(I - 1, J))
      GO TO 90
   70 JJ(1) = 9
      JJ(2) = 6
      JJ(3) = 3
      JJ(4) = 0
      IF (IB .EQ. 1) IT = MMI(NP(I, J - 1))
      GO TO 90
   80 JJ(1) = 0
      JJ(2) = 3
      JJ(3) = 6
      JJ(4) = 9
   90 CONTINUE
      DO 100 LR = 2, 4
      II(LR) = LL(LR)
      IF (LL(LR) .GT. 10000) II(LR) = 0
  100 CONTINUE
      KMAX = MAX0(LL(1), II(2), II(3), II(4), KMAX)
      IF (IC .NE. 1 .OR. LU .NE. 1) GO TO 140
      KMIN = MINO(LL(1), LL(2), LL(3), LL(4))
      LU = 2
      IF (I .NE. 1) GO TO 110
      IF (IA .EQ. 1) KMIN = 1
      GO TO 140
  110 IF (NP(I - 1, J) .NE. 0 .OR. IA .NE. 1) GO TO 140
      L6 = J
      DO 120 IN = L6, M
      IF (NP(I - 1, IN) .EQ. 0) GO TO 120
      KMIN = IM(LP - 1, (IN - 1)*NN + 1)
      GO TO 140
  120 CONTINUE
      DO 130 IN = 1, M
      IF (NP(I, IN) .EQ. 0) GO TO 130
      KMIN = IM(LP, (IN - 1)*NN + 2)
      GO TO 140
  130 CONTINUE
  140 DO 200 J3 = 1, 4
      IF (LL(J3) .GT. 10000) GO TO 170
      LL(J3) = LL(J3) - KMIN + 1
      DO 160 J1 = K1, K2
      J5 = J1 - K1 + 1
      DO 150 J2 = 1, 3
      SS(J1, J2, LL(J3)) = SS(J1, J2, LL(J3)) + K(JJ(1) + J5, JJ(J3) +
     &J2, IT)
  150 CONTINUE
  160 CONTINUE
      GO TO 200
  170 LL(J3) = LL(J3) - 10000
      LMAX = LL(J3)
      DO 190 J1 = K1, K2
      J5 = J1 - K1 + 1
      DO 180 J2 = 1, 3
      SC(J1, J2, LMAX) = SC(J1, J2, LMAX) + K(JJ(1) + J5, JJ(J3) + J2,
     &IT)
  180 CONTINUE
  190 CONTINUE
  200 CONTINUE
  210 CONTINUE
C
C     CONDENSATION OF NODAL STIFFNESS ON INTERNAL SLAB NODES
C
      IF (NC .EQ. 1) GO TO 220
      KH(NC) = MAX0(KH(NC - 1), KMAX)
      LH(NC) = MAX0(LH(NC - 1), LMAX)
      GO TO 230
  220 KH(NC) = KMAX
      LH(NC) = LMAX
  230 CONTINUE
      ICC = 1
      IF (IC .EQ. NODE) GO TO 240
      IF (NC .NE. NT2) GO TO 490
      ICC = 0
  240 K3 = 3*IC
      IL = NC/NODE - ICC
      IK = NODE*IL + 1
      IT = IK
      IW = (KMIN + NODE - 1)/NODE
      NBSP = IL - IW + 1
      IF (NBSP .EQ. 0) GO TO 320
      DO 250 I1 = 1, NBSP
  250 BACKSPACE 14
  260 DO 310 I1 = IW, IL
      KM3 = 1
      DO 270 I2 = 1, NODE
  270 JJ(I2) = NODE*I1 + I2 - NODE
      IF (I1 .EQ. IW) KM3 = KMIN - JJ(1) + 1
      NTT = LH(I1*NODE)
      KM1 = JJ(1)
      NT = KH(I1*NODE) - KM1 + 1
      READ (14) (((S(M2, M3, M1), M3 = 1, 3), M2 = 1, NODE3), M1 = 1, NT
     &), (((C(M2, M3, M1), M3 = 1, 3), M2 = 1, NODE3), M1 = 1, NTT)
      DO 300 J1 = KM3, NODE
      NT = KH(JJ(J1)) - KM1 + 1
      NTT = LH(JJ(J1))
      I2 = J1
      I22 = JJ(J1) - KMIN + 1
      DO 290 L1 = 1, 3
      L5 = L1 + 3*(J1 - 1)
      DO 280 L2 = 1, K3
      IF (SS(L2, L1, I22) .EQ. 0.) GO TO 280
      CALL CONDEN(1, SS, SC, S, C, KK, NT, I22, L1, L2, L5, I2)
      CALL CONDEN(2, SS, SC, S, C, KK, NTT, I22, L1, L2, L5, 0)
  280 CONTINUE
  290 CONTINUE
```

```
  300 CONTINUE
  310 CONTINUE
  320 DO 380 L1 = 1, K3
      L5 = L1 - (L1 - 1)/3*3
      IK = IT + (L1 - 1)/3
      NT = KH(IK) - KMIN + 1
      NTT = LH(IK)
      IK = IK - KMIN + 1
      DO 340 L2 = IK, NT
      IF = 1
      IF (L2 .EQ. IK) IF = L5 + 1
      IF (IF .GT. 3) GO TO 340
      DO 330 L3 = IF, 3
      IF (SS(L1, L3, L2) .EQ. 0.) GO TO 330
      SS(L1, L3, L2) = SS(L1, L3, L2)/SS(L1, L5, IK)
  330 CONTINUE
  340 CONTINUE
      DO 360 L2 = 1, NTT
      DO 350 L3 = 1, 3
      IF (SC(L1, L3, L2) .EQ. 0.) GO TO 350
      SC(L1, L3, L2) = SC(L1, L3, L2)/SS(L1, L5, IK)
  350 CONTINUE
  360 CONTINUE
      IF = L1 + 1
      IF (L1 .EQ. K3) GO TO 380
      DO 370 L2 = IF, K3
      IF (SS(L2, L5, IK) .EQ. 0.) GO TO 370
      CALL CONDEN(3, SS, SC, S, C, KK, NT, IK, L1, L2, L5, 0)
      CALL CONDEN(5, SS, SC, S, C, KK, NTT, IK, L1, L2, L5, 0)
  370 CONTINUE
  380 CONTINUE
      KMAX1 = KH(NC) - KMIN + 1
      KMAX2 = LH(NC)
      IT = IT - KMIN + 1
      NOD = 3*IC
      WRITE (14) (((SS(I2, I3, I1), I3 = 1, 3), I2 = 1, NOD), I1 = IT,
     &KMAX1), (((SC(I2, I3, I1), I3 = 1, 3), I2 = 1, NOD), I1 = 1, KMAX2
     &)
      DO 390 I1 = 1, KMAX1
      DO 390 I2 = 1, NOD
      DO 390 I3 = 1, 3
  390 SS(I2, I3, I1) = 0.
      DO 400 I1 = 1, KMAX2
      DO 400 I2 = 1, NOD
      DO 400 I3 = 1, 3
  400 SC(I2, I3, I1) = 0.
      IC = 0
      GO TO 490
  410 NC = NC - 10000
      DO 480 L1 = 1, 4
      IF (MP(L1) .LT. 1 .OR. MP(L1) .GT. NT4) GO TO 480
      IF (MO(L1) .LT. 1 .OR. MO(L1) .GT. NT3) GO TO 480
      LL(3) = IM(MP(L1), MO(L1))
      IF (LL(3) .EQ. 0) GO TO 480
      IT = MMI(NP(I, J))
      GO TO (420, 430, 440, 450), L1
  420 J4 = 6
      IF (IA .EQ. NN1 .AND. IB .EQ. NN1) GO TO 460
      IT = MMI(NP(I - 1, J - 1))
      IF (IA .EQ. NN1) IT = MMI(NP(I, J - 1))
      IF (IB .EQ. NN1) IT = MMI(NP(I - 1, J))
      GO TO 460
  430 J4 = 3
      IF (IA .EQ. NN1) GO TO 460
      IT = MMI(NP(I - 1, J))
      GO TO 460
  440 J4 = 9
      IF (IB .EQ. NN1) GO TO 460
      IT = MMI(NP(I, J - 1))
      GO TO 460
  450 J4 = 0
  460 CONTINUE
      IL = 3*NC - 3
      DO 470 J1 = 1, 3
      DO 470 J2 = 1, 3
      IF (J2 .LT. J1) GO TO 470
      K1 = IL + J1
      K2 = IL + J2
      K3 = K2*(K2 - 1)/2 + K1
      KK(K3) = KK(K3) + K(J4 + J1, J4 + J2, IT)
  470 CONTINUE
  480 CONTINUE
  490 CONTINUE
  500 CONTINUE
  510 CONTINUE
  520 CONTINUE
  530 IF (NN .NE. 1) GO TO 710
C
C     GENERATE PANEL NODAL NUMBERS AT CORNER NODES
C
      J1 = 1
      DO 550 J = 1, M
      IF (NP(1, J) .EQ. 0) GO TO 540
      J2 = NP(1, J)
      MI(1, J2) = J1
      MI(2, J2) = J1 + 1
      J1 = J1 + 1
      GO TO 550
  540 IF (J .EQ. 1) GO TO 550
      IF (NP(1, J - 1) .EQ. 0) GO TO 550
      J1 = J1 + 1
  550 CONTINUE
      IF (NP(1, M) .GT. 0) J1 = J1 + 1
      IF (N .EQ. 1) GO TO 610
      DO 600 I = 2, N
      DO 590 J = 1, M
      IF (NP(I, J) .EQ. 0) GO TO 560
      J2 = NP(I, J)
      MI(1, J2) = J1
      MI(2, J2) = J1 + 1
  560 IF (NP(I - 1, J) .EQ. 0) GO TO 570
      J2 = NP(I - 1, J)
      MI(3, J2) = J1
      MI(4, J2) = J1 + 1
      J1 = J1 + 1
      GO TO 590
  570 IF ((NP(I, J) .EQ. 0) .AND. (NP(I - 1, J) .EQ. 0)) GO TO 580
      J1 = J1 + 1
      GO TO 590
  580 IF (J .EQ. 1) GO TO 590
      IF (NP(I, J - 1) .EQ. 0 .AND. NP(I - 1, J - 1) .EQ. 0) GO TO 590
      J1 = J1 + 1
  590 CONTINUE
      IF ((NP(I, M) .EQ. 0) .AND. (NP(I - 1, M) .EQ. 0)) GO TO 600
      J1 = J1 + 1
  600 CONTINUE
  610 DO 630 J = 1, M
      IF (NP(N, J) .EQ. 0) GO TO 620
      J2 = NP(N, J)
      MI(3, J2) = J1
      MI(4, J2) = J1 + 1
      J1 = J1 + 1
```

```
      GO TO 630
  620 IF (J .EQ. 1) GO TO 630
      IF (NP(N, J - 1) .EQ. 0) GO TO 630
      J1 = J1 + 1
  630 CONTINUE
      IF (NP(N, M) .GT. 0) J1 = J1 + 1
      IF (NJ .EQ. J1 - 1) GO TO 640
      NJ = J1 - 1
      CALL ERROR(NJ1, NJ, WT)
  640 CONTINUE
C
C     ASSEMBLE STIFFNESS MATRIX FOR THE CASE OF SINGLE PANEL ELEMENT
C
      DO 700 I = 1, N
      DO 690 J = 1, M
      IF (NP(I, J) .EQ. 0) GO TO 690
      JT = NP(I, J)
      IT = MMI(JT)
      LL(1) = 0
      LL(2) = 3
      LL(3) = 6
      LL(4) = 9
      JJ(1) = 3*MI(1, JT) - 3
      JJ(2) = 3*MI(3, JT) - 3
      JJ(3) = 3*MI(4, JT) - 3
      JJ(4) = 3*MI(2, JT) - 3
      DO 680 I1 = 1, 4
      DO 670 I2 = 1, 4
      IF (JJ(I2) .LT. JJ(I1)) GO TO 670
      DO 660 I3 = 1, 3
      DO 650 I4 = 1, 3
      K1 = JJ(I1) + I3
      K2 = JJ(I2) + I4
      IF (K2 .LT. K1) GO TO 650
      K3 = K2*(K2 - 1)/2 + K1
      KK(K3) = KK(K3) + K(LL(I1) + I3, LL(I2) + I4, IT)
  650 CONTINUE
  660 CONTINUE
  670 CONTINUE
  680 CONTINUE
  690 CONTINUE
  700 CONTINUE
      GO TO 960
C
C     ASSEMBLE STIFFNESS RELATING TO NODAL POINTS ON COLUMN LINES
C
  710 NT = NT2/NODE
      NREM = NT2 - NT*NODE
      IF (NREM .NE. 0) NT = NT + 1
      DO 950 I = 1, MT5
      IC = 0
      KMAX = 0
      LP = NN*I - NN + 1
      MP(1) = LP - 1
      MP(2) = LP - 1
      MP(3) = LP + 1
      MP(4) = LP + 1
      DO 810 J = 1, MT4
      LT = NN*J - NN + 1
      IF (IM(LP, LT) .EQ. 0) GO TO 810
      IC = IC + 1
      MO(1) = LT - 1
      MO(2) = LT + 1
      MO(3) = LT - 1
      MO(4) = LT + 1
      K1 = 3*IC - 2
      K2 = 3*IC
      NC = IM(LP, LT) - 10000
      IF (IC .EQ. 1) IL = NC
      LU = 1
      DO 800 L1 = 1, 4
      IF (MP(L1) .LT. 1 .OR. MP(L1) .GT. NT4) GO TO 800
      IF (MO(L1) .LT. 1 .OR. MO(L1) .GT. NT3) GO TO 800
      LL(1) = IM(MP(L1), LT)
      LL(2) = IM(MP(L1), MO(L1))
      LL(3) = IM(LP, MO(L1))
      IF (LL(2) .EQ. 0) GO TO 800
      GO TO (720, 730, 740, 750), L1
  720 JC = 6
      JJ(1) = 9
      JJ(2) = 0
      JJ(3) = 3
      IT = MMI(NP(I - 1, J - 1))
      GO TO 760
  730 JC = 3
      JJ(1) = 0
      JJ(2) = 9
      JJ(3) = 6
      IT = MMI(NP(I - 1, J))
      GO TO 760
  740 JC = 9
      JJ(1) = 6
      JJ(2) = 3
      JJ(3) = 0
      IT = MMI(NP(I, J - 1))
      GO TO 760
  750 JC = 0
      JJ(1) = 3
      JJ(2) = 6
      JJ(3) = 9
      IT = MMI(NP(I, J))
  760 IF (IC .NE. 1 .OR. LU .NE. 1) GO TO 780
      KMIN = MINO(LL(1), LL(2), LL(3))
      LU = 2
      IF (I .EQ. 1) GO TO 780
      IF (NP(I - 1, J) .NE. 0) GO TO 780
      DO 770 IN = 1, M
      IF (NP(I - 1, IN) .EQ. 0) GO TO 770
      KMIN = IM(LP - 1, (IN - 1)*NN + 1)
      GO TO 780
  770 CONTINUE
  780 KMAX = MAXO(LL(1), LL(2), LL(3), KMAX)
      DO 790 J3 = 1, 3
      LL(J3) = LL(J3) - KMIN + 1
      DO 790 J1 = K1, K2
      J5 = J1 - K1 + 1
      DO 790 J2 = 1, 3
  790 SS(J1, J2, LL(J3)) = SS(J1, J2, LL(J3)) + K(J5 + JC, JJ(J3) + J2,
     &IT)
  800 CONTINUE
  810 CONTINUE
      REWIND 14
C
C     CONDENSATION OF NODAL STIFFNESS ON COLUMN NODES
C
      K3 = 3*IC
      K2 = NODE
      NOD = 3*NODE
      LMAX = KMAX
      IW = (KMIN + NODE - 1)/NODE
      IF (IW .EQ. 1) GO TO 830
      IZ = IW - 1
```

```
      DO 820 I1 = 1, IZ
  820 READ (14)
  830 DO 940 I1 = IW, NT
      K1 = I1*NODE
      K4 = NODE
      KM3 = 1
      IF (I1 .LE. NT2/NODE) GO TO 840
      K2 = NREM
      K1 = (I1 - 1)*NODE + NREM
      NOD = 3*NREM
  840 LMAX = MAX0(LMAX, KH(K1))
      IMAX = LMAX - K1
      DO 850 I2 = 1, K2
  850 JJ(I2) = NODE*I1 + I2 - NODE
      KM1 = JJ(1)
      NTT = LH(K1)
      NNT = KH(K1) - KM1 + 1
      IF (I1 .NE. IW) GO TO 860
      KM3 = KMIN - KM1 + 1
      K4 = K1 - KMIN + 1
  860 READ (14) (((S(M2, M3, M1), M3 = 1, 3), M2 = 1, NOD), M1 = 1, NNT)
     &, (((C(M2, M3, M1), M3 = 1, 3), M2 = 1, NOD), M1 = 1, NTT)
      DO 890 J1 = KM3, K2
      NNT = KH(JJ(J1)) - KM1 + 1
      NTT = LH(JJ(J1))
      I2 = J1
      I22 = J1
      IF (I1 .EQ. IW) I22 = JJ(J1) - KMIN + 1
      DO 890 L1 = 1, 3
      L5 = L1 + 3*(J1 - 1)
      DO 870 L2 = 1, K3
      IF (SS(L2, L1, I22) .EQ. 0.) GO TO 870
      CALL CONDEN(1, SS, SC, S, C, KK, NNT, I22, L1, L2, L5, I2)
      CALL CONDEN(4, SS, SC, S, C, KK, NTT, I22, L1, L2, L5, IL)
  870 CONTINUE
  880 CONTINUE
  890 CONTINUE
      IF (IMAX .EQ. 0) GO TO 920
      DO 900 M1 = 1, IMAX
      M4 = M1 + K4
      DO 900 M2 = 1, K3
      DO 900 M3 = 1, 3
  900 SS(M2, M3, M1) = SS(M2, M3, M4)
      IM1 = IMAX + 1
      IM2 = IMAX + K4
      DO 910 M1 = IM1, IM2
      DO 910 M2 = 1, K3
      DO 910 M3 = 1, 3
  910 SS(M2, M3, M1) = 0.
      GO TO 940
  920 DO 930 M1 = 1, NODE
      DO 930 M2 = 1, K3
      DO 930 M3 = 1, 3
  930 SS(M2, M3, M1) = 0.
  940 CONTINUE
  950 CONTINUE
  960 CONTINUE
      IF (NN .EQ. 1) RETURN
      REWIND 9
      WRITE (9) NP, MMI, IM, A1, B1, LH, KH, XL, YL
      RETURN
      END
C     ****************************************************************
C     * THIS SUBROUTINE PERFORMS THE CONDENSATION OF SYMMETRICAL
C     * MATRIX.  STIFFNESS COEFICIENTS RELATING TO EIGHT PARTICULAR
C     * NODES ARE ASSEMBLED AND CONDENSATION PERFORMED ON THOSE
C     * PARTICULAR NODES.  THIS MODIFIED NODAL STIFFNESS COEFFICIENTS
C     * ARE SAVED FOR BACK SUPSTITUTION FOR MEMBER END FORCES.  THE
C     * PROCEDURE IS REPEATED FOR ALL NODES AND ENDS UP WITH THE
C     * CONDENSED STIFFNESS MATRIX.  THE AMOUNT OF CORE STORAGE IS
C     * THAT REQUIRED FOR ANY 16 NODAL POINTS.
      ****************************************************************
      SUBROUTINE CONDEN(NA, SS, SC, S, C, KK, NT, I1, L1, L2, L5, IL)
      COMMON /LSLAB/JK(4), NJ
      COMMON /MDIM/NT1, NT2, NT3, NT4, NT5, NT6, MT4, MT5, NODE3, ITE
      DIMENSION SS(NODE3, 3, NT5), SC(NODE3, 3, NJ), S(NODE3, 3, NT6),
     &(NODE3, 3, NJ)
      REAL*4 KK(NT1)
      GO TO (10, 40, 70, 100, 130), NA
C
C
   10 DO 30 L3 = IL, NT
      L6 = I1 + L3 - IL
      IF = 1
      IF (L3 .EQ. IL) IF = L1 + 1
      IF (IF .GT. 3) GO TO 30
      DO 20 L4 = IF, 3
      IF (S(L5, L4, L3) .EQ. 0.) GO TO 20
      SS(L2, L4, L6) = SS(L2, L4, L6) - SS(L2, L1, I1)*S(L5, L4, L3)
   20 CONTINUE
   30 CONTINUE
      RETURN
C
C
   40 DO 60 L3 = 1, NT
      DO 50 L4 = 1, 3
      IF (C(L5, L4, L3) .EQ. 0.) GO TO 50
      SC(L2, L4, L3) = SC(L2, L4, L3) - SS(L2, L1, I1)*C(L5, L4, L3)
   50 CONTINUE
   60 CONTINUE
      RETURN
C
C
   70 DO 90 L3 = I1, NT
      IF = 1
      IF (L3 .EQ. I1) IF = L5 + 1
      IF (IF .GT. 3) GO TO 90
      DO 80 L4 = IF, 3
      IF (SS(L1, L4, L3) .EQ. 0.) GO TO 80
      SS(L2, L4, L3) = SS(L2, L4, L3) - SS(L2, L5, I1)*SS(L1, L4, L3)
   80 CONTINUE
   90 CONTINUE
      RETURN
C
C
  100 J1 = 3*IL - 3 + L2
      IM = (J1 + 2)/3
      DO 120 L3 = IM, NT
      DO 110 L4 = 1, 3
      IF (C(L5, L4, L3) .EQ. 0.) GO TO 110
      J2 = 3*L3 - 3 + L4
      IF (J2 .LT. J1) GO TO 110
      J3 = J2*(J2 - 1)/2 + J1
      KK(J3) = KK(J3) - SS(L2, L1, I1)*C(L5, L4, L3)
  110 CONTINUE
  120 CONTINUE
      RETURN
C
C
  130 DO 150 L3 = 1, NT
      DO 140 L4 = 1, 3
      IF (SC(L1, L4, L3) .EQ. 0.) GO TO 140
```

```
      SC(L2, L4, L3) = SC(L2, L4, L3) - SS(L2, L5, I1)*SC(L1, L4, L3)
  140 CONTINUE
  150 CONTINUE
      RETURN
      END
C     ******************************************************************
C     * THIS SUBROUTINE GENERATES STIFFNESS MATRIX FOR A RECTANGULAR   *
C     *           FINITE ELEMENT -- NON-CONFORMING ELEMENT             *
C     ******************************************************************
      SUBROUTINE PLSTIF(A, B, L, K)
      COMMON /LSLAB/M, N, NSP, NDP
      COMMON /TEV/T, E, V
      REAL*4 K(12, 12, NDP)
      C1 = (B/A)**2
      C2 = (A/B)**2
      C3 = E*T**3/(12.*(1. - V**2)*A*B)
      C4 = .2*(1. + 4.*V)
      C5 = .2*(14. - 4.*V)
      C6 = .2*(1. - V)
      K(3, 3, L) = (4.*(C1 + C2) + C5)*C3
      K(3, 1, L) = (2.*C2 + C4)*B*C3
      K(3, 2, L) = - (2.*C1 + C4)*A*C3
      K(6, 3, L) = (2.*(C1 - 2.*C2) - C5)*C3
      K(4, 3, L) = (2.*C2 + C6)*B*C3
      K(5, 3, L) = ( - C1 + C4)*A*C3
      K(9, 3, L) = ( - 2.*(C1 + C2) + C5)*C3
      K(7, 3, L) = (C2 - C6)*B*C3
      K(8, 3, L) = ( - C1 + C6)*A*C3
      K(12, 3, L) = ( - 2.*(2.*C1 - C2) - C5)*C3
      K(10, 3, L) = (C2 - C4)*B*C3
      K(11, 3, L) = - (2.*C1 + C6)*A*C3
      K(1, 1, L) = (C2 + C6)*B**2*C3*4./3.
      K(2, 1, L) = - V*A*B*C3
      K(6, 1, L) = - (2.*C2 + C6)*B*C3
      K(4, 1, L) = (C2 - C6/2.)*B**2*C3*2./3.
      K(5, 1, L) = 0.
      K(9, 1, L) = ( - C2 + C6)*B*C3
      K(7, 1, L) = (C2 + C6)*B**2*C3*1./3.
      K(8, 1, L) = 0.
      K(12, 1, L) = (C2 - C4)*B*C3
      K(10, 1, L) = (C2 - 2.*C6)*B**2*C3*2./3.
      K(11, 1, L) = 0.
      K(2, 2, L) = (C1 + C6)*A**2*C3*4./3.
      K(6, 2, L) = ( - C1 + C4)*A*C3
      K(4, 2, L) = 0.
      K(5, 2, L) = (C1 - 2.*C6)*A**2*C3*2./3.
      K(9, 2, L) = (C1 - C6)*A*C3
      K(7, 2, L) = 0.
      K(8, 2, L) = (C1 + C6)*A**2*C3*1./3.
      K(12, 2, L) = (2.*C1 + C6)*A*C3
      K(10, 2, L) = 0.
      K(11, 2, L) = (2*C1 - C6)*A**2*C3*1./3.
      K(6, 6, L) = K(3, 3, L)
      K(6, 4, L) = - K(3, 1, L)
      K(6, 5, L) = K(3, 2, L)
      K(9, 6, L) = K(12, 3, L)
      K(7, 6, L) = - K(10, 3, L)
      K(8, 6, L) = K(11, 3, L)
      K(12, 6, L) = K(9, 3, L)
      K(10, 6, L) = - K(7, 3, L)
      K(11, 6, L) = K(8, 3, L)
      K(4, 4, L) = K(1, 1, L)
      K(5, 4, L) = - K(2, 1, L)
      K(9, 4, L) = - K(12, 1, L)
      K(7, 4, L) = K(10, 1, L)
      K(8, 4, L) = 0.
```

```
      K(12, 4, L) = - K(9, 1, L)
      K(10, 4, L) = K(7, 1, L)
      K(11, 4, L) = 0.
      K(5, 5, L) = K(2, 2, L)
      K(9, 5, L) = K(12, 2, L)
      K(7, 5, L) = 0.
      K(8, 5, L) = K(11, 2, L)
      K(12, 5, L) = K(9, 2, L)
      K(10, 5, L) = 0.
      K(11, 5, L) = K(8, 2, L)
      K(9, 9, L) = K(3, 3, L)
      K(9, 7, L) = - K(3, 1, L)
      K(9, 8, L) = - K(3, 2, L)
      K(12, 9, L) = K(6, 3, L)
      K(10, 9, L) = - K(4, 3, L)
      K(11, 9, L) = - K(5, 3, L)
      K(7, 7, L) = K(1, 1, L)
      K(8, 7, L) = K(2, 1, L)
      K(12, 7, L) = - K(6, 1, L)
      K(10, 7, L) = K(4, 1, L)
      K(11, 7, L) = 0.
      K(8, 8, L) = K(2, 2, L)
      K(12, 8, L) = - K(6, 2, L)
      K(10, 8, L) = 0.
      K(11, 8, L) = K(5, 2, L)
      K(12, 12, L) = K(3, 3, L)
      K(12, 10, L) = K(3, 1, L)
      K(12, 11, L) = - K(3, 2, L)
      K(10, 10, L) = K(1, 1, L)
      K(11, 10, L) = - K(2, 1, L)
      K(11, 11, L) = K(2, 2, L)
      DO 20 I1 = 1, 11
      I3 = I1 + 1
      DO 10 I2 = I3, 12
      K(I1, I2, L) = K(I2, I1, L)
   10 CONTINUE
   20 CONTINUE
      RETURN
      END
C     ******************************************************************
C     * THIS SUBROUTINE GENERATES STIFFNESS MATRIX FOR A RECTANGULAR   *
C     *           FINITE ELEMENT -- CONFORMING ELEMENT                 *
C     ******************************************************************
      SUBROUTINE PLSTIC(A, B, L, K)
      COMMON /LSLAB/M, N, NSP, NDP
      COMMON /TEV/T, E, V
      REAL*4 K(12, 12, NDP)
      C1 = (B/A)**2
      C2 = (A/B)**2
      C3 = E*T**3/(12.*(1. - V**2)*A*B)
      C4 = 1. + 5.*V
      K(3, 3, L) = (156./35.*(C1 + C2) + 72./25.)*C3
      K(3, 1, L) = (22./35.*C1 + 78./35.*C2 + 6./25.*C4)*B*C3
      K(3, 2, L) = - (78./35.*C1 + 22./35.*C2 + 6./25.*C4)*A*C3
      K(6, 3, L) = (54./35.*C1 - 156./35.*C2 - 72./25.)*C3
      K(4, 3, L) = ( - 13./35.*C1 + 78./35.*C2 + 6./25.)*B*C3
      K(5, 3, L) = ( - 27./35.*C1 + 22./35.*C2 + 6./25.*C4)*A*C3
      K(9, 3, L) = ( - 54./35.*(C1 + C2) + 72./25.)*C3
      K(7, 3, L) = (13./35.*C1 + 27./35.*C2 - 6./25.)*B*C3
      K(8, 3, L) = ( - 27./35.*C1 - 13./35.*C2 + 6./25.)*A*C3
      K(12, 3, L) = ( - 156./35.*C1 + 54./35.*C2 - 72./25.)*C3
      K(10, 3, L) = ( - 22./35.*C1 + 27./35.*C2 - 6./25.*C4)*B*C3
      K(11, 3, L) = ( - 78./35.*C1 + 13./35.*C2 - 6./25.)*A*C3
      K(1, 1, L) = (4./35.*C1 + 52./35.*C2 + 8./25.)*B**2*C3
      K(2, 1, L) = - (11./35.*(C1 + C2) + 1./50.*(1. + 60.*V))*A*B*C3
      K(6, 1, L) = (13./35.*C1 - 78./35.*C2 - 6./25.)*B*C3
```

```
      K(4, 1, L) = ( - 3./35.*C1 + 26./35.*C2 - 2./25.)*B**2*C3
      K(5, 1, L) = ( - 13./70.*C1 + 11./35.*C2 + 1./50.*C4)*A*B*C3
      K(9, 1, L) = ( - 13./35.*C1 - 27./35.*C2 + 6./25.)*B*C3
      K(7, 1, L) = (3./35.*C1 + 9./35.*C2 + 2./25.)*B**2*C3
      K(8, 1, L) = ( - 13./70.*(C1 + C2) + 1./50.)*A*B*C3
      K(12, 1, L) = ( - 22./35.*C1 + 27./35.*C2 - 6./25.*C4)*B*C3
      K(10, 1, L) = ( - 4./35.*C1 + 18./35.*C2 - 8./25.)*B**2*C3
      K(11, 1, L) = ( - 11./35.*C1 + 13./70.*C2 - 1./50.*C4)*A*B*C3
      K(2, 2, L) = (52./35.*C1 + 4./35.*C2 + 8./25.)*A**2*C3
      K(6, 2, L) = ( - 27./35.*C1 + 22./35.*C2 + 6./25.*C4)*A*C3
      K(4, 2, L) = (13./70.*C1 - 11./35.*C2 - 1./50.*C4)*A*B*C3
      K(5, 2, L) = (18./35.*C1 - 4./35.*C2 - 8./25.)*A**2*C3
      K(9, 2, L) = (27./35.*C1 + 13./35.*C2 - 6./25.)*A*C3
      K(7, 2, L) = (13./70.*(C1 + C2) - 1./50.)*A*B*C3
      K(8, 2, L) = (9./35.*C1 + 3./35.*C2 + 2./25.)*A**2*C3
      K(12, 2, L) = (78./35.*C1 - 13./35.*C2 + 6./25.)*A*C3
      K(10, 2, L) = (11./35.*C1 - 13./70.*C2 + 1./50.*C4)*A*B*C3
      K(11, 2, L) = (26./35.*C1 - 3./35.*C2 - 2./25.)*A**2*C3
      K(6, 6, L) = K(3, 3, L)
      K(6, 4, L) = - K(3, 1, L)
      K(6, 5, L) = K(3, 2, L)
      K(9, 6, L) = K(12, 3, L)
      K(7, 6, L) = - K(10, 3, L)
      K(8, 6, L) = K(11, 3, L)
      K(12, 6, L) = K(9, 3, L)
      K(10, 6, L) = - K(7, 3, L)
      K(11, 6, L) = K(8, 3, L)
      K(4, 4, L) = K(1, 1, L)
      K(5, 4, L) = - K(2, 1, L)
      K(9, 4, L) = - K(12, 1, L)
      K(7, 4, L) = K(10, 1, L)
      K(8, 4, L) = - K(11, 1, L)
      K(12, 4, L) = - K(9, 1, L)
      K(10, 4, L) = K(7, 1, L)
      K(11, 4, L) = - K(8, 1, L)
      K(5, 5, L) = K(2, 2, L)
      K(9, 5, L) = K(12, 2, L)
      K(7, 5, L) = - K(10, 2, L)
      K(8, 5, L) = K(11, 2, L)
      K(12, 5, L) = K(9, 2, L)
      K(10, 5, L) = K(7, 2, L)
      K(11, 5, L) = K(8, 2, L)
      K(9, 9, L) = K(3, 3, L)
      K(9, 7, L) = - K(3, 1, L)
      K(9, 8, L) = - K(3, 2, L)
      K(12, 9, L) = K(6, 3, L)
      K(10, 9, L) = - K(4, 3, L)
      K(11, 9, L) = - K(5, 3, L)
      K(7, 7, L) = K(1, 1, L)
      K(8, 7, L) = K(2, 1, L)
      K(12, 7, L) = - K(6, 1, L)
      K(10, 7, L) = K(4, 1, L)
      K(11, 7, L) = K(5, 1, L)
      K(8, 8, L) = K(2, 2, L)
      K(12, 8, L) = - K(6, 2, L)
      K(10, 8, L) = K(4, 2, L)
      K(11, 8, L) = K(5, 2, L)
      K(12, 12, L) = K(3, 3, L)
      K(12, 10, L) = K(3, 1, L)
      K(12, 11, L) = - K(3, 2, L)
      K(10, 10, L) = K(1, 1, L)
      K(11, 10, L) = - K(2, 1, L)
      K(11, 11, L) = K(2, 2, L)
      DO 20 I1 = 1, 11
      I3 = I1 + 1
      DO 10 I2 = I3, 12
      K(I1, I2, L) = K(I2, I1, L)
   10 CONTINUE
   20 CONTINUE
      RETURN
      END
C     ******************************************************************
C     *               THIS SUBROUTINE PRINTS ERROR MESSAGES            *
C     ******************************************************************
      SUBROUTINE ERROR(BB, NNN, WT)
      INTEGER BB, WT
      WRITE (WT,10) BB, NNN
      STOP
   10 FORMAT (11X, 'ERROR IN INPUT, CHANGE ', A4, ' TO', I6)
      END
C     ******************************************************************
C     * THIS SUBROUTINE INCORPORATES THE FLAT PLATE FLOOR STIFFNESS    *
C     * MATRIX WITH THE STIFFNESS MATRICES FROM ALL STRUCTURAL MEMBERS *
C     *         TO FORM THE BUILDING LATERAL STIFFNESS MATRIX          *
C     ******************************************************************
      SUBROUTINE FORM(SD, CLN, KK, CP, LP, BP, FEF, LB, LDB, LC, S, R,
     &NST, NB, NC, NCP, NBP, NFEF, NPB, IFEF, PP, NN, LT, TP, NTRU, NPP)
      INTEGER RD, WT
      INTEGER*2 INPT(80), B1/'C'/
      COMMON /JRW/RD, WT
      COMMON /INPUT/INPT, FN(8)
      COMMON /MDIM/NT1, NT2, MT4, MT5, NN3
      COMMON /LSLAB/M1, N1, JK(4)
      DIMENSION SD(NST, 18), CLN(NC, 2), CP(9, NCP), BP(9, NBP), FEF(7,
     &NFEF), LB(NST, NB, 3), LDB(NST, NB, 3), LC(NST, NC, 1), LP(NST,
     &NPB, 3), S(1), IFEF(NFEF), PP(5, NPP), LT(3, NTRU), TP(2, NTRU)
      REAL*4 KK(NT1)
      COMMON /JUNK/NLD, N, ND, MM, MN, LM(12), P(12, 3), TT(8)
      COMMON /STIF/ASA(12, 12), SA(8, 12), T(8, 12)
   10 NLD = 3
C
C     READ AND PRINT OF COLUMN PROPERTIES
C
   20 L = NCP - 1
      WRITE (WT,610)
      IF (L .EQ. 0) GO TO 50
      DO 40 I = 1, L
      READ (RD,600) INPT
      CALL CNVRT(1, 8)
      DO 30 J = 1, 7
   30 CP(J, I) = FN(J + 1)
      READ (RD,600) INPT
      CALL CNVRT(1, 2)
      CP(8, I) = FN(1)
      CP(9, I) = FN(2)
   40 CONTINUE
      WRITE (WT,620) (I, (CP(J, I), J = 1, 9), I = 1, L)
   50 DO 60 I = 1, 9
   60 CP(I, NCP) = 0.
C
C     READ AND PRINT OF BEAM PROPERTIES
C
      L = NBP - 1
      IF (L .EQ. 0) GO TO 90
      WRITE (WT,630)
      DO 80 I = 1, L
      READ (RD,600) INPT
      CALL CNVRT(1, 8)
      DO 70 J = 1, 7
   70 BP(J, I) = FN(J + 1)
      IF (BP(5, I) .EQ. 0.) BP(5, I) = 4.
      IF (BP(6, I) .EQ. 0.) BP(6, I) = 4.
```

```
      IF (BP(7, I) .EQ. 0.) BP(7, I) = 2.
      READ (RD,600) INPT
      CALL CNVRT(1, 2)
      BP(8, I) = FN(1)
      BP(9, I) = FN(2)
   80 CONTINUE
      WRITE (WT,640) (I, (BP(J, I), J = 1, 9), I = 1, L)
   90 DO 100 I = 1, 9
  100 BP(I, NBP) = 0.
C
C     READ AND PRINT OF WALL PANEL PROPERTIES
C
      L = NPP - 1
      IF (L .EQ. 0) GO TO 130
      WRITE (WT,650)
      DO 120 I = 1, L
      READ (RD,600) INPT
      CALL CNVRT(1, 6)
      DO 110 J = 1, 5
  110 PP(J, I) = FN(J + 1)
  120 CONTINUE
      WRITE (WT,660) (I, (PP(J, I), J = 1, 5), I = 1, L)
  130 DO 140 I = 1, 5
  140 PP(I, NPP) = 0.
C
C     READ FIXED END BEAM LOADS
C
      IF (NFEF .EQ. 0) GO TO 180
      DO 170 I = 1, NFEF
      READ (RD,600) INPT
      CALL CNVRT(1, 7)
      IFEF(I) = 0
      DO 150 J = 11, 20
      IF (INPT(J) .EQ. B1) IFEF(I) = 1
  150 CONTINUE
      DO 160 J = 1, 5
  160 FEF(J, I) = FN(J + 2)
  170 CONTINUE
C
C     READ (OR GENERATE) AND PRINT OF BEAM LOCARION CARDS - LB
C
  180 IF (NB .EQ. 0) GO TO 260
  190 WRITE (WT,700)
      DO 230 M = 1, NB
      K = 0
      DO 220 N = 1, NST
      IF (K .NE. 0) GO TO 200
      LN = NST + 1 - N
      READ (RD,600) INPT
      CALL CNVRT(1, 8)
      I = FN(1)
      LB(N, M, 1) = FN(2)
      LB(N, M, 2) = FN(3)
      LB(N, M, 3) = FN(4)
      K = FN(5)
      LDB(N, M, 1) = FN(6)
      LDB(N, M, 2) = FN(7)
      LDB(N, M, 3) = FN(8)
      IF (LB(N, M, NLD) .EQ. 0) LB(N, M, NLD) = NBP
      GO TO 220
  200 K = K - 1
      DO 210 L = 1, NLD
      LB(N, M, L) = LB(N - 1, M, L)
  210 LDB(N, M, L) = LDB(N - 1, M, L)
  220 CONTINUE
  230 CONTINUE
      DO 250 N = 1, NST
      LN = NST + 1 - N
      DO 240 M = 1, NB
      IF (LB(N, M, 3) .EQ. NBP) GO TO 240
      WRITE (WT,690) LN, LB(N, M, 3), LB(N, M, 1), LB(N, M, 2), (LDB(N,
     &M, L), L = 1, 3)
  240 CONTINUE
  250 WRITE (WT,760)
C
C     READ (OR GENERATE) AND PRINT COLUMN LOCATIONS -LC
C
  260 WRITE (6,720)
      DO 290 M = 1, NC
      K = 0
      DO 280 N = 1, NST
      IF (K .NE. 0) GO TO 270
      LN = NST + 1 - N
      READ (RD,600) INPT
      CALL CNVRT(1, 3)
      KL = FN(1)
      LC(N, M, 1) = FN(2)
      K = FN(3)
      IF (LC(N, M, 1) .EQ. 0) LC(N, M, 1) = NCP
      GO TO 280
  270 K = K - 1
      LC(N, M, 1) = LC(N - 1, M, 1)
  280 CONTINUE
  290 CONTINUE
      DO 310 N = 1, NST
      LN = NST + 1 - N
      DO 300 M = 1, NC
      IF (LC(N, M, 1) .EQ. NCP) GO TO 300
      WRITE (WT,710) LN, M, LC(N, M, 1)
  300 CONTINUE
  310 WRITE (WT,760)
C
C     READ AND PRINT OF PANEL CARDS - LP,PP
C
      IF (NPB .EQ. 0) GO TO 380
      WRITE (WT,740)
      DO 350 M = 1, NPB
      K = 0
      DO 340 N = 1, NST
      IF (K .NE. 0) GO TO 320
      LN = NST + 1 - N
      READ (RD,600) INPT
      CALL CNVRT(1, 5)
      I = FN(1)
      LP(N, M, 1) = FN(2)
      LP(N, M, 2) = FN(3)
      LP(N, M, 3) = FN(4)
      K = FN(5)
      GO TO 340
  320 K = K - 1
      DO 330 L = 1, 3
  330 LP(N, M, L) = LP(N - 1, M, L)
  340 CONTINUE
  350 CONTINUE
      DO 370 N = 1, NST
      LN = NST + 1 - N
      DO 360 M = 1, NPB
      IF (LP(N, M, 3) .EQ. 0) GO TO 360
      WRITE (WT,730) LN, LP(N, M, 3), LP(N, M, 1), LP(N, M, 2)
  360 CONTINUE
  370 WRITE (WT,760)
C
```

```
C     READ AND PRINT DIAGONALS
C
  380 IF (NTRU .EQ. 0) GO TO 400
      DO 390 I = 1, NTRU
      READ (RD,600) INPT
      CALL CNVRT(1, 5)
      LT(1, I) = FN(1)
      LT(2, I) = FN(2)
      LT(3, I) = FN(3)
      TP(1, I) = FN(4)
      TP(2, I) = FN(5)
  390 CONTINUE
      WRITE (WT,750) (LT(1, I), LT(2, I), LT(3, I), TP(1, I), TP(2, I),
     &I = 1, NTRU)
C
C     PRINT LOADING INFORMATION
C
  400 WRITE (WT,790)
      WRITE (WT,770)
      DO 410 N = 1, NST
      LN = NST + 1 - N
      WRITE (WT,780) LN, SD(N, 9), SD(N, 10), SD(N, 12), SD(N, 13), SD(N
     &, 15), SD(N, 16), SD(N, 17), SD(N, 18)
  410 CONTINUE
      IF (NFEF .EQ. 0) GO TO 420
      WRITE (WT,670)
      WRITE (WT,680) (I, (FEF(J, I), J = 1, 5), I = 1, NFEF)
C
C     STORY BY STORY FORMATION AND REDUCTION OF STIFFNESS MATRIX
C
  420 CONTINUE
      MM = 3*NC
      MN = 6*NC
      NN = MN + 3*NST + 3
      CALL STIFF(1, S, R, NN)
C
      DO 580 N = 1, NST
      CALL STIFF(2, S, R, NN)
C
C     1. FORM COLUMN MATRICES
C
      DO 430 I = 1, 12
      DO 430 L = 1, NLD
  430 P(I, L) = 0.
      NF = 6
      ND = 12
      DO 440 M = 1, NC
      MC = LC(N, M, 1)
      IF (MC .EQ. NCP) GO TO 440
      XLL = SD(N, 2)
      CSA = 0.
      SNA = 1.
      XL1 = CLN(M, 1)
      YL1 = CLN(M, 2)
      CALL COLUMN(1, MC, XLL, SNA, CSA, NCP, XL1, YL1, CP)
      LM(6) = 3*M
      LM(5) = LM(6) - 1
      LM(4) = LM(5) - 1
      LM(11) = LM(5) + MM
      LM(10) = LM(4) + MM
      LM(12) = LM(6) + MM
      LM(3) = MN + 3*N
      LM(2) = LM(3) - 1
      LM(1) = LM(2) - 1
      LM(9) = LM(3) + 3
      LM(8) = LM(9) - 1
      LM(7) = LM(8) - 1
      CALL STIFF(5, S, R, NN)
  440 CONTINUE
C
C     2. FORM BEAM MATRICES
C
      IF (NB .EQ. 0) GO TO 500
      NF = 3
      ND = 6
      DO 490 M = 1, NB
      MB = LB(N, M, 3)
      IF (MB .EQ. NBP) GO TO 490
      KI = LB(N, M, 1)
      KJ = LB(N, M, 2)
      B1 = CLN(KJ, 1) - CLN(KI, 1)
      B2 = CLN(KJ, 2) - CLN(KI, 2)
      B3 = B1*B1 + B2*B2
      B3 = SQRT(B3)
      CSA = B1/B3
      SNA = B2/B3
      XLL = B3
      XLR = XLL - BP(8, MB) - BP(9, MB)
      DO 480 L = 1, NLD
      DO 450 I = 1, 6
  450 P(I, L) = 0.
      J = LDB(N, M, L)
      IF (J) 480, 480, 460
  460 V = FEF(5, J)*XLR/2.
      YM = V*XLR/6.
      P(1, L) = - FEF(1, J) - YM
      P(2, L) = - FEF(2, J) - V
      P(3, L) = FEF(3, J) + YM
      P(4, L) = - FEF(4, J) - V
      IF (IFEF(J)) 480, 470, 480
  470 P(1, L) = P(1, L) + P(2, L)*BP(8, MB)
      P(3, L) = P(3, L) - P(4, L)*BP(9, MB)
  480 CONTINUE
C
      CALL BEAM(1, MB, XLL, SNA, CSA, NBP, BP)
C
      LM(3) = 3*KI
      LM(2) = LM(3) - 1
      LM(1) = LM(2) - 1
      LM(6) = 3*KJ
      LM(5) = LM(6) - 1
      LM(4) = LM(5) - 1
      CALL STIFF(5, S, R, NN)
  490 CONTINUE
C
C     FORM PANEL STIFFNESS
C
  500 IF (NPB .EQ. 0) GO TO 530
      DO 510 I = 1, 10
      DO 510 J = 1, 3
  510 P(I, J) = 0.
      NF = 3
      ND = 10
      DO 520 M = 1, NPB
      MB = LP(N, M, 3)
      IF (MB .EQ. 0) GO TO 520
      XLL = SD(N, 2)
      KI = LP(N, M, 1)
      KJ = LP(N, M, 2)
      B1 = CLN(KJ, 1) - CLN(KI, 1)
      B2 = CLN(KJ, 2) - CLN(KI, 2)
      B3 = B1*B1 + B2*B2
```

```
      B3 = SQRT(B3)
      CSA = B1/B3
      SNA = B2/B3
      D = B3
      AR = CLN(KI, 1)*CLN(KJ, 2) - CLN(KJ, 1)*CLN(KI, 2)
      DF = AR/B3
      CALL PANEL(1, MB, XLL, D, DF, SNA, CSA, NPP, PP)
C
      LM(3) = MN + 3*N
      LM(2) = LM(3) - 1
      LM(1) = LM(2) - 1
      LM(4) = 3*KI
      LM(5) = 3*KJ
      LM(8) = MN + 3*N + 3
      LM(7) = LM(8) - 1
      LM(6) = LM(7) - 1
      LM(9) = MM + 3*KI
      LM(10) = MM + 3*KJ
      CALL STIFF(5, S, R, NN)
  520 CONTINUE
C
C     FORM DIAGONAL STIFFNESS
C
  530 IF (NTRU .EQ. 0) GO TO 560
      DO 540 I = 1, 8
      DO 540 J = 1, NLD
  540 P(I, J) = 0.
      NF = 1
      ND = 8
      DO 550 L = 1, NTRU
      NP = NST - LT(1, L) + 1
      IF (NP .NE. N) GO TO 550
      YLL = SD(N, 2)
      KJ = LT(2, L)
      KI = LT(3, L)
      B1 = CLN(KJ, 1) - CLN(KI, 1)
      B2 = CLN(KJ, 2) - CLN(KI, 2)
      B3 = B1*B1 + B2*B2
      B3 = SQRT(B3)
      CSA = B1/B3
      SNA = B2/B3
      VANG = ATAN(YLL/B3)
      DL = SQRT(B3*B3 + YLL*YLL)
      AR = CLN(KI, 1)*CLN(KJ, 2) - CLN(KJ, 1)*CLN(KI, 2)
      D1 = AR/B3
      CALL DIAG(1, L, VANG, DL, SNA, CSA, NTRU, D1, TP)
C
      LM(3) = MN + 3*N
      LM(2) = LM(3) - 1
      LM(1) = LM(2) - 1
      LM(4) = 3*KJ
      LM(7) = MN + 3*N + 3
      LM(6) = LM(7) - 1
      LM(5) = LM(6) - 1
      LM(8) = MM + 3*KI
      CALL STIFF(5, S, R, NN)
  550 CONTINUE
C
C     INSERT FLOOR STIFFNESS TO FRAME SATIFFNESS
  560 DO 570 I = 1, NT1
      S(I) = S(I) + KK(I)
  570 CONTINUE
C
C     REDUCE STIFFNESS MATRIX FOR LEVEL N
C
      CALL STIFF(3, S, R, NN)
C
  580 CONTINUE
C
C     WRITE LATERAL STIFFNESS ON TAPE
C
  590 N = 1
      CALL STIFF(4, S, R, NN)
      WRITE (3) CLN, CP, BP, LC, PP
      IF (NB .NE. 0) WRITE (3) LB, LDB
      IF (NFEF .NE. 0) WRITE (3) FEF, IFEF
      IF (NPB .NE. 0) WRITE (3) LP
      IF (NTRU .NE. 0) WRITE (3) LT, TP
      WRITE (3) NC, NCP, NBP, NFEF, NPP, NTRU, NB, NPB
C
      RETURN
  600 FORMAT (80A1)
  610 FORMAT (////' FRAME DATA', ////' COLUMN PROPERTIES')
  620 FORMAT (/17X, 'ELASTIC', 2X, 'CROSS-SECT', 2X, '      SHEAR', 2X, '
     &      SHEAR', 3X, 'TORSIONAL', 7X, 'MOMENT OF INERTIA', 3X, 'DEPTH
     &OF FINITE JOINT', /' COLUMN TYPE', 5X, 'MODULUS', 8X, 'AREA', 6X,
     &'AREA Y', 6X, 'AREA X', 5X, 'INERTIA', 9X, 'X-X', 9X, 'Y-Y', 5X, '
     &COL TOP', 5X, 'COL BTM', /(I10, 2X, 9F12.2))
  630 FORMAT (///' BEAM PROPERTIES')
  640 FORMAT (/17X, 'ELASTIC', 7X, 'SHEAR', 3X, 'TORSIONAL', 4X, 'FLEXUR
     &AL', 8X, ' BEAM STIFFNESS COEFICIENTS ', 3X, 'WIDTH OF FINITE JOIN
     &T', /'    BEAM TYPE', 5X, 'MODULUS', 8X, 'AREA', 5X, 'INERTIA', 5X,
     &'INERTIA', 9X, 'KII', 9X, 'KJJ', 5X, 'KIJ=KJI', 7X, 'END I', 7X, '
     &END J'/(I10, 2X, 9F12.2))
  650 FORMAT (///' WALL PANEL PROPERTIES')
  660 FORMAT (/17X, 'ELASTIC', 2X, 'CROSS-SECT', 6X, 'MOMENT', 7X, 'SHEA
     &R', 4X, 'SHEARING', /'    WALL TYPE', 5X, 'MODULUS', 8X, 'AREA', 5X
     &, 'INERTIA', 8X, 'AREA', 5X, 'MODULUS', /(I10, 2X, 5F12.2))
  670 FORMAT (///' BEAM GRAVITY LOADING INFORMATION'/' (SEE BEAM LOCATIO
     &N TABLE FOR LOADING LOCATIONS)')
  680 FORMAT (/20X, 'FIXED END FORCES-END I', 8X, 'FIXED END FORCES-END
     &J', 6X, 'UNIFORMLY'/'    LOAD TYPE', 9X, 'MOMENT', 10X, 'SHEAR', 9X
     &, 'MOMENT', 10X, 'SHEAR', 5X, 'DIST. LOAD'/(I10, 2X, 5F15.3))
  690 FORMAT (4I10, 3X, 3I10)
  700 FORMAT (///15H BEAM LOCATIONS//46X, ' GRAVITY LOADING TYPE(SEE BEL
     &OW)', /5X, 'LEVEL', 3X, 'BM TYPE', 5X, 'COL I', 5X, 'COL J', 10X,
     &'  I', 7X, ' II', 7X, 'III')
  710 FORMAT (3I10)
  720 FORMAT (//17H COLUMN LOCATIONS/30H0     LEVEL     COLUMN  COL TYPE)
  730 FORMAT (4I10)
  740 FORMAT (//21H WALL PANEL LOCATIONS/40H0     LEVEL WALL TYPE     COL
     & I     COL J)
  750 FORMAT (//42H BRACING ELEMENT LOCATIONS AND PROPERTIES  //38X, 'EL
     &ASTIC', 5X, 'CROSS-SECT', /30H     LEVEL   TOP COL   BTM COL, 8X,
     &'MODULUS', 5X, '      AREA', /(3I10, 2F15.2))
  760 FORMAT (1H  )
  770 FORMAT (///37H STORY LATERAL LOADS ...CASES A AND B//19X, 'LATERAL
     & LOAD CASE A', 7X, 'LATERAL LOAD CASE B', 12X, 'LOCATION OF POINT
     &OF APPLICATION'/' LEVEL NO', 13X, 'F-X', 10X, 'F-Y', 10X, 'F-X',
     &10X, 'F-Y', 12X, 'XA', 8X, 'YA', 8X, 'XB', 8X, 'YB')
  780 FORMAT (I8, 4X, 4F13.2, 4X, 4F10.2)
  790 FORMAT ('1', 43X, '........ LOADING INFORMATION ........'////' TWO
     & LATERAL LOADING CASES DESIGNATED A AND B,'/' THREE GRAVITY LOADIN
     &G CASES DESIGNATED I, II AND III, PERMITTED')
      END
C     ******************************************************************
C     * THIS SUBROUTINE GENERATES TRANSFORMATION MATRICES FOR VARIOUS  *
C     *                    STRUCTURAL ELEMENTS                          *
C     ******************************************************************
      SUBROUTINE TRANS(KK, XLP, A, B, SNA, CSA, XL1, YL1, XL2, YL2)
      COMMON /STIF/ASA(12, 12), SA(8, 12), T(8, 12)
      COMMON /JUNK/NLD, N, ND, MM, MN, LM(12), P(12, 3), TF(6, 12), TT(
```

```
      &12, 12)
C
C     FORM TRANSFORMATION MATRICES
C
      DO 10 I = 1, 6
      DO 10 J = 1, 12
   10 TP(I, J) = 0.
C
      GO TO (20, 90, 130, 190, 130), KK
C
C     COLUMN TRANSFORMATION
C
   20 D1 = 1./XLR
      TP(2, 2) = D1
      TP(3, 2) = D1
      TP(2, 8) = - D1
      TP(3, 8) = - D1
      TP(5, 1) = - D1
      TP(6, 1) = - D1
      TP(5, 7) = D1
      TP(6, 7) = D1
      TP(1, 3) = 1.
      TP(4, 6) = 1.
      TP(1, 9) = - 1.
      TP(4, 12) = - 1.
      TP(3, 4) = A/XLP
      TP(6, 5) = A/XLP
      TP(2, 4) = 1. + A/XLR
      TP(5, 5) = 1. + A/XLR
      TP(2, 10) = B/XLR
      TP(5, 11) = B/XLR
      TP(3, 10) = 1. + B/XLR
      TP(6, 11) = 1. + B/XLR
      DO 30 I = 1, 12
      DO 30 J = 1, 12
   30 TT(I, J) = 0.
      TT(1, 1) = SNA
      TT(2, 2) = SNA
      TT(1, 2) = - CSA
      TT(2, 1) = CSA
      TT(3, 3) = 1.
      DO 50 II = 3, 9, 3
      DO 40 I = 1, 3
      DO 40 J = 1, 3
   40 TT(II + I, II + J) = TT(I, J)
   50 CONTINUE
      TT(1, 3) = - YL1*SNA - XL1*CSA
      TT(2, 3) = - YL1*CSA + XL1*SNA
      TT(7, 9) = TT(1, 3)
      TT(8, 9) = TT(2, 3)
      DO 80 I = 1, 6
      DO 80 I1 = 1, 10, 3
      I2 = I1 + 2
      DO 70 J = I1, I2
      XX = 0.
      DO 60 K = I1, I2
   60 XX = XX + TP(I, K)*TT(K, J)
   70 T(I, J) = XX
   80 CONTINUE
      RETURN
C
C     BEAM LOCAL TRANSFORMATION
   90 D1 = - 1./XLR
      TP(1, 2) = - 1.
      TP(1, 5) = 1.
      TP(2, 1) = 1. + A/XLR
      TP(3, 1) = A/XLR
      TP(2, 4) = B/XLR
      TP(3, 4) = 1. + B/XLR
      TP(2, 3) = - D1
      TP(3, 3) = - D1
      TP(2, 6) = D1
      TP(3, 6) = D1
      DO 100 I = 1, 3
      DO 100 J = 1, 3
  100 TT(I, J) = 0.
      TT(1, 1) = SNA
      TT(2, 2) = SNA
      TT(1, 2) = - CSA
      TT(2, 1) = CSA
      TT(3, 3) = 1.
      DO 120 J = 1, 3
      DO 120 II = 1, 4, 3
      I2 = II - 1
      DO 120 J1 = 1, 3
      XX = 0.
      DO 110 K = 1, 3
  110 XX = XX + TP(J, I2 + K)*TT(K, J1)
      I = J1 + I2
  120 T(J, I) = XX
      RETURN
C
C     PANEL TRANSFORMATION
C
  130 D1 = 1./XLR
      D2 = 1./A
C
      IF (KK .EQ. 5) GO TO 140
C
      TP(1, 4) = - D1
      TP(2, 4) = - D1
      TP(1, 1) = D1
      TP(2, 1) = D1
      TP(2, 5) = - D2
      TP(2, 6) = D2
      TP(1, 2) = - D2
      TP(1, 3) = D2
      TP(3, 5) = - 0.5
      TP(3, 6) = - 0.5
      TP(3, 2) = 0.5
      TP(3, 3) = 0.5
C
      KN = 3
      GO TO 150
C
  140 TP(1, 1) = D1
      TP(1, 4) = - D1
      TP(1, 2) = - D2/2.
      TP(1, 6) = - D2/2.
      TP(1, 3) = D2/2.
      TP(1, 5) = D2/2.
C
      KN = 1
C
  150 DO 160 I = 1, 3
      DO 160 J = 1, 5
  160 TT(I, J) = 0.
      TT(1, 1) = CSA
      TT(1, 2) = SNA
      TT(1, 3) = B
      TT(2, 4) = 1.
      TT(3, 5) = 1.
```

```
C
      DO 180 J = 1, KN
      DO 180 II = 1, 4, 3
      I2 = II - 1
      JJ = 0
      IF (II .EO. 4) JJ = 5
      DO 180 J1 = 1, 5
      XX = 0.
      DO 170 K = 1, 3
  170 XX = XX + TF(J, I2 + K)*TT(K, J1)
      I = J1 + JJ
  180 T(J, I) = XX
      RETURN
C
C     DIAGONAL TRANSFORMATION
  190 T(1, 1) = COS(XLR)*CSA
      T(1, 2) = COS(XLR)*SNA
      T(1, 3) = COS(XLR)*XL1
      T(1, 4) = SIN(XLR)
      T(1, 5) = - T(1, 1)
      T(1, 6) = - T(1, 2)
      T(1, 7) = - T(1, 3)
      T(1, 8) = - T(1, 4)
      RETURN
      END
C     *****************************************************************
C     *        THIS SUBROUTINE FORMS PANEL STIFFNESS MATRIX           *
C     *****************************************************************
      SUBROUTINE PANEL(IO, L, XL, D, DF, SNA, CSA, NPP, PP)
      DIMENSION PP(5, NPP)
      COMMON /STIF/ASA(12, 12), SA(8, 12), T(8, 12)
      COMMON /JUNK/NLD, N, ND, MM, MN, LM(12), P(12, 3), TT(8)
C
C     PANEL STIFFNESS AND FORCE MATRICES
C
      NP = 3
C
C     CHECK PANEL TYPE (DEFAULT TO PURE SHEAR PANEL IF FLEX I=ZERO)
C
      IF (PP(3, L) .LE. 0.) NP = 5
C
C     TRANSFORMATION MATRIX
C
      CALL TRANS(NP, XL, D, DF, SNA, CSA, 0., 0., 0., 0.)
C
      IF (NP .EO. 5) GO TO 20
C
C     FLEXURAL PANEL STIFFNESS (NP=3)
C
      E = PP(1, L)
      SHFY = 0.
      EIX = E*PP(3, L)/(XL*XL)
      IF (PP(4, L) .GT. 0.) SHFY = 6.*EIX/(PP(4, L)*PP(5, L))
      COMMX = (EIX*XL*2.)/(1. + 2.*SHFY)
C
      S1 = COMMX*2.*(1. + 0.5*SHFY)
      S2 = COMMX*(1. - SHFY)
      S3 = E*PP(2, L)/XL
C
C     FORCE-FRAME DISPLACEMENT MATRIX
C
      DO 10 I = 1, 10
      SA(1, I) = S1*T(1, I) + S2*T(2, I)
      SA(2, I) = S2*T(1, I) + S1*T(2, I)
      SA(3, I) = S3*T(3, I)
   10 SA(4, I) = (SA(1, I) + SA(2, I))/XL
C
      KK = 3
      GO TO 40
C
C     SHEAR PANEL STIFFNESS (NP=5)
C
   20 IF (IO .EO. 2) XL = 1.
      S1 = PP(2, L)*PP(5, L)*XL
C
C     FORCE-FRAME DISPLACEMENT MATRIX
C
      DO 30 I = 1, 10
      SA(1, I) = S1*T(1, I)
   30 SA(2, I) = SA(1, I)/PP(2, L)
C
      KK = 1
C
   40 IF (IO .EQ. 2) RETURN
C
C     STIFFNESS MATRIX
      DO 60 I = 1, 10
      DO 60 J = 1, 10
      ASA(I, J) = 0.
      DO 50 K = 1, KK
   50 ASA(I, J) = ASA(I, J) + T(K, I)*SA(K, J)
   60 ASA(J, I) = ASA(I, J)
      RETURN
      END
C     *****************************************************************
C     *        THIS SUBROUTINE FORMS COLUMN STIFFNESS MATRIX          *
C     *****************************************************************
      SUBROUTINE COLUMN(IO, MC, XL, SNA, CSA, NCP, XL1, YL1, CP)
      DIMENSION CP(9, NCP)
      COMMON /STIF/ASA(12, 12), SA(8, 12), T(8, 12)
      COMMON /JUNK/NLD, N, ND, MM, MN, LM(12), P(12, 3), TT(8)
C
C     COLUMN STIFFNESS AND FORCE MATRICES
C
C     TRANSFORMATION MATRICES
      DT = CP(8, MC)
      DB = CP(9, MC)
      XLR = XL - DT - DB
      CALL TRANS(1, XLR, DT, DB, SNA, CSA, XL1, YL1, 0., 0.)
C
C     COLUMN MEMBER STIFFNESS
      E = CP(1, MC)
      AZ = CP(2, MC)
      AY = CP(3, MC)
      AX = CP(4, MC)
      AAZ = CP(5, MC)
      AAY = CP(6, MC)
      AAX = CP(7, MC)
      SHFX = 0.
      SHFY = 0.
      EIY = AAY*E/(XLR*XLR)
      EIX = AAX*E/(XLR*XLR)
      IF (AY .GT. 0.) SHFY = 14.4*EIY/(AY*E)
      IF (AX .GT. 0.) SHFX = 14.4*EIX/(AX*E)
      COMMY = 2.*EIX*XLR/(1. + 2.*SHFX)
      COMMX = 2.*EIY*XLR/(1. + 2.*SHFY)
C
      S1 = 0.414*E*AAZ/XLR
      S2 = COMMX*(2. + SHFY)
      S3 = COMMX*(1. - SHFY)
      S4 = E*AZ/XLR
      S5 = COMMY*(2. + SHFX)
```

```
      S6 = COMMY*(1. - SHFX)
C
C     FORCE-FRAME DISPLACEMENT MATRIX
      DO 10 I = 1, 12
      SA(1, I) = S1*T(1, I)
      SA(2, I) = S2*T(2, I) + S3*T(3, I)
      SA(3, I) = S3*T(2, I) + S2*T(3, I)
      SA(4, I) = S4*T(4, I)
      SA(5, I) = S5*T(5, I) + S6*T(6, I)
      SA(6, I) = S6*T(5, I) + S5*T(6, I)
      SA(7, I) = (SA(2, I) + SA(3, I))/XLR
   10 SA(8, I) = (SA(5, I) + SA(6, I))/XLR
C
   20 IF (IO .EQ. 2) RETURN
C
C     STIFFNESS MATRIX ASA
      DO 40 I = 1, 12
      DO 40 J = 1, 12
      ASA(I, J) = 0.
      DO 30 K = 1, 6
   30 ASA(I, J) = ASA(I, J) + T(K, I)*SA(K, J)
   40 ASA(J, I) = ASA(I, J)
      RETURN
      END
C     ****************************************************************
C     *          THIS SUBROUTINE FORMS BEAM STIFFNESS MATRIX         *
C     ****************************************************************
      SUBROUTINE BEAM(IO, MB, XL, SNA, CSA, NBP, BP)
      DIMENSION BP(9, NBP)
C
      COMMON /STIF/ASA(12, 12), SA(8, 12), T(8, 12)
      COMMON /JUNK/NLD, N, ND, MM, MN, LM(12), P(12, 3), TX(8)
C
C     BEAM STIFFNESS AND FORCE MATRICES
C
C     TRANSFORMATION MATRIX
      WI = BP(8, MB)
      WJ = BP(9, MB)
      XLP = XL - WI - WJ
      CALL TRANS(2, XLR, WI, WJ, SNA, CSA, 0., 0., 0., 0.)
C
C     MEMBER STIFFNESS
      E = BP(1, MB)
      SHFZ = 0.
      EIX = BP(4, MB)*E/(XLR*XLR)
      IF (BP(2, MB) .GT. 0.) SHFZ = 14.4*EIX/(BP(2, MB)*E)
      COMMX = (XLR*EIX)/(1. + 2.*SHFZ)
C
      S1 = 0.414*E*BP(3, MB)/XLR
      S2 = COMMX*BP(5, MB)*(1. + 0.5*SHFZ)
      S3 = COMMX*BP(6, MB)*(1. + 0.5*SHFZ)
      S4 = COMMX*BP(7, MB)*(1. - SHFZ)
C
C     FORCE-FRAME DISPLACEMENT MATRIX
      DO 10 I = 1, 6
      SA(1, I) = S1*T(1, I)
      SA(2, I) = S2*T(2, I) + S4*T(3, I)
   10 SA(3, I) = S4*T(2, I) + S3*T(3, I)
C
      IF (IO .EQ. 2) RETURN
C
C     STIFFNESS MATRIX ASA
      DO 30 I = 1, 6
      DO 30 J = 1, 6
      ASA(I, J) = 0.
      DO 20 K = 1, 3
   20 ASA(I, J) = ASA(I, J) + T(K, I)*SA(K, J)
   30 ASA(J, I) = ASA(I, J)
C
C     FIXFD END FORCES
      DO 50 L = 1, NLD
      TX(3) = P(2, L)
      TX(1) = SNA*P(1, L)
      TX(2) = - CSA*P(1, L)
      TX(6) = P(4, L)
      TX(4) = SNA*P(3, L)
      TX(5) = - CSA*P(3, L)
      DO 40 I = 1, 6
   40 P(I, L) = TX(I)
   50 CONTINUE
      RETURN
      END
C     ****************************************************************
C     *     THIS SUBROUTINE FORMS BRACING ELEMENT STIFFNESS MATRIX   *
C     ****************************************************************
      SUBROUTINE DIAG(IO, L, VANG, DL, SNA, CSA, NTRU, XL1, TP)
      DIMENSION TP(2, NTRU)
C
      COMMON /STIF/ASA(12, 12), SA(8, 12), T(8, 12)
C
C     DIAGONAL STIFFNESS AND FORCE MATRICES
C
C     TRANSFORMATION MATRIX
      CALL TRANS(4, VANG, 0., DL, SNA, CSA, XL1, 0., 0., 0.)
C
C     MEMBER STIFFNESS
      COMM = TP(1, L)*TP(2, L)/DL
C
C     FORCE-FRAME DISPLACEMENT MATRIX
      DO 10 I = 1, 8
   10 SA(1, I) = COMM*T(1, I)
C
      IF (IO .EQ. 2) RETURN
C
C     STIFFNESS MATRIX
      DO 20 I = 1, 8
      DO 20 J = 1, 8
      ASA(I, J) = T(1, I)*SA(1, J)
   20 ASA(J, I) = ASA(I, J)
      RETURN
      END
C     ****************************************************************
C     * THIS SUBROUTINE ALLOCATES STORAGE REQUIRED FOR THE BUILDING  *
C     *                       STIFFNESS MATRIX                       *
C     ****************************************************************
      SUBROUTINE STIFF(IO, S, R, NN)
      DIMENSION S(1)
      COMMON /JUNK/NLD, N, ND, MM, MN, LM(12), P(12, 3), TT(8)
      COMMON /STIF/ES(12, 12), SA(8, 12), T(8, 12)
      COMMON /FTAPES/NTS, NTE, NTSL, NBKS
      NBKS = 3
      NTS = 11
      NTE = 12
      NTSL = 13
      NLD = 3
      MN = 2*MM
      MN6 = MN + 6
      LS = (MM*(MM + 1))/2
      LC = MM*MM
      NE = NN + NLD - MN
      LE = MM*NE
      NNM = NN - MN
```

```
      LSL = (NNM*(NNM + 1))/2 + NNM*NLD
      NNL = NN + NLD
      N1 = 1
      N2 = N1 + MAXO(LS, LE)
      N3 = N2 + LC
      N4 = N1 + LSL
      N3 = MAXO(N3, N4)
      N4 = N3 + MAXO(LS, LE)
      N5 = N4 + MN6*6
      N6 = N5 + MN*3
      N7 = N6 + NNL
      N8 = N7 + MM
      IF (IO .GT. 0) GO TO 10
      IO = N8 - N1
      RETURN
   10 GO TO (20, 40, 60, 70, 80), IO
C
C     IO=1 ZERO ARRAYS
C
   20 NL = N8 - 1
      DO 30 I = N1, NL
   30 S(I) = 0.0
      REWIND NTSL
      REWIND NTS
      REWIND NTE
      WRITE (NTS)  (S(I), I = 1, LS)
      WRITE (NTE)  (S(I), I = 1, LE)
      WRITE (NTSL) (S(I), I = 1, LSL)
      RETURN
C
C     IO=2 SHIFT STORAGE
C
   40 REWIND NTS
      READ (NTS)  (S(I), I = 1, LS)
      NL = N8 - N1
      DO 50 I = N2, NL
   50 S(I) = 0.0
      RETURN
C
C     IO=3 REDUCE EQN,S WRITE BS EQNS ON TAPE 3 (NBKS)
C
   60 NNN = MN + 3*N + 3
      MN6 = MN + 6
      CALL SHRINK(S(N1), LS, S(N6), MM, NNL, S(N7), S(N2), S(N3), S(N[illegible],
     &S(N1), S(N1), NE, LSL, S(N4), S(N5), MN6, MN)
      RETURN
C
C     IO=4  WRITE LAT STIF ON TAPE 2
C
   70 NFR = NN - MN - 3
      NT2 = 2
      NF = N1
      NL = (NFR*(NFR + 1))/2
      M = LSL - NNM
      L = M - NNM
      K = L - NNM
      WRITE (NT2) N, NFR, (S(I), I = NF, NL), (S(K + I), I = 1, NFR), (S
     &(L + I), I = 1, NFR), (S(M + I), I = 1, NFR)
      RETURN
C
C     IO=5  ADD ELEMENT STIFF TO FRAME STIFFNESS
C
   80 MN6 = MN + 6
      CALL ADDSM(S(N1), S(N2), S(N3), S(N4), S(N5), LS, MM, MN6, MN)
      RETURN
      END
C     ******************************************************************
C     *    THIS SUBROUTINE ASSEMBLES THE BUILDING STIFFNESS MATRIX      *
C     ******************************************************************
      SUBROUTINE ADDSM(S, C, SB, EF, PF, LS, MM, MN6, MN)
      DIMENSION S(LS), C(MM, MM), SB(LS), EF(MN6, 6), PF(MN, 3)
      COMMON /STIF/ES(12, 12)
      COMMON /JUNK/NLD, N, ND, OM, XN, LM(12), P(12, 3), TT(8)
      DIMENSION KM(12)
C
      DO 30 I = 1, ND
      II = LM(I)
      DO 20 J = 1, 3
      II = II - MM
      IF (II) 10, 10, 20
   10 KM(I) = J
      GO TO 30
   20 KM(I) = 3
   30 CONTINUE
C
C     ADD ELEMENT STIFF TO FRAME STIFFNESS
C
      MJ = MN + 3*(N - 1)
      DO 150 I = 1, ND
      II = LM(I)
      IF (II) 150, 150, 40
   40 KI = KM(I)
      DO 120 J = 1, ND
      JJ = LM(J)
      IF (JJ .LT. II) GO TO 120
      KJ = KM(J)
      MI = 3*(N - 1)
      GO TO (50, 80, 110), KI
   50 GO TO (60, 70, 100), KJ
   60 L = (JJ*(JJ - 1))/2 + II
      S(L) = S(L) + ES(I, J)
      GO TO 120
   70 C(II, JJ - MM) = C(II, JJ - MM) + ES(I, J)
      GO TO 120
   80 GO TO (120, 90, 100), KJ
   90 JM = JJ - MM
      L = (JM*(JM - 1))/2 + II - MM
      SB(L) = SB(L) + ES(I, J)
      GO TO 120
  100 MI = 0
  110 EF(II - MI, JJ - MJ) = EF(II - MI, JJ - MJ) + ES(I, J)
  120 CONTINUE
      GO TO (130, 130, 150), KI
  130 DO 140 L = 1, 3
  140 PF(II, L) = PF(II, L) + P(I, L)
  150 CONTINUE
C
      RETURN
      END
C     ******************************************************************
C     *   THIS SUBROUTINE CONDENSES THE BUILDING STIFFNESS MATRIX AND   *
C     *         GIVES THE BUILDING LATERAL STIFFNESS MATRIX             *
C     ******************************************************************
      SUBROUTINE SHRINK(S, LS, KH, MM, NNL, SD, C, SB, E, EB, SL, NE,
     &LSL, EF, PF, MN6, MN)
C
C      PERFORMS BLOCK BY BLOCK STATIC CONDENSATION ON SUBASSEMBLAGE
C     EQUATIONS
C
C     -----------
C     I         I      NSS=3*NUMBER OF STORIES
C     I S  C  E I      MM=3*NUMBER OF COLUMN LINES
```

```
C     I          I        E=(MM*NSS)
C     I   SB  EB I        S=(MM*(MM+1))/2
C     I          I        EB=(MM*NSS)
C     I      SL  I        C=MM*MM
C     I          I        SL=(NSS*(NSS+1))/2
C     ------------        SB=(MM*(MM+1))/2
C
C     MAXIMUM STORAGE GOVERNED BY THAT REQUIRED FOR ANY THREE BLOCKS
C
      DIMENSION S(LS), KH(NNL), C(MM, MM), SB(LS), SD(MM), E(MM, NE), EB
     &(MM, NE), SL(LSL), EF(MN6, 6), PF(MN, 3)
      COMMON /JUNK/NLD, N, ND, QM, XN, LM(12), P(12, 3), TT(8)
      COMMON /FTAPES/NTS, NTE, NTSL, NBKS
C
      NLD = 3
      MN = 2*MM
      MMP = MM + 1
      MNP = MN + 1
      NNN = MN + 3*N + 3
      NN = NNL - NLD
      NNM = NN - MN
C
C     DETERMINE PROFILE OF (S,C,SB)
C
      JB = 0
      DO 20 J = 1, MM
      KH(J) = 1
      DO 10 I = 1, J
      IF (S(JB + I)) 20, 10, 20
   10 KH(J) = I + 1
   20 JB = JB + J
      DO 40 J = 1, MM
      KH(J + MM) = 1
      DO 30 I = 1, MM
      IF (C(I, J)) 40, 30, 40
   30 KH(J + MM) = I + 1
   40 CONTINUE
C
C     REDUCTION OF S .
C
      SD(1) = S(1)
      JB = 1
      DO 110 J = 2, MM
      SD(J) = 0.0
      JM = J - 1
      IF (J .EQ. 2) GO TO 70
      IB = 1
      DO 60 I = 2, JM
      IM = I - 1
      SS = 0.0
      KF = MAXO(KH(I), KH(J))
      IF (KF .GT. IM) GO TO 60
C
C     REDUCE S BY S,S
C
      DO 50 K = KF, IM
   50 SS = SS + S(IB + K)*S(JB + K)
      S(JB + I) = S(JB + I) - SS
   60 IB = IB + I
C
C     RED COL BY SELF
C
   70 SS = 0.0
      KF = KH(J)
      IF (KF .GT. JM) GO TO 100
      KD = (KF*(KF + 1))/2
      DO 90 K = KF, JM
      IF (S(KD)) 80, 90, 80
   80 T = S(JB + K)/S(KD)
      SS = SS + T*S(JB + K)
      S(JB + K) = T
   90 KD = KD + K + 1
      S(JB + J) = S(JB + J) - SS
  100 SD(J) = S(JB + J)
  110 JB = JB + J
C
C     REDUCTION OF C,SB
C
      JB = 0
      DO 180 J = MMP, MN
      JJ = J - MM
      JM = J - 1
      IB = 1
      DO 150 I = 2, JM
      II = I - MM
      IM = I - 1
      KL = MINO(IM, MM)
      KF = MAXO(KH(I), KH(J))
      IF (KF .GT. KL) GO TO 150
      SS = 0.0
      IF (I .GT. MM) GO TO 130
C
C     RED C BY S,C
C
      DO 120 K = KF, KL
  120 SS = SS + S(IB + K)*C(K, JJ)
      C(I, JJ) = C(I, JJ) - SS
      GO TO 150
C
C     RED SB BY C,C
C
  130 DO 140 K = KF, KL
  140 SS = SS + C(K, II)*C(K, JJ)
      SB(JB + II) = SB(JB + II) - SS
  150 IB = IB + I
C
C     RED COL BY SELF
C
      SS = 0.0
      KF = KH(J)
      IF (KF .GT. MM) GO TO 180
      DO 170 K = KF, MM
      IF (SD(K)) 160, 170, 160
  160 T = C(K, JJ)/SD(K)
      SS = SS + T*C(K, JJ)
      C(K, JJ) = T
  170 CONTINUE
      SB(JB + JJ) = SB(JB + JJ) - SS
  180 JB = JB + JJ
      REWIND NTS
      WRITE (NTS) SB
      WRITE (NBKS) S, C
C
C     OVERWRITE SB WITH (E,EB)
      REWIND NTE
      READ (NTE) E
      L = 3*(N - 1)
      DO 190 I = 1, MM
      DO 190 J = 1, 6
  190 E(I, J + L) = E(I, J + L) + EF(I, J)
      DO 200 I = 1, MM
      DO 200 J = 1, 3
```

```
  200 E(I, J + NNM) = E(I, J + NNM) + PF(I, J)
C
C     DETERMINE PROFILE OF (E,EB)
C
      DO 220 J = MNP, NNL
      JJ = J - MN
      KH(J) = 1
      DO 210 I = 1, MM
      IF (E(I, JJ)) 220, 210, 220
  210 KH(J) = I + 1
  220 CONTINUE
C
C     REDUCE E BY S,E
C
      JB = 0
      DO 250 J = MNP, NNL
      JJ = J - MN
      IB = 1
      DO 240 I = 2, MM
      IM = I - 1
      KF = MAXO(KH(I), KH(J))
      SS = 0.0
      IF (KF .GT. IM) GO TO 240
      DO 230 K = KF, IM
  230 SS = SS + S(IB + K)*E(K, JJ)
      E(I, JJ) = E(I, JJ) - SS
  240 IB = IB + I
  250 CONTINUE
      DO 260 I = 1, MM
      DO 260 J = 1, NE
  260 EB(I, J) = 0.0
      L = 3*(N - 1)
      DO 270 I = 1, MM
      DO 270 J = 1, 6
  270 EB(I, J + L) = EB(I, J + L) + EF(I + MM, J)
      DO 280 I = 1, MM
      DO 280 J = 1, NLD
  280 EB(I, J + NNM) = EB(I, J + NNM) + PF(I + MM, J)
C
C     RED EB BY C,E
C
      DO 310 J = MNP, NNL
      JJ = J - MN
      DO 300 I = MMP, MN
      II = I - MM
      KF = MAXO(KH(I), KH(J))
      SS = 0.0
      IF (KF .GT. MM) GO TO 300
      DO 290 K = KF, MM
  290 SS = SS + C(K, II)*E(K, JJ)
  300 EB(II, JJ) = EB(II, JJ) - SS
  310 CONTINUE
      REWIND NTE
      REWIND NTSL
      WRITE (NTF) EB
C
C     OVERWRITE S BY SL
C
      READ (NTSL) SL
      L = 3*(N - 1)
      K = (L*(L + 1))/2
      DO 330 J = 1, 6
      DO 320 I = 1, J
  320 SL(K + L + I) = SL(K + L + I) + EF(I + MN, J)
  330 K = K + L + J
C
C     RED SL BY E,E
C
      JB = 0
      DO 400 J = MNP, NNL
      JJ = J - MN
      JM = J - 1
      JM = MINO(JM, NNN)
      IF (MNP .GT. JM) GO TO 360
      DO 350 I = MNP, JM
      KF = MAXO(KH(I), KH(J))
      II = I - MN
      SS = 0.0
      IF (KF .GT. MM) GO TO 350
      DO 340 K = KF, MM
  340 SS = SS + E(K, II)*E(K, JJ)
  350 SL(JB + II) = SL(JB + II) - SS
C
C     REDUCE COLS BY SELF
C
  360 SS = 0.0
      KF = KH(J)
      IF (KF .GT. MM) GO TO 390
      DO 380 K = KF, MM
      IF (SD(K)) 370, 380, 370
  370 T = E(K, JJ)/SD(K)
      SS = SS + T*E(K, JJ)
      E(K, JJ) = T
  380 CONTINUE
  390 JA = MINO(JJ, NNM)
      IF (J .GT. NNN) GO TO 400
      SL(JB + JA) = SL(JB + JA) - SS
  400 JB = JB + JA
      REWIND NTSL
      WRITE (NBKS) E, KH
      WRITE (NTSL) SL
C
      RETURN
C
      END
C     ****************************************************************
C     *   THIS SUBROUTINE TRANSFORMS THE BUILDING LATERAL STIFFNESS   *
C     *        MATRIX TO THE BUILDING REFERENCE POINTS                *
C     ****************************************************************
      SUBROUTINE LAT(A, SS, RR, S, R, D, MSS, NFS, MST)
      DIMENSION SS(MSS, MSS), RR(MSS, 5), A(MST, 14), S(NFS, NFS), R(NFS
     &, 3), D(MST, 2)
      INTEGER RD, WT
      COMMON /JRW/RD, WT
      COMMON /GFN/NST, NSF, NLD, NAT, NPO, NSD, NSS, RLAB(3), IS, I3
      COMMON /JUNK/AI(3, 3), AJ(3, 3), AR(3, 3), AS(3, 3)
C
C     COMPUTE STRUCTURE LATERAL STIFFNESS, SOLVE STATIC LOAD CASES
C
      REWIND 2
      DO 20 I = 1, NSS
      DO 10 J = 1, 3
   10 RR(I, J) = 0.
      DO 20 J = 1, NSS
   20 SS(I, J) = 0.
      DO 30 I = 1, NST
      II = I3*(I - 1)
      SS(II + 1, II + 1) = A(I, 7 + IS)
      IF (NSD .EQ. 0) SS(II + 2, II + 2) = A(I, 8)
      DO 30 J = 1, I3
      DO 30 L = 1, 2
      LL = 5 + 3*L + J + IS
```

```
   30 RR(II + J, L + 3) = A(I, LL)
C
      READ (2) NT, NFR, ((S(I, J), I = 1, J), J = 1, NFR), ((R(I, L), I
     & = 1, NFR), L = 1, 3)
      DO 40 I = 1, NFR
      DO 40 J = I, NFP
   40 S(J, I) = S(I, J)
      DO 50 I = 1, 3
      DO 50 J = 1, 3
   50 AI(I, J) = 0.
C
      AI(1, 1) = 1.
      AI(2, 2) = 1.
      AI(3, 3) = 1.
      DO 60 I = 1, 3
      DO 60 J = 1, 3
   60 AJ(J, I) = AI(I, J)
C
      DO 70 N = 1, NST
      XM = - A(N, 5)
      YM = - A(N, 6)
      D(N, 1) = - YM
   70 D(N, 2) = XM
C
      DO 130 N = 1, NST
      J2 = 3*(N - 1)
      AI(1, 3) = D(N, 1)
      AI(2, 3) = D(N, 2)
C
      DO 130 M = 1, NST
      I2 = 3*(M - 1)
      AJ(3, 1) = D(M, 1)
      AJ(3, 2) = D(M, 2)
C
      DO 80 II = 1, 3
      DO 80 J1 = 1, I3
      JJ = J1 + IS
      AR(II, JJ) = 0.
      DO 80 KK = 1, 3
   80 AR(II, JJ) = AR(II, JJ) + S(I2 + II, J2 + KK)*AI(KK, JJ)
C
      DO 110 I1 = 1, I3
      II = I1 + IS
C
      IF (N .NE. 1) GO TO 100
      DO 90 L = 1, 3
      DO 90 KK = 1, 3
   90 RR(I2 + I1, L) = RR(I2 + I1, L) + AJ(II, KK)*R(I2 + KK, L)
C
  100 DO 110 J1 = 1, I3
      JJ = J1 + IS
      AS(II, JJ) = 0.
      DO 110 KK = 1, 3
  110 AS(II, JJ) = AS(II, JJ) + AJ(II, KK)*AR(KK, JJ)
C
C
      DO 120 I = 1, I3
      II = I3*(M - 1) + I
      DO 120 J = 1, I3
      JJ = I3*(N - 1) + J
  120 SS(II, JJ) = SS(II, JJ) + AS(I + IS, J + IS)
  130 CONTINUE
C
      NT2 = 2
      REWIND NT2
      WRITE (NT2) SS, XM
      IF (NAT .EQ. 1) RETURN
      CALL SOLVE(SS, NSS, RR, 5)
C
C     PRINT STRUCTURE DISPLACEMENTS
C
      WRITE (WT,150)
      DO 140 I = 1, NST
      LN = NST + 1 - I
      WRITE (WT,170)
      DO 140 J = 1, I3
      II = I3*(I - 1) + J
  140 WRITE (WT,160) LN, RLAB(J + IS), (RR(II, L), L = 1, 5)
C
      WRITE (2) RR
      WRITE (2) D
C
      RETURN
  150 FORMAT ('1', 29X, '........ STRUCTURAL DISPLACEMENTS FOR FIVE BASI
     &C LOADING CASES ........'///35X, 'GRAVITY  LOADING  CASE', 15X, 'L
     &ATERAL LOADING CASE'/' LEVEL NO', 6X, 'DIRECTION', 8X, 'I', 12X, '
     &II', 11X, 'III', 13X, 'A', 13X, 'B')
  160 FORMAT (I8, 10X, A4, 2X, 5F14.6)
  170 FORMAT (1H )
      END
C     ******************************************************************
C     *     THIS SUBROUTINE SOLVES FOR FIVE STATIC LOAD CASES          *
C     ******************************************************************
      SUBROUTINE SOLVE(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
   10 M = M + 1
      MM = M + 1
      DO 20 L = 1, NL
   20 B(M, L) = B(M, L)/A(M, M)
      IF (M - N) 30, 90, 30
   30 DO 40 J = MM, N
   40 A(M, J) = A(M, J)/A(M, M)
C
C     SUBSTITUTION INTO REMAINING EQUATIONS
C
      DO 80 I = MM, N
      IF (A(I, M)) 50, 80, 50
   50 DO 60 J = I, N
      A(I, J) = A(I, J) - A(I, M)*A(M, J)
   60 A(J, I) = A(I, J)
      DO 70 L = 1, NL
   70 B(I, L) = B(I, L) - A(I, M)*B(M, L)
   80 CONTINUE
      GO TO 10
C
C     BACK SUBSTITUTION
C
   90 M = M - 1
      IF (M .EQ. 0) GO TO 110
      MM = M + 1
      DO 100 L = 1, NL
      DO 100 J = MM, N
  100 B(M, L) = B(M, L) - A(M, J)*B(J, L)
      GO TO 90
```

```
C
  110 RETURN
      END
C     ****************************************************************
C     *    THIS SUBROUTINE PERFORMS MODE SHAPES AND FREQUENCIES AND    *
C     *                      SPECTRUM ANALYSIS                         *
C     ****************************************************************
      SUBROUTINE EARTH(A, F, XM, W, IQ, S, PA, MSS, MST, NMD)
      DIMENSION F(MSS, MSS), S(MSS, MSS), XM(1), W(1), IQ(1), A(MST, 4),
     &PA(2, 1)
      INTEGER RD, WT
      INTEGER*2 INPT(80)
      COMMON /JRW/RD, WT
      COMMON /INPUT/INPT, FN(8)
      COMMON /GEN/NST, NSP, NLD, NAT, NFQ, NSD, NSS, RLAB(3), IS, I3
      COMMON /JUNK/SC(2)
      COMMON /DYN/NTIME, NPC, DT, DAMP
C
C     COMPUTE MODE SHAPES AND FREQUENCIES
C
      REWIND 2
      TPI = 8.0*ATAN(1.0)
      READ (2) S
      IF (NAT .EQ. 0) GO TO 180
      DO 10 I = 1, NST
      II = I3*(I - 1) + 1
      XM(II) = A(I, 3)
      XM(II + 1) = A(I, 3)
   10 XM(II + 2) = A(I, 4)
C
      DO 20 I = 1, NSS
      IF (XM(I) .GT. 0.) GO TO 20
      WRITE (WT,290)
      STOP
   20 XM(I) = 1.0/SQRT(XM(I))
      DO 30 I = 1, NSS
      DO 30 J = 1, NSS
   30 S(I, J) = S(I, J)*XM(I)*XM(J)
C
      CALL EIGEN(S, NSS, 0, F, NR, W, IQ)
C
      DO 40 I = 1, NSS
      W(I) = S(I, I)
      DO 40 J = 1, NSS
   40 F(I, J) = F(I, J)*XM(I)
      DO 60 I = 1, NFQ
      WM = W(I)
      DO 50 J = I, NSS
      IF (W(J) .GT. WM) GO TO 50
      WM = W(J)
      K = J
   50 CONTINUE
C
      W(K) = W(I)
      W(I) = TPI/SQRT(WM)
      DO 60 J = 1, NSS
      WM = F(J, K)
      F(J, K) = F(J, I)
   60 F(J, I) = WM
C     PRINT MODES,PERIODS
   70 WRITE (WT,210) (I, W(I), I = 1, NFQ)
      DO 80 I = 1, NFQ, 8
      IH = I + 7
      IF (IH .GT. NFQ) IH = NFQ
      WRITE (WT,230) (J, J = I, IH)
      DO 80 N = 1, NST
      LN = NST + 1 - N
      NN = I3*(N - 1)
      WRITE (WT,220)
      DO 80 J = 1, I3
   80 WRITE (WT,240) LN, RLAB(J + IS), (F(NN + J, K), K = I, IH)
      IF (NAT .EQ. 1) RETURN
C
C     DYNAMIC ANALYSIS
C
      IF (NAT .NE. 4) GO TO 90
C
C     GROUND MOTION CONTROL DATA
      READ (RD,190) INPT
      CALL CNVRT(1, 5)
      NPC = FN(1)
      NTIME = FN(2)
      SF = FN(3)
      FI = FN(4)
      DT = FN(5)
      WRITE (WT,250) NPC, NTIME, SF, FI, DT
      GO TO 110
C
   90 IF (NAT .NE. 3) GO TO 180
C
C     RESPONSE SPECTRUM DATA
      READ (RD,190) INPT
      CALL CNVRT(1, 4)
      NPC = FN(1)
      NMD = FN(2)
      SF = FN(3)
      FI = FN(4)
      DO 100 I = 1, NPC
      READ (RD,190) INPT
      CALL CNVRT(1, 2)
      PA(1, I) = FN(1)
      PA(2, I) = FN(2)
  100 CONTINUE
      WRITE (WT,200) NPC, NMD, SF, FI, (PA(1, I), PA(2, I), I = 1, NPC)
C
C     MODAL PARTICIPATION FACTORS
C
  110 FI = FI*ATAN(1.0)/45.
      SC(1) = SIN(FI)
      SC(2) = COS(FI)
      DO 130 I = 1, NFQ
      RLM = 0.
      DO 120 J = 1, NST
      JJ = I3*(J - 1) + 1
      FR = F(JJ, I)*SC(1 + IS)
      IF (NSD .EQ. 0) FR = FR + F(JJ + 1, I)*SC(2)
  120 RLM = RLM + FR*A(J, 3)
      WW = (TPI/W(I))**2
      IF (NAT .EQ. 3) YM = RLM*TABA(NPC, SF, W(I), PA)/WW
      IF (NAT .EQ. 4) YM = RLM*SF
      DO 130 J = 1, NSS
  130 F(J, I) = F(J, I)*YM
C
C     SAVE MODE SHAPES ON TAPE 1  NAT=4
C
      IF (NAT .NE. 4) GO TO 160
      REWIND 1
      WRITE (WT,270)
      DO 150 K = 1, NFQ
      READ (RD,190) INPT
      CALL CNVRT(1, 2)
      I = FN(1)
```

```
      DAMP = FN(2)
      IF (DAMP .LE. 1.0) GO TO 140
      DAMP = 1.0
      WRITE (WT,280) I
  140 WRITE (WT,260) I, DAMP
      WRITE (1) W(K), DAMP, (F(I, K), I = 1, NSS)
  150 CONTINUE
C
  160 DO 170 K = 1, NFQ
      I = NFQ + 1 - K
      DO 170 J = 1, NSS
  170 F(J, I + 5) = F(J, I)
C
  180 READ (2) ((F(I, J), I = 1, NSS), J = 1, 5)
C
      RETURN
C
  190 FORMAT (80A1)
  200 FORMAT (22H1ACCELERATION SPECTRUM, //25H NUMBER OF PERIOD CARDS =
     &I8/25H NUMBER OF LOWEST MODES =I8/25H ACCEL., UNITS/SEC/SEC  =
     &F10.3/25H ANGLE OF EQ INCIDENCE  =F10.3//26H  PERIOD      ACCELERA
     &TION//(F10.3, 5X, F10.3))
  210 FORMAT (22H1MODE NUMBER     PERIOD//(I7, 6X, F11.6))
  220 FORMAT (1H  )
  230 FORMAT (///12H MODE SHAPES//18H LEVEL    DIRECTION, 8I13)
  240 FORMAT (I4, 10X, A4, 2X, 8F13.6)
  250 FORMAT (23H1RESPONSE ANALYSIS DATA///30H NUMBER OF ACCELERATION CA
     &RDS  I3/30H NUMBER OF OUTPUT TIMES        I3/30H ACCELERATION SCAL
     &E FACTOR     F10.4/30H ANGLE OF EQ INCIDENCE          F10.4/30H TIM
     &E INCREMENT FOR OUTPUT     F10.4/)
  260 FORMAT (I4, F12.3)
  270 FORMAT (///16H MODE     DAMPING)
  280 FORMAT (30H DAMPING MUST BE LESS THAN 1.0/15H VALUE FOR MODEI3, 6H
     & RESET)
  290 FORMAT (///22H NEGATIVE OR ZERO MASS//21H EXECUTION TERMINATED)
      END
C     ******************************************************************
C     *           THIS SUBPROGRAM PERFORMS SPECTRUM INTERPOLATION      *
C     ******************************************************************
      FUNCTION TABA(NPC, SF, T, PA)
      DIMENSION PA(2, NPC)
C
C     SPECTRUM INTERPOLATION
C
      DO 10 I = 2, NPC
      T1 = PA(1, I - 1)
      T2 = PA(1, I)
      IF (T .LE. T2) GO TO 20
   10 CONTINUE
   20 R1 = (T2 - T)/(T2 - T1)
      R2 = (T - T1)/(T2 - T1)
      TABA = SF*(PA(2, I - 1)*R1 + PA(2, I)*R2)
      RETURN
      END
C     ******************************************************************
C     *        THIS SUBROUTINE FORMS EIGENVECTORS OF THE SYSTEM        *
C     ******************************************************************
      SUBROUTINE EIGEN(H, N, IEGEN, U, NR, X, IO)
C
      DIMENSION H(N, N), U(N, N), X(N), IO(N)
C
      IF (IEGEN) 50, 10, 50
   10 DO 40 I = 1, N
      DO 40 J = 1, N
      IF (I - J) 30, 20, 30
   20 U(I, J) = 1.0
      GO TO 40
   30 U(I, J) = 0.
   40 CONTINUE
C
   50 NR = 0
      IF (N - 1) 450, 450, 60
C
C     SCAN FOR LARGEST OFF DIAGONAL ELEMENT IN EACH ROW
C     X(I) CONTAINS LARGEST ELEMENT IN I TH ROW
C     IO(I) HOLDS SECOND SUBSCRIPT DEFINING POSITION OF ELEMENT
C
   60 NMI1 = N - 1
      DO 80 I = 1, NMI1
      X(I) = 0.
      IPL1 = I + 1
      DO 80 J = IPL1, N
      IF (X(I) - ABS(H(I, J))) 70, 70, 80
   70 X(I) = ABS(H(I, J))
      IO(I) = J
   80 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
   90 DO 120 I = 1, NMI1
      IF (I - 1) 110, 110, 100
  100 IF (XMAX - X(I)) 110, 120, 120
  110 XMAX = X(I)
      IPIV = I
      JPIV = IO(I)
  120 CONTINUE
C
C     IS MAX. X(I) EQUAL TO ZERO, IF LESS THAN HDTEST,REVISE HDTEST
      IF (XMAX) 450, 450, 130
  130 IF (HDTEST) 150, 150, 140
  140 IF (XMAX - HDTEST) 150, 150, 180
  150 HDIMIN = ABS(H(1, 1))
      DO 170 I = 2, N
      IF (HDIMIN - ABS(H(I, I))) 170, 170, 160
  160 HDIMIN = ABS(H(I, I))
  170 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) 180, 450, 450
  180 NR = NR + 1
C
C     COMPUTE TANGENT, SINE AND COSINE, H(I,I),H(J,J)
  190 TANG = SIGN(2., (H(IPIV, IPIV) - H(JPIV, JPIV)))*H(IPIV, JPIV)/(
     &ABS(H(IPIV, IPIV) - H(JPIV, JPIV)) + SQRT((H(IPIV, IPIV) - H(JPIV,
     &JPIV))**2 + 4.*H(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)))
      H(JPIV, JPIV) = COSINE**2*(H(JPIV, JPIV) - TANG*(2.*H(IPIV, JPIV)
     & - TANG*HII))
      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)) 200, 210, 210
  200 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
  210 CONTINUE
C
C     INSPECT THE IOS 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 290 I = 1, NMI1
      IF (I - IPIV) 230, 290, 220
  220 IF (I - JPIV) 230, 290, 230
  230 IF (IO(I) - IPIV) 240, 250, 240
  240 IF (IO(I) - JPIV) 290, 250, 290
  250 K = IO(I)
  260 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 280 J = IPL1, N
      IF (X(I) - ABS(H(I, J))) 270, 270, 280
  270 X(I) = ABS(H(I, J))
      IO(I) = J
  280 CONTINUE
      H(I, K) = HTEMP
  290 CONTINUE
C
      X(IPIV) = 0.
      X(JPIV) = 0.
C
C     CHANGE THE OTHER ELEMENTS OF H
C
      DO 420 I = 1, N
C
      IF (I - IPIV) 300, 420, 340
  300 HTEMP = H(I, IPIV)
      H(I, IPIV) = COSINE*HTEMP + SINE*H(I, JPIV)
      IF (X(I) - ABS(H(I, IPIV))) 310, 320, 320
  310 X(I) = ABS(H(I, IPIV))
      IO(I) = IPIV
  320 H(I, JPIV) = - SINE*HTEMP + COSINE*H(I, JPIV)
      IF (X(I) - ABS(H(I, JPIV))) 330, 420, 420
  330 X(I) = ABS(H(I, JPIV))
      IO(I) = JPIV
      GO TO 420
C
  340 IF (I - JPIV) 350, 420, 380
  350 HTEMP = H(IPIV, I)
      H(IPIV, I) = COSINE*HTEMP + SINE*H(I, JPIV)
      IF (X(IPIV) - ABS(H(IPIV, I))) 360, 370, 370
  360 X(IPIV) = ABS(H(IPIV, I))
      IO(IPIV) = I
  370 H(I, JPIV) = - SINE*HTEMP + COSINE*H(I, JPIV)
      IF (X(I) - ABS(H(I, JPIV))) 330, 420, 420
C
  380 HTEMP = H(IPIV, I)
      H(IPIV, I) = COSINE*HTEMP + SINE*H(JPIV, I)
      IF (X(IPIV) - ABS(H(IPIV, I))) 390, 400, 400
  390 X(IPIV) = ABS(H(IPIV, I))
      IO(IPIV) = I
  400 H(JPIV, I) = - SINE*HTEMP + COSINE*H(JPIV, I)
      IF (X(JPIV) - ABS(H(JPIV, I))) 410, 420, 420
  410 X(JPIV) = ABS(H(JPIV, I))
      IO(JPIV) = I
  420 CONTINUE
C
C     TEST FOR COMPUTATION OF EIGENVECTORS
C
      IF (IEGEN) 90, 430, 90
  430 DO 440 I = 1, N
      HTEMP = U(I, IPIV)
      U(I, IPIV) = COSINE*HTEMP + SINE*U(I, JPIV)
  440 U(I, JPIV) = - SINE*HTEMP + COSINE*U(I, JPIV)
      GO TO 90
  450 RETURN
      END
C     *****************************************************************
C     *       THIS SUBROUTINE PERFORMS DYNAMIC RESPONSE ANALYSIS      *
C     *****************************************************************
      SUBROUTINE DYNA(F, PA, X, T, MSS)
C     EVALUATION OF 3D TIME-DEPENDENT LATERAL DISPLACEMENTS
      DIMENSION F(MSS, 1), PA(2, 1), X(1), T(1)
      INTEGER RD, WT
      INTEGER*2 INPT(80)
      COMMON /JRW/RD, WT
      COMMON /INPUT/INPT, FN(8)
      COMMON /GEN/NST, NSF, NLD, NAT, NFO, NSD, NSS, RLAB(3), IS, T3
      COMMON /DYN/NTIME, NPC, DT, DAMP
C     ZERO DISPLACEMENTS AND READ GROUND ACCELERATIONS
      DO 10 I = 1, MSS
      DO 10 K = 1, NTIME
   10 F(I, K + 5) = 0.0
      DO 20 I = 1, NPC
      READ (RD,70) INPT
      CALL CNVRT(1, 2)
      PA(1, I) = FN(1)
      PA(2, I) = FN(2)
   20 CONTINUE
      WRITE (WT,80) (PA(1, I), PA(2, I), I = 1, NPC)
C
C     CHECK GROUND ACCELERATION DATA
C
      DO 30 I = 2, NPC
      IF (PA(1, I) .GT. PA(1, I - 1)) GO TO 30
      WRITE (WT,100) PA(1, I - 1), PA(1, I)
      STOP
   30 CONTINUE
      TIME = NTIME
      TIME = TIME*DT
      ATIME = PA(1, NPC) - PA(1, 1)
      IF (ATIME .GE. TIME) GO TO 40
      WRITE (WT,90)
      STOP
C
C     FOR EACH MODE CALCULATE RESPONSE AND TOTAL DISPLACEMENTS
C
   40 REWIND 1
      DO 60 L = 1, NFO
      READ (1) W, DAMP, (T(K), K = 1, MSS)
      CALL RESP(PA, X, W, NTIME, NPC, DT, DAMP)
      DO 50 I = 1, MSS
      DO 50 J = 1, NTIME
   50 F(I, J + 5) = F(I, J + 5) + T(I)*X(J)
```

```
   60 CONTINUE
C
      RETURN
   70 FORMAT (80A1)
   80 FORMAT (///26H GROUND ACCELERATION CARDS//11X, 4HTIME, 3X, 12HACCE
     &LERATION/(2F15.4))
   90 FORMAT (44H1GROUND ACCELERATION TIME SPAN IS LESS THAN  32HSPECIFI
     &ED BUILDING RESPONSE TIME)
  100 FORMAT (32H0 INCONSISTENT ACCELERATION DATA//34H  TIMES MUST INCRE
     &ASE SEQUENTIALLY//16H  ERROR AT TIMESF12.5/16X, F12.6)
      END
C     ******************************************************************
C     *     THIS SUBROUTINE CALCULATES RESPONSES FOR EACH TIME STEP    *
C     ******************************************************************
      SUBROUTINE RESP(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
C
      L = 0
      II = 1
      VO = 0.
      VDO = 0.
      TO = PA(1, 1)
   10 DT = DDT
   20 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 30
      DELT = PA(1, II + 1) - TO
      GO TO 40
   30 DELT = DT
C
   40 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 50
      DT = DT - DELT
      II = II + 1
      TO = PA(1, II)
      IF (DT .EQ. 0.) GO TO 60
      GO TO 20
C
   50 TO = TO + DT
   60 L = L + 1
      X(L) = VT
      IF (L .LT. NTIME) GO TO 10
C
      RETURN
      END
C     ******************************************************************
C     *     THIS SUBROUTINE READS LOADING COMBINATION CARDS            *
C     ******************************************************************
      SUBROUTINE LOAD(XM, NLD, NAT)
      DIMENSION XM(8, NLD)
      INTEGER RD, WT
      INTEGER*2 INPT(80)
      COMMON /JFW/RD, WT
      COMMON /INPUT/INPT, FN(8)
C
      WRITE (WT,100)
      IF (NAT .NE. 0) GO TO 10
      WRITE (WT,90)
      K = 5
      GO TO 20
   10 WRITE (WT,60)
      K = 8
   20 DO 40 L = 1, NLD
      READ (RD,50) INPT
      CALL CNVRT(1, K)
      DO 30 I = 1, K
   30 XM(I, L) = FN(I)
   40 WRITE (WT,70) L, (XM(I, L), I = 1, K)
      IF (NAT .EQ. 0) RETURN
      WRITE (WT,80)
      RETURN
C
   50 FORMAT (80A1)
   60 FORMAT (///25X, 'GRAVITY LOADING TYPE', 10X, 'LATERAL LOADING TYPE
     &'/' COMBINATION', 11X, 'I', 10X, 'II', 9X, 'III', 11X, 'A', 11X, '
     &B', 10X, 'SPECTRUM-1', 2X, 'SPECARUM-2', 4X, 'RESPONSE')
   70 FORMAT (I10, 2X, 5F12.2, 8X, 3F12.2)
   80 FORMAT (////43H SPECTRUM-1... ROOT MEAN SQUARE COMBINATION//37H SP
     &ECTRUM-2... SUM OF ABSOLUTE VALUES)
   90 FORMAT (///25X, 'GRAVITY LOADING TYPE', 10X, 'LATERAL LOADING TYPE
     &'/' COMBINATION', 11X, 'I', 10X, 'II', 9X, 'III', 11X, 'A', 11X, '
     &B')
  100 FORMAT ('1', 43X, '........ LOADING COMBINATIONS ........')
C
      END
C     ******************************************************************
C     *  THIS SUBROUTINE COMPUTES STRUCTURE LATERAL DISPLACEMENTS AT   *
C     *              THE REFERENCE POINTS OF THE BUILDING              *
C     ******************************************************************
      SUBROUTINE DISP(SD, R, D, RF, MSS, MLD, MST, NFS)
      DIMENSION SD(MST, 8), R(MSS, MLD), D(MST, 2), RF(NFS, MLD)
      COMMON /GEN/SSPACE(7), RLAB(3), IS, I3
      COMMON /JUNK/AI(3, 3), AJ(3, 5)
C
      READ (2) D
      REWIND 2
C
      DO 10 L = 1, 3
      DO 10 J = 1, 3
   10 AI(L, J) = 0.
      AI(1, 1) = 1.
      AI(2, 2) = 1.
      NT = 1
```

```
      AI(3, 3) = 1.
C
      NS = MST + 1 - NT
      NFR = NS*3
      DO 30 N = NT, MST
      NN = (N - NT)*I3
      N1 = (N - NT)*3
      AI(1, 3) = D(N, 1)
      AI(2, 3) = D(N, 2)
      DO 20 II = 1, 3
      DO 20 L = 1, MLD
      RF(N1 + II, L) = 0.
      DO 20 KK = 1, I3
      K1 = KK + IS
   20 RF(N1 + II, L) = RF(N1 + II, L) + AI(II, K1)*R(NN + KK, L)
   30 CONTINUE
C
      NTS = (NT - 1)*3 + 1
      WRITE (2) NFR, ((RF(N, L), N = NTS, NFS), L = 1, MLD)
      RETURN
      END
C     ******************************************************************
C     * THIS SUBROUTINE ALLOCATES STORAGE REQUIRED FOR THE CALCULATION *
C     *                    OF THE MEMBER FORCES                        *
C     ******************************************************************
      SUBROUTINE FRAME(MTOT, NMD, NFS)
      COMMON A(1)
      INTEGER RD, WT
      COMMON /JRW/RD, WT
      COMMON /GEN/NST, NSF, NLD, NAT, NFO, SSPACE(2), RLAB(3), IS, I3
      COMMON /DYN/NTIME, NPC, DT, DAMP
      COMMON /STIF/ASA(12, 12), SA(8, 12), T(8, 12)
      INTEGER MTOT1/'MTOT'/
C
C     OUTPUT DISPLACEMENTS AND FORCES
C
      N0 = 1 + 2*NST
      CALL LOAD(A(N0), NLD, NAT)
C
      REWIND 2
      MLD = 5 + NFO
      IF (NAT .EQ. 4) MLD = 5 + NTIME
      NN1 = N0 + 8*NLD
      NN2 = NN1 + NSF*3
      N1 = NN2 + 32*NSF
C
      CALL SORT(A(N1), A(N0), NFS, NLD, MLD, NAT, NMD, NFR)
      BACKSPACE 3
      READ (3) NC, NCP, NBP, NFEF, NPP, NTRU, NB, NPB
      BACKSPACE 3
C
      MM = 3*NC
      NN = 6*NC + 3*NST + 3
      N2 = N1 + NFS*MLD
      N3 = N2 + NC*2
      N4 = N3 + 9*NCP
      N5 = N4 + 9*NBP
      N6 = N5 + 7*NFEF
      N7 = N6 + NB*NST*3
      N8 = N7 + NST*NB*3
      N9 = N8 + NST*NC*1
      N10 = N9 + 3*NPB*NST
      LS = (MM*(MM + 1))/2
      LC = MM*MM
      LE = MM*(3*NST + 3)
      LKH = NN
      N11 = N10 + LS + LE + LC + LKH
      N12 = N11 + NN*MLD
      N13 = N12 + NFEF
      N14 = N13 + 5*NPP
      N15 = N14 + 8*NFO
      N16 = N15 + 3*NTRU
      N17 = N16 + 2*NTRU
      IF (N17 .LE. MTOT) GO TO 10
      MTOT = N17
      CALL ERROR(MTOT1, MTOT, WT)
   10 CALL OUTPUT(A(1), A(N0), A(N1), A(N2), A(N3), A(N4), A(N5), A(N6),
     &A(N7), A(N8), A(N9), A(N10), A(N11), NST, NB, NC, NCP, NBP, NFEF,
     &NPB, A(N12), A(N13), NN, MM, NLD, MLD, A(N14), NMD, A(N15), A(N16)
     &, NTRU, NFR, NFS, NFO, NSF, A(NN1), A(NN2), NPP)
      RETURN
C
      END
C     ******************************************************************
C     * THIS SUBROUTINE SORTS THE STRUCTURE LATERAL DISPLACEMENTS FROM *
C     *  INDIVIDUAL LOAD CASES ACCORDING TO LOADING COMBINATION CARDS  *
C     ******************************************************************
      SUBROUTINE SORT(U, XM, NFS, NLD, MLD, NAT, NMD, NFR)
      DIMENSION U(NFS, MLD), XM(8, NLD)
      INTEGER RD, WT
      COMMON /JRW/RD, WT
      COMMON /JUNK/UU(8, 3), UMIN(8, 3)
      COMMON /GEN/SSPACE(7), RLAB(3), IS, I3
      COMMON /DYN/NTIME, NPC, DT, DAMP
      DIMENSION HDM(8), LNUM(8)
      DATA HD1, HD2, HD3, HD4/3HMAX, 3HMIN, 1H , 1HM/
C
C     LOAD CASE COMBINATION AND OUTPUT OF DISPLACEMENTS
C
      READ (2) NFR, ((U(N, L), N = 1, NFR), L = 1, MLD)
C
      LLD = NLD + NMD
      DO 100 LL = 1, LLD, 8
      LH = LL + 7
      IF (LH .GT. LLD) LH = LLD
      IH = LH + 1 - LL
      WRITE (WT,120)
      IF (NAT .GE. 3) WRITE (WT,190)
      IF (NMD .GT. 0) WRITE (WT,170)
      DO 10 L = LL, LH
      LNUM(L + 1 - LL) = L
      HDM(L + 1 - LL) = HD3
      IF (L .LE. NLD) GO TO 10
      LNUM(L + 1 - LL) = L - NLD
      HDM(L + 1 - LL) = HD4
   10 CONTINUE
      WRITE (WT,130) (HDM(I), LNUM(I), I = 1, IH)
C
C     STATIC DISPLACEMENT COMPINENTS
      NS = NFR/3
      DO 90 N = 1, NS
      NL = NS + 1 - N
      DO 90 II = 1, 3
      NV = (N - 1)*3 + II
      DO 20 L = LL, LH
      I = L - LL + 1
      UU(I, II) = 0.
      DO 20 J = 1, 5
   20 UU(I, II) = UU(I, II) + U(NV, J)*XM(J, L)
      IF (NAT .GE. 3) GO TO 30
      WRITE (WT,140) NL, RLAB(II), (UU(I, II), I = 1, IH)
      GO TO 90
```

```
C     DYNAMIC DISP.ACEMENT COMPONENTS
   30 NATT = NAT - 3
      S1 = 0.
      S2 = 0.
      TH = 0.
      DO 60 I = 6, MLD
      UA = ABS(U(NV, I))
      IF (NATT) 40, 40, 50
   40 S1 = S1 + UA*UA
      S2 = S2 + UA
      GO TO 60
   50 IF (UA .GT. TH) TH = UA
   60 CONTINUE
      S1 = SQRT(S1)
      DO 80 L = LL, LH
      I = L - LL + 1
      IF (L .GT. NLD) GO TO 70
      UD = S1*XM(6, L) + S2*XM(7, L) + TH*XM(8, L)
      UMIN(I, II) = UU(I, II) - UD
      UU(I, II) = UU(I, II) + UD
      GO TO 80
   70 UMIN(I, II) = U(NV, L + 5 - NLD)
      UU(I, II) = UMIN(I, II)
   80 CONTINUE
      WRITE (WT,150) NL, HD1, RLAB(II), (UU(I, II), I = 1, IH)
      WRITE (WT,160) HD2, RLAB(II), (UMIN(I, II), I = 1, IH)
   90 IF (II .EQ. 3) WRITE (WT,180)
  100 CONTINUE
C
  110 RETURN
C
  120 FORMAT ('1', 24X, '........ LATERAL FRAME DISPLACEMENTS FOR LOADIN
     &G COMBINATION 1  2  3  ETC ........'////)
  130 FORMAT (19H LEVEL    DIRECTION, 6X, A1, I2, 7(11X, A1, I2))
  140 FORMAT (I5, 9X, A4, 1X, 8F14.7)
  150 FORMAT (I5, 3X, A3, 3X, A4, 1X, 8F14.7)
  160 FORMAT (8X, A3, 3X, A4, 1X, 8F14.7)
  170 FORMAT (45HO CONTRIBUTION TO SPECTRAL RESPONSE FROM EACH/60H  INDI
     &VIDUAL MODE LISTED UNDER HEADINGS    M 1    M 2     ETC)
  180 FORMAT (1H  )
  190 FORMAT (23HOMAX...STATIC + DYNAMIC/23H MIN...STATIC - DYNAMIC)
      END
C     ******************************************************************
C     *  THIS SUBROUTINE COMPUTES JOINT DISPLACEMENTS FROM STRUCTURE    *
C     *                    LATERAL DISPLACEMENTS                        *
C     ******************************************************************
      SUBROUTINE BKSUB(S, C, E, LS, MM, NNM, R, NN, MLD, KH)
      DIMENSION S(LS), C(MM, MM), E(MM, NNM), R(NN, MLD), KH(NN)
C
C     COMPUTE FRAME DISPLACEMENTS AT ONE LEVEL
C
      NBKS = 3
      BACKSPACE NBKS
      BACKSPACE NBKS
      READ (NBKS) S, C
      READ (NBKS) E, ((R(I, J), I = 1, MM), J = 1, 3), KH
      BACKSPACE NBKS
      BACKSPACE NBKS
      MN = 2*MM
      MMP = MM + 1
      DO 60 L = 1, MLD
      DO 40 J = MMP, MN
      KP = KH(J)
      IF (KP .GT. MM) GO TO 40
      IF (J .GT. MN) GO TO 20
      DO 10 K = KP, MM
```

```
   10 R(K, L) = R(K, L) - C(K, J - MM)*R(J, L)
      GO TO 40
   20 DO 30 K = KP, MM
   30 R(K, L) = R(K, L) - E(K, J - MN)*R(J, L)
   40 CONTINUE
      IB = LS
      DO 60 II = 1, MM
      I = MM + 1 - II
      KK = I - 1
      KP = KH(I)
      IB = IB - I
      IF (KP .GT. KK) GO TO 60
      DO 50 K = KP, KK
   50 R(K, L) = R(K, L) - S(IB + K)*R(I, L)
   60 CONTINUE
      RETURN
      END
C     ******************************************************************
C     *  THIS SUBROUTINE CALCULATES MEMBER FORCES FOR ALL STRUCTURAL    *
C     *                        ELEMENTS                                  *
C     ******************************************************************
      SUBROUTINE OUTPUT(SD, XM, U, CLN, CP, BP, FEF, LB, LDB, LC, LP, S,
     &R, NST, NB, NC, NCP, NBP, NFEF, NPB, IFEF, PP, NN, MM, NLD, MLD,
     &FM, NMD, LT, TP, NTRU, NFR, NFS, NFO, NSF, IJ, CIJ, NPP)
      DIMENSION CLN(NC, 2), CP(9, NCP), BP(9, NBP), FEF(7, NFEF), SD(NST
     &, 2), LB(NST, NB, 3), LDB(NST, NB, 3), LC(NST, NC, 1), LP(NST, NPB
     &, 3), IFEF(NFEF), PP(5, NPP), S(1), R(NN, MLD), U(NFS, MLD), XM(8,
     &NLD), LT(3, NTRU), TP(2, NTRU), FM(8, NFO), IJ(NSF, 3), CIJ(NSF, 2
     &, 16)
C
      INTEGER RD, WT
      INTEGER*2 INPT(80)
      COMMON /JRW/RD, WT
      COMMON /INPUT/INPT, FN(8)
      COMMON /JUNK/ND, NF, LM(12), F(8, 8), Q(3, 3), P(8), PMIN(8)
      COMMON /STIF/ASA(12, 12), SA(8, 12), T(8, 12)
C
C     SET FOUNDATION DISPLACEMENTS AND LATERAL DISPLACEMENTS
C
      REWIND 11
      BACKSPACE 3
      IF (NTRU .EQ. 0) GO TO 10
      READ (3) LT, TP
      BACKSPACE 3
      BACKSPACE 3
   10 IF (NPB .EQ. 0) GO TO 20
      READ (3) LP
      BACKSPACE 3
      BACKSPACE 3
   20 IF (NFEF .EQ. 0) GO TO 30
      READ (3) FEF, IFEF
      BACKSPACE 3
      BACKSPACE 3
   30 IF (NB .EQ. 0) GO TO 40
      READ (3) LB, LDB
      BACKSPACE 3
      BACKSPACE 3
   40 READ (3) CLN, CP, BP, LC, PP
      BACKSPACE 3
   50 MN = 6*NC
      KL = MM + 1
      DO 80 L = 1, MLD
      DO 60 K = 1, MN
   60 R(K, L) = 0.0
      DO 70 N = 1, NFR
      KK = MN + N
```

```
      KM = MN + NFR
   70 R(KK, L) = U(N, L)
      DO 80 II = 1, 3
   80 R(KM + II, L) = 0.
C
C     READ STORY LEVELS AND SLAB LINES WHERE SLAB FORCES ARE REQUIRED
C
      IF (NSF .EQ. 0) GO TO 150
      DO 140 I = 1, NSF
      READ (RD,500) INPT
      CALL CNVRT(1, 3)
      IJ(I, 1) = FN(1)
      IJ(I, 2) = FN(2)
      IJ(I, 3) = FN(3)
      L1 = IJ(I, 2)
      IF (L1 .EQ. 0) GO TO 110
      READ (RD,500) INPT
      M1 = L1
      IF (L1 .GT. 8) M1 = 8
      CALL CNVRT(1, M1)
      DO 90 J = 1, M1
   90 CIJ(I, 1, J) = FN(J)
      IF (L1 .LE. 8) GO TO 110
      READ (RD,500) INPT
      CALL CNVRT(1, 8)
      DO 100 J = 9, L1
  100 CIJ(I, 1, J) = FN(J - 8)
  110 L1 = IJ(I, 3)
      IF (L1 .EQ. 0) GO TO 140
      READ (RD,500) INPT
      M1 = L1
      IF (L1 .GT. 8) M1 = 8
      CALL CNVRT(1, M1)
      DO 120 J = 1, M1
  120 CIJ(I, 2, J) = FN(J)
      IF (L1 .LE. 8) GO TO 140
      READ (RD,500) INPT
      CALL CNVRT(1, 8)
      DO 130 J = 9, L1
  130 CIJ(I, 2, J) = FN(J - 8)
  140 CONTINUE
C
C     STORY BY STORY CALCULATION OF MEMBER FORCES
C
  150 N = NST
C
C     CALCULATE ROTATIONS AND VERTICAL DISPLACEMENT AT LEVEL N
C
  160 CONTINUE
      NNM = NN - 2*MM
      KC = (MM*(MM + 1))/2
      KE = MM*MM + KC
      KKH = MM*NNM + KE
      CALL BKSUB(S(1), S(KC + 1), S(KE + 1), KC, MM, NNM, R, NN, MLD, S(
     &KKH + 1))
C
C     CALCULATE COLUMN FORCES
C
      LV = NST - N + 1
      IF (NSF .EQ. 0) GO TO 180
      DO 170 II = 1, NSF
      IF (LV .NE. IJ(II, 1)) GO TO 170
      WRITE (11) ((R(I, J), I = 1, MM), J = 1, MLD)
  170 CONTINUE
  180 WRITE (WT,390) LV
      WRITE (WT,460)
      WRITE (WT,400)
      NF = 8
      ND = 12
      DO 210 M = 1, NC
      MC = LC(N, M, 1)
      IF (MC .EQ. NCP) GO TO 190
      XL = SD(N, 2)
      CSA = 0.
      SNA = 1.
      XL1 = CLN(M, 1)
      YL1 = CLN(M, 2)
      CALL COLUMN(2, MC, XL, SNA, CSA, NCP, XL1, YL1, CP)
C
      LM(6) = 3*M
      LM(5) = LM(6) - 1
      LM(4) = LM(5) - 1
      LM(11) = LM(5) + MM
      LM(10) = LM(4) + MM
      LM(12) = LM(6) + MM
      LM(3) = MN + 3*N
      LM(2) = LM(3) - 1
      LM(1) = LM(2) - 1
      LM(9) = LM(3) + 3
      LM(8) = LM(9) - 1
      LM(7) = LM(8) - 1
      M1 = M
      M2 = 0
      CALL STRESS(R, XM, NN, NLD, MLD, M1, M2, FM, NMD, NFQ)
      GO TO 200
  190 WRITE (WT,480) M
  200 WRITE (WT,440)
  210 CONTINUE
C
C     CALCULATE BEAM FORCES
C
      IF (NB .EQ. 0) GO TO 270
      WRITE (WT,470)
      WRITE (WT,410)
      NF = 3
      ND = 6
      DO 260 M = 1, NB
      MB = LB(N, M, 3)
      IF (MB .EQ. NBP) GO TO 260
      KI = LB(N, M, 1)
      KJ = LB(N, M, 2)
      B1 = CLN(KJ, 1) - CLN(KI, 1)
      B2 = CLN(KJ, 2) - CLN(KI, 2)
      B3 = B1*B1 + B2*B2
      B3 = SQRT(B3)
      CSA = B1/B3
      SNA = B2/B3
      XL = B3
      CALL BEAM(2, MB, XL, SNA, CSA, NBP, BP)
C
      LM(3) = 3*KI
      LM(2) = LM(3) - 1
      LM(1) = LM(2) - 1
      LM(6) = 3*KJ
      LM(5) = LM(6) - 1
      LM(4) = LM(5) - 1
      M1 = KI
      M2 = KJ
      DO 240 L = 1, 3
      DO 220 I = 1, 2
  220 Q(I, L) = 0.
      J = LDB(N, M, L)
```

```
      IF (J) 240, 240, 230
  230 V = FEF(5, J)*XL/2.
      YM = V*XL/6.
      O(1, L) = FEF(1, J) + YM
      O(2, L) = - FEF(3, J) - YM
  240 O(3, L) = 0.0
      CALL STRESS(R, XM, NN, NLD, MLD, M1, M2, FM, NMD, NFO)
  250 WRITE (WT,440)
  260 CONTINUE
C
C     CALCULATE PANEL FORCES
C
  270 IF (NPB .EQ. 0) GO TO 310
      WRITE (WT,430)
      LNF = 0
      ND = 10
      DO 300 M = 1, NPB
      MB = LP(N, M, 3)
      IF (MB .EQ. 0) GO TO 300
      NF = 4
      IF (PP(3, MB) .EQ. 0.) NF = 2
      IF (LNF .EQ. NF) GO TO 280
      IF (NF .EQ. 2) WRITE (WT,490)
      IF (NF .EQ. 4) WRITE (WT,420)
      LNF = NF
  280 XL = SD(N, 2)
      KI = LP(N, M, 1)
      KJ = LP(N, M, 2)
      B1 = CLN(KJ, 1) - CLN(KI, 1)
      B2 = CLN(KJ, 2) - CLN(KI, 2)
      B3 = B1*B1 + B2*B2
      B3 = SQRT(B3)
      CSA = B1/B3
      SNA = B2/B3
      AF = CLN(KI, 1)*CLN(KJ, 2) - CLN(KJ, 1)*CLN(KI, 2)
      DF = AF/B3
      D = B3
      CALL PANEL(2, MB, XL, D, DF, SNA, CSA, NPP, PP)
C
      LM(3) = MN + 3*N
      LM(2) = LM(3) - 1
      LM(1) = LM(2) - 1
      LM(4) = 3*KI
      LM(5) = 3*KJ
      LM(8) = MN + 3*N + 3
      LM(7) = LM(8) - 1
      LM(6) = LM(7) - 1
      LM(9) = MM + 3*KI
      LM(10) = MM + 3*KJ
      M1 = KI
      M2 = KJ
      CALL STRESS(R, XM, NN, NLD, MLD, M1, M2, FM, NMD, NFO)
  290 WRITE (WT,440)
  300 CONTINUE
C
  310 IF (NTRU .EQ. 0) GO TO 350
      DO 320 L = 1, NTRU
      NP = NST - LT(1, L) + 1
      IF (NP .NE. N) GO TO 320
      WRITE (WT,450)
      GO TO 330
  320 CONTINUE
      GO TO 350
  330 NF = 1
      ND = 8
      DO 340 L = 1, NTRU
      NP = NST - LT(1, L) + 1
      IF (NP .NE. N) GO TO 340
      KJ = LT(2, L)
      KI = LT(3, L)
      B1 = CLN(KJ, 1) - CLN(KI, 1)
      B2 = CLN(KJ, 2) - CLN(KI, 2)
      B3 = B1*B1 + B2*B2
      B3 = SQRT(B3)
      CSA = B1/B3
      SNA = B2/B3
      YL = SD(N, 2)
      DL = SQRT(B3*B3 + YL*YL)
      VANG = ATAN(YL/B3)
      AR = CLN(KI, 1)*CLN(KJ, 2) - CLN(KJ, 1)*CLN(KI, 2)
      D1 = AR/B3
      CALL DIAG(2, L, VANG, DL, SNA, CSA, NTRU, D1, TP)
C
      LM(3) = MN + 3*N
      LM(2) = LM(3) - 1
      LM(1) = LM(2) - 1
      LM(4) = 3*KJ
      LM(7) = MN + 3*N + 3
      LM(6) = LM(7) - 1
      LM(5) = LM(6) - 1
      LM(8) = 3*KI + MM
      M1 = KJ
      M2 = KI
      CALL STRESS(R, XM, NN, NLD, MLD, M1, M2, FM, NMD, NFO)
      WRITE (WT,440)
  340 CONTINUE
C
C     SHIFT DISPLACEMENTS
C
  350 DO 360 K = 1, MM
      KK = MM + K
      DO 360 L = 1, MLD
      R(KK, L) = R(K, L)
  360 R(K, L) = 0.0
C
  370 N = N - 1
      IF (N .NE. 0) GO TO 160
C
  380 RETURN
C
  390 FORMAT ('1', 42X, '........ MEMBER FORCES   LEVEL NO.', I4, ' ....
     &....'//)
  400 FORMAT (131H0COLUMN   LOADING        TORSIONAL             X-X AXIS
     &                    AXIAL            Y-Y AXIS              Y-Y AXIS
     &    X-X AXIS/131H         COMBINATION          MOMENT    TOP MOMENT
     & BTM MOMENT           FORCE    TOP MOMENT    BTM MOMENT         SHEA
     &R             SHEAR)
  410 FORMAT (74H    COL I    COL J    LOADING          TORS MOMENT      I  M
     &OMENT     J  MOMENT/18X, 'COMBINATION')
  420 FORMAT (16H FLEXURAL PANELS/'   COL I', 3X, 'COL J', 4X, 'LOADING'
     &, 9X, 'TOP-MOMENT', 4X, 'BTM-MOMENT', 3X, 'AXIAL-FORCE', 3X, 'SHEA
     &R-FORCE'/18X, 'COMBINATION')
  430 FORMAT (18H WALL PANEL FORCES//)
  440 FORMAT (1H )
  450 FORMAT (' BRACING ELEMENT FORCES'//' TOP COL', 2X, 'BTM COL', 3X,
     &'LOADING', 8X, 'AXIAL FORCE'/18X, 'COMBINATION')
  460 FORMAT (14H COLUMN FORCES/)
  470 FORMAT (12H BEAM FORCES/)
  480 FORMAT (' COLUMN', I5, 1X, 'IS A DUMMY COLUMN')
  490 FORMAT (13H SHEAR PANELS/'   COL I', 3X, 'COL J', 4X, 'LOADING',
     &8X, 'SHEAR FORCE', 2X, 'SHEAR STRESS'/18X, 'COMBINATION')
  500 FORMAT (80A1)
```

```
      END
C     ******************************************************************
C     *    THIS SUBROUTINE PRINTS MEMBER FORCES FOR EACH LOADING       *
C     *                      COMBINATION                               *
C     ******************************************************************
      SUBROUTINE STRESS(R, XM, NN, NLD, MLD, M, M2, FM, NMD, NFO)
      DIMENSION R(NN, MLD), XM(8, NLD), FM(8, NFO)
      INTEGER RD, WT
      COMMON /JRW/RD, WT
      COMMON /JUNK/ND, NF, LM(12), F(8, 8), O(3, 3), P(8), PMIN(8)
      COMMON /GEN/FIL(3), NAT, FIL2(8)
      COMMON /STIF/ASA(12, 12), SA(8, 12), T(8, 12)
      DATA HD1, HD2/3HMAX, 3HMIN/
C
C     LOAD CASE COMBINATION AND OUTPUT OF MEMBER FORCES
C
      LLD = NLD + NMD
      DO 10 I = 1, NF
      DO 10 J = 1, 8
   10 F(I, J) = 0.0
      IF (NF .NE. 3) GO TO 30
      DO 20 L = 1, 3
      F(2, L) = Q(1, L)
   20 F(3, L) = Q(2, L)
C
C     CALCULATE STATIC FORCES
C
   30 DO 40 K = 1, ND
      KK = LM(K)
      DO 40 I = 1, NF
      DO 40 J = 1, 5
   40 F(I, J) = F(I, J) + SA(I, K)*R(KK, J)
C
C     CALCULATE DYNAMIC FORCES
C
      NATT = NAT - 3
      IF (NATT) 110, 50, 50
   50 DO 100 I = 1, NF
      DO 90 J = 6, MLD
      X = 0.0
      DO 60 K = 1, ND
      KK = LM(K)
   60 X = X + SA(I, K)*R(KK, J)
      IF (NMD .GT. 0) FM(I, J - 5) = X
      XA = ABS(X)
      IF (NATT) 70, 70, 80
   70 F(I, 6) = F(I, 6) + XA*XA
      F(I, 7) = F(I, 7) + XA
      GO TO 90
   80 IF (XA .GT. F(I, 8)) F(I, 8) = XA
   90 CONTINUE
  100 F(I, 6) = SQRT(F(I, 6))
C
C     PRINT MEMBER FORCES FOR ALL LOAD CONDITIONS
C
  110 DO 200 L = 1, LLD
      IF (L .GT. NLD) GO TO 150
C     STATIC FORCE COMPONENTS
      DO 120 I = 1, NF
      P(I) = 0.
      DO 120 J = 1, 5
  120 P(I) = P(I) + F(I, J)*XM(J, L)
      IF (NATT) 130, 140, 140
  130 IF (M2 .EQ. 0) WRITE (WT,210) M, L, (P(I), I = 1, NF)
      IF (M2 .NE. 0) WRITE (WT,240) M, M2, L, (P(I), I = 1, NF)
      GO TO 200
C     DYNAMIC FORCE COMPONENTS
  140 IF (L .GT. NLD) GO TO 150
      IF ((XM(6, L) .NE. 0.) .OR. (XM(7, L) .NE. 0.) .OR. (XM(8, L) .NE.
     &0.)) GO TO 150
      IF (M2 .EQ. 0) WRITE (WT,210) M, L, (P(I), I = 1, NF)
      IF (M2 .NE. 0) WRITE (WT,240) M, M2, L, (P(I), I = 1, NF)
      GO TO 200
  150 DO 170 I = 1, NF
      IF (L .GT. NLD) GO TO 160
      PI = F(I, 6)*XM(6, L) + F(I, 7)*XM(7, L) + F(I, 8)*XM(8, L)
      PMIN(I) = P(I) - PI
      P(I) = P(I) + PI
      GO TO 170
  160 P(I) = FM(I, L - NLD)
  170 CONTINUE
      IF (L .GT. NLD) GO TO 190
      IF (M2 .NE. 0) GO TO 180
      WRITE (WT,220) M, L, HD1, (P(I), I = 1, NF)
      WRITE (WT,230) L, HD2, (PMIN(I), I = 1, NF)
      GO TO 200
  180 WRITE (WT,250) M, M2, L, HD1, (P(I), I = 1, NF)
      WRITE (WT,260) L, HD2, (PMIN(I), I = 1, NF)
      GO TO 200
  190 IF (M2 .EQ. 0) WRITE (WT,210) M, L, (P(I), I = 1, NF)
      IF (M2 .NE. 0) WRITE (WT,240) M, M2, L, (P(I), I = 1, NF)
  200 CONTINUE
C
      RETURN
C
  210 FORMAT (I6, I8, 5X, 8F14.4)
  220 FORMAT (I6, I8, 2X, A3, 8F14.4)
  230 FORMAT (6X, I8, 2X, A3, 8F14.4)
  240 FORMAT (3I8, 8X, 4F14.4)
  250 FORMAT (3I8, 5X, A3, 4F14.4)
  260 FORMAT (16X, I8, 5X, A3, 4F14.4)
      END
C     ******************************************************************
C     * THIS SUBROUTINE ALLOCATES STORAGE REQUIRED FOR THE CALCULATION *
C     *                       OF SLAB FORCES                           *
C     ******************************************************************
      SUBROUTINE SLAB(MTOT, MTOT1, NMD)
      COMMON A(1)
      INTEGER RD, WT
      COMMON /JRW/RD, WT
      COMMON /GEN/NST, NSF, NLD, NAT, NFO, SSPACE(5), IS, I3
      COMMON /DYN/NTIME, NPC, DT, DAMP
      COMMON /MDIM/NT1, NT2, NT3, NT4, NT5, NT6, MT4, MT5, NODE3
      COMMON /LSLAB/M, N, NSP, NDP, NJ, NN, NODE
      MLD = 5 + NFO
      LLD = NLD + NMD
      IF (NAT .EQ. 4) MLD = 5 + NTIME
      MM = 3*NJ
      NT7 = 3*NT2
      N1 = 2*NST + 1
      N2 = N1 + 8*NLD
      M3 = N2 + NSP*3
      N3 = M3 + 32*NSP
      N4 = N3 + NT3*NT4
      N5 = N4 + NSP
      N6 = N5 + 3*NFO
      N7 = N6 + 144*NDP
      N8 = N7 + NODE3*3*NT6
      N9 = N8 + NODE3*3*NJ
      N10 = N9 + NT7*MLD
      N11 = N10 + MM*MLD
      N12 = N11 + NT2
```

```
      N13 = N12 + N*M
      N13 = M13 + NT2
      N14 = N13 + LLD*3
      N15 = N14 + LLD*3
      N16 = N15 + NDP
      N17 = N16 + NDP
      IF (N17 .LE. MTOT) GO TO 10
      MTOT = N17
      CALL ERROR(MTOT1, MTOT, WT)
   10 CALL SLABFL(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
     &(M3), NMD, NT7, MLD, MM, LLD, A(M13))
      RETURN
      END
C     *****************************************************************
C     *   THIS SUBROUTINE COMPUTES SLAB FORCES AT REQUIRED LOCATIONS   *
C     *****************************************************************
      SUBROUTINE SLABFL(XM, IJ, IM, MMI, PM, S, SS, SC, RR, R, LH, NP,
     &PP, PMIN, A1, B1, CIJ, NMD, NT7, MLD, MM, LLD, KH)
      INTEGER RD, WT
      COMMON /JPW/RD, WT
      COMMON /JUNK/II(12), LL(4), IL(12), P(3), F(3, 8), ILL(8), MP(4),
     &MQ(4), XL(20), YL(20)
      COMMON /GEN/NST, NSF, NLD, NAT, NFO, SSPACE(5), IS, I3
      COMMON /STIF/WX(120), WY(120)
      COMMON /MDIM/NT1, NT2, NT3, NT4, NT5, NT6, MT4, MT5, NODE3
      COMMON /LSLAB/M, N, NSP, NDP, NJ, NN, NODE
      DIMENSION XM(8, NLD), IM(NT4, NT3), MMI(NSP), PM(3, NFO), SS(NODE3
     &, 3, NT6), SC(NODE3, 3, NJ), RR(NT7, MLD), R(MM, MLD), IJ(NSF, 3),
     &CIJ(NSF, 2, 16), NP(N, M), PP(LLD, 3), PMIN(LLD, 3), S(12, 12, NDP
     &), A1(NDP), B1(NDP), KH(NT2), LH(NT2)
      DATA HD1, HD2, H1, H2/3HMAX, 3HMIN, 3H X , 3H Y  /
      NATT = NAT - 3
      NN1 = NN + 1
C
C     TEST FOR XM
C
      DO 30 I = 1, 5
      DO 10 J = 1, NLD
      IF (XM(I, J) .NE. 0.) GO TO 20
      ILL(I) = I
   10 CONTINUE
      GO TO 30
   20 ILL(I) = 0
   30 CONTINUE
      DO 40 I = 6, 8
   40 ILL(I) = 0
C
      NT = NT2/NODE
      NREM = NT2 - NT*NODE
      IF (NREM .NE. 0) NT = NT + 1
C
      REWIND 9
      REWIND 11
      READ (9) NP, MMI, IM, A1, B1, LH, KH, XL, YL
C
C     GENERATE COORDINATES FOR SLAB LINES
C
      WX(1) = 0.
      WY(1) = 0.
      DX = 0.
      DY = 0.
      DO 60 I = 1, M
      DX = DX + XL(I)
      X = XL(I)/NN
      DO 50 J = 2, NN
      LP = (I - 1)*NN + J
      WX(LP) = WX(LP - 1) + X
   50 CONTINUE
      WX(LP + 1) = DX
   60 CONTINUE
      DO 80 I = 1, N
      DY = DY + YL(I)
      Y = YL(I)/NN
      DO 70 J = 2, NN
      LT = (I - 1)*NN + J
      WY(LT) = WY(LT - 1) + Y
   70 CONTINUE
      WY(LT + 1) = DY
   80 CONTINUE
      DO 90 I = 1, NDP
      A = A1(I)/NN
      B = B1(I)/NN
      L = I
      CALL MSTIF(A, B, L, S)
   90 CONTINUE
C
C
C     COMPUTE SLAB FORCES FLOOR BY FLOOR
C
C
      DO 740 I = 1, NSF
      WRITE (WT,800) IJ(I, 1)
      READ (11) R
C
C     SOLVE FOR SLAB DISPLACEMENTS AT ALL FINITE ELEMENT NODAL POINTS
C
      DO 100 J1 = 1, NT7
      DO 100 J2 = 1, MLD
  100 RR(J1, J2) = 0.
      DO 180 J = 1, NT
      BACKSPACE 14
      K1 = NODE*NT
      K2 = NODE3
      I1 = K1 - NODE*(J - 1)
      KMIN = I1 - NODE + 1
      IF (K1 .EQ. NT2 .OR. J .NE. 1) GO TO 110
      I1 = K1 - NODE + NREM
      KMIN = I1 - NREM + 1
      K2 = 3*NREM
  110 NNT = KH(I1) - KMIN + 1
      NTT = LH(I1)
      READ (14) (((SS(M2, M3, M1), M3 = 1, 3), M2 = 1, K2), M1 = 1, NNT)
     &, (((SC(M2, M3, M1), M3 = 1, 3), M2 = 1, K2), M1 = 1, NTT)
      BACKSPACE 14
      DO 170 J2 = 1, MLD
      IF (ILL(J2) .EQ. J2) GO TO 170
      DO 160 J3 = 1, K2
      L4 = K2 - J3 + 1
      I2 = I1 - (J3 - 1)/3
      L1 = J3 - (J3 - 1)/3*3
      L2 = I2*3 - L1 + 1
      L3 = L2 + 1
      NNT = KH(I2) - KMIN + 1
      NTT = LH(I2)
      I2 = I2 - KMIN + 1
      DO 130 J4 = I2, NNT
      IP = 1
      IF (J4 .EQ. I2) IP = 5 - L1
      IF (IP .GT. 3) GO TO 130
      DO 120 J5 = IP, 3
      RR(L2, J2) = RR(L2, J2) - SS(L4, J5, J4)*RR(L3, J2)
```

```
      L3 = L3 + 1
  120 CONTINUE
  130 CONTINUE
      L3 = 1
      DO 150 J4 = 1, NTT
      DO 140 J5 = 1, 3
      RP(L2, J2) = RP(L2, J2) - SC(L4, J5, J4)*R(L3, J2)
      L3 = L3 + 1
  140 CONTINUE
  150 CONTINUE
  160 CONTINUE
  170 CONTINUE
  180 CONTINUE
      IF (I .EQ. NSF) GO TO 200
      DO 190 M1 = 1, NT
  190 READ (14)
C
C     CALCULATE SLAB FORCES LINE BY LINE IN THE ORDER OF INPUT
C
  200 DO 730 L = 1, 2
      MJ = IJ(I, L + 1)
      IF (MJ .EQ. 0) GO TO 730
      IF (L .NE. 1) GO TO 210
      MD = M
      H3 = H1
      H4 = H2
      GO TO 220
  210 MD = N
      H3 = H2
      H4 = H1
  220 DO 720 J = 1, MJ
      CM = CIJ(I, L, J)
      IF (L .EQ. 2) GO TO 240
      DO 230 IW = 1, NT4
      LP = IW
      IF (ABS(CM - WY(IW)) .LE. 0.1) GO TO 260
  230 CONTINUE
      WRITE (WT,810) H2, CM
      GO TO 720
  240 DO 250 IW = 1, NT3
      LT = IW
      IF (ABS(CM - WX(IW)) .LE. 0.1) GO TO 260
  250 CONTINUE
      WRITE (WT,810) H1, CM
      GO TO 720
  260 WRITE (WT,780) H3, H4, CM, H3
      IF (L .EQ. 2) GO TO 270
      MK = (LP + NN - 1)/NN
      MC = LP - (MK - 1)*NN
      IF (MK .LE. N) GO TO 280
      MC = NN1
      MK = MK - 1
      GO TO 280
  270 MK = (LT + NN - 1)/NN
      MC = LT - (MK - 1)*NN
      IF (MK .LE. M) GO TO 280
      MC = NN1
      MK = MK - 1
  280 DO 710 I1 = 1, MD
      K4 = NN
      IF (I1 .EQ. MD) K4 = NN1
      DO 700 I2 = 1, K4
      IF (L .EQ. 1) LT = NN*I1 + I2 - NN
      IF (L .EQ. 2) LP = NN*I1 + I2 - NN
      NC = IM(LP, LT)
      IF (NC .EQ. 0) GO TO 700
      IF (L .EQ. 1) CM = WX(LT)
      IF (L .EQ. 2) CM = WY(LP)
      MP(1) = LP + 1
      MP(2) = LP - 1
      MP(3) = LP - 1
      MP(4) = LP + 1
      MO(1) = LT + 1
      MO(2) = LT + 1
      MO(3) = LT - 1
      MO(4) = LT - 1
      IC = 0
      DO 290 L1 = 1, LLD
      DO 290 L2 = 1, 3
      PMIN(L1, L2) = 0.
  290 PP(L1, L2) = 0.
      DO 620 K1 = 1, 4
      IF (MP(K1) .LT. 1 .OR. MP(K1) .GT. NT4) GO TO 620
      IF (MO(K1) .LT. 1 .OR. MO(K1) .GT. NT3) GO TO 620
      MC1 = IM(MP(K1), MO(K1))
      MC2 = IM(MP(K1), LT)
      MC3 = IM(LP, MO(K1))
      IF (MC1 .EQ. 0 .OR. MC2 .EQ. 0 .OR. MC3 .EQ. 0) GO TO 620
      IC = IC + 1
      IF (L .EQ. 1) IT = MMI(NP(MK, I1))
      IF (L .EQ. 2) IT = MMI(NP(I1, MK))
      LL(K1) = IM(LP, LT)
      GO TO (300, 310, 320, 340), K1
  300 LL(2) = IM(LP + 1, LT)
      LL(3) = IM(LP + 1, LT + 1)
      LL(4) = IM(LP, LT + 1)
      IF (L .EQ. 1 .AND. MC .EQ. NN1) IT = MMI(NP(MK + 1, I1))
      IF (L .EQ. 2 .AND. MC .EQ. NN1) IT = MMI(NP(I1, MK + 1))
      GO TO 360
  310 LL(1) = IM(LP - 1, LT)
      LL(3) = IM(LP, LT + 1)
      LL(4) = IM(LP - 1, LT + 1)
      IF (L .EQ. 1 .AND. MC .EQ. 1) IT = MMI(NP(MK - 1, I1))
      IF (L .NE. 2) GO TO 360
      IF (MC .EQ. NN1) IT = MMI(NP(I1, MK + 1))
      IF (I2 .EQ. 1) IT = MMI(NP(I1 - 1, MK))
      IF (I2 .EQ. 1 .AND. MC .EQ. NN1) IT = MMI(NP(I1 - 1, MK + 1))
      GO TO 360
  320 LL(1) = IM(LP - 1, LT - 1)
      LL(2) = IM(LP, LT - 1)
      LL(4) = IM(LP - 1, LT)
      IF (L .NE. 1) GO TO 330
      IF (MC .EQ. 1) IT = MMI(NP(MK - 1, I1))
      IF (I2 .EQ. 1) IT = MMI(NP(MK, I1 - 1))
      IF (MC .EQ. 1 .AND. I2 .EQ. 1) IT = MMI(NP(MK - 1, I1 - 1))
      GO TO 360
  330 IF (MC .EQ. 1) IT = MMI(NP(I1, MK - 1))
      IF (I2 .EQ. 1) IT = MMI(NP(I1 - 1, MK))
      IF (MC .EQ. 1 .AND. I2 .EQ. 1) IT = MMI(NP(I1 - 1, MK - 1))
      GO TO 360
  340 LL(1) = IM(LP, LT - 1)
      LL(2) = IM(LP + 1, LT - 1)
      LL(3) = IM(LP + 1, LT)
      IF (L .NE. 1) GO TO 350
      IF (MC .EQ. NN1) IT = MMI(NP(MK + 1, I1))
      IF (I2 .EQ. 1) IT = MMI(NP(MK, I1 - 1))
      IF (MC .EQ. NN1 .AND. I2 .EQ. 1) IT = MMI(NP(MK + 1, I1 - 1))
      GO TO 360
  350 IF (MC .EQ. 1) IT = MMI(NP(I1, MK - 1))
      GO TO 360
  360 DO 390 K2 = 1, 4
      K3 = 3*K2
```

```
      IF (LL(K2) .GT. 10000) GO TO 370
      II(K3) = 3*LL(K2)
      II(K3 - 1) = II(K3) - 1
      II(K3 - 2) = II(K3) - 2
      GO TO 380
  370 LL(K2) = LL(K2) - 10000
      II(K3) = 0
      II(K3 - 1) = 0
      II(K3 - 2) = 0
      IL(K3) = 3*LL(K2)
      IL(K3 - 1) = IL(K3) - 1
      IL(K3 - 2) = IL(K3) - 2
  380 CONTINUE
C
C     CALCULATE STATIC FORCES
C
  390 IR1 = 3*K1 - 3
      DO 400 L1 = 1, 3
      DO 400 L2 = 1, 8
  400 F(L1, L2) = 0.
  410 DO 440 L1 = 1, 3
      IR2 = IR1 + L1
      DO 440 L2 = 1, 5
      IF (ILL(L2) .EQ. L2) GO TO 440
      DO 430 L3 = 1, 12
      IK = II(L3)
      IF (IK .EQ. 0) GO TO 420
      F(L1, L2) = F(L1, L2) + S(IR2, L3, IT)*RR(IK, L2)
      GO TO 430
  420 IK = IL(L3)
      F(L1, L2) = F(L1, L2) + S(IR2, L3, IT)*R(IK, L2)
  430 CONTINUE
  440 CONTINUE
C
C     CALCULATE DYNAMIC FORCES
      IF (NATT) 520, 450, 450
  450 DO 510 L1 = 1, 3
      IR2 = IR1 + L1
      DO 500 L2 = 6, MLD
      X = 0.0
      DO 470 L3 = 1, 12
      IK = II(L3)
      IF (IK .EQ. 0) GO TO 460
      X = X + S(IR2, L3, IT)*RR(IK, L2)
      GO TO 470
  460 IK = IL(L3)
      X = X + S(IR2, L3, IT)*R(IK, L2)
  470 CONTINUE
      IF (NMD .GT. 0) FM(L1, L2 - 5) = X
      XA = ABS(X)
      IF (NATT) 480, 480, 490
  480 F(L1, 6) = F(L1, 6) + XA*XA
      F(L1, 7) = F(L1, 7) + XA
      GO TO 500
  490 IF (XA .GT. F(L1, 8)) F(L1, 8) = XA
  500 CONTINUE
  510 F(L1, 6) = SQRT(F(L1, 6))
  520 DO 610 L1 = 1, LLD
      IF (L1 .GT. NLD) GO TO 580
      DO 530 L2 = 1, 3
      P(L2) = 0.
      DO 530 L3 = 1, 5
  530 P(L2) = P(L2) + F(L2, L3)*XM(L3, L1)
      IF (NATT) 540, 560, 560
  540 DO 550 L2 = 1, 3
  550 PP(L1, L2) = PP(L1, L2) + P(L2)
      GO TO 610
  560 IF ((XM(6, L1) .NE. 0) .OR. (XM(7, L1) .NE. 0.) .OR. (XM(8, L1)
     &.NE. 0.)) GO TO 580
      DO 570 L2 = 1, 3
  570 PP(L1, L2) = PP(L1, L2) + P(L2)
      GO TO 610
  580 DO 600 L2 = 1, 3
      IF (L1 .GT. NLD) GO TO 590
      PI = F(L2, 6)*XM(6, L1) + F(L2, 7)*XM(7, L1) + F(L2, 8)*XM(8, L1)
      PMIN(L1, L2) = PMIN(L1, L2) + P(L2) - PI
      PP(L1, L2) = PP(L1, L2) + P(L2) + PI
      GO TO 600
  590 PP(L1, L2) = PP(L1, L2) + FM(L2, L1 - NLD)
  600 CONTINUE
  610 CONTINUE
  620 CONTINUE
C
C     PRINT SLAB FORCES FOR ALL LOAD CONDITIONS
C
      DO 690 L1 = 1, LLD
      DO 630 L2 = 1, 3
  630 PP(L1, L2) = PP(L1, L2)/IC
      IF (NATT) 640, 650, 650
  640 WRITE (WT,750) CM, L1, (PP(L1, L2), L2 = 1, 3)
      GO TO 690
  650 IF (L1 .GT. NLD) GO TO 660
      IF ((XM(6, L1) .NE. 0.) .OR. (XM(7, L1) .NE. 0.) .OR. (XM(8, L1)
     &.NE. 0.)) GO TO 670
  660 WRITE (WT,750) CM, L1, (PP(L1, L2), L2 = 1, 3)
      GO TO 690
  670 DO 680 L2 = 1, 3
  680 PMIN(L1, L2) = PMIN(L1, L2)/IC
      WRITE (WT,760) CM, L1, HD1, (PP(L1, L2), L2 = 1, 3)
      WRITE (WT,770) L1, HD2, (PMIN(L1, L2), L2 = 1, 3)
  690 CONTINUE
      WRITE (WT,790)
  700 CONTINUE
  710 CONTINUE
  720 CONTINUE
  730 CONTINUE
  740 CONTINUE
      RETURN
  750 FORMAT (F9.2, 5X, I5, 8X, 3F14.4)
  760 FORMAT (F9.2, 5X, I5, 5X, A3, 3F14.4)
  770 FORMAT (14X, I5, 5X, A3, 3F14.4)
  780 FORMAT (//' SLAB FORCE IN ', A3, ' DIRECTION , ', ' ALONG ORDINAT
     &E ', A3, ' =', F8.2//A2, '-ORDINATE', 4X, 'LOADING', 10X, 'MOMENT-
     &MX', 5X, 'MOMENT-MY', 4X, 'MOMENT-MXY'/13X, 'COMBINATION')
  790 FORMAT (1H )
  800 FORMAT ('1', 41X, '........ SLAB FORCES   LEVEL NO.', I4, ' ......
     &..'/)
  810 FORMAT (//' SLAB LINE ALONG', A3, '=', F8.2, '  IS NOT A FINITE EL
     &EMENT DIVISION LINE')
      END
C     ******************************************************************
C     *   THIS SUBROUTINE GENERATES STRESS MATRIX FOR A RECTANGULAR    *
C     *                     FINITE ELEMENT                             *
C     ******************************************************************
      SUBROUTINE MSTIF(A, B, L, S)
      COMMON /LSLAB/M, N, NSP, NDP
      COMMON /TEV/T, E, V
      DIMENSION S(12, 12, NDP)
      DO 10 I = 1, 12
      DO 10 J = 1, 12
   10 S(I, J, L) = 0.
      C1 = E*T**3/(12.*A*B*(1. - V*V))
```

```
      C2 = 6.*A/B
      C3 = 6.*B/A
      S(1, 1, L) = - 4.*V*A*C1
      S(1, 2, L) = 4.*B*C1
      S(1, 3, L) = ( - C3 - C2*V) *C1
      S(1, 4, L) = S(1, 1, L)/2.
      S(1, 6, L) = C2*V*C1
      S(1, 11, L) = 2.*B*C1
      S(1, 12, L) = C3*C1
      S(2, 1, L) = - 4.*A*C1
      S(2, 2, L) = 4.*B*V*C1
      S(2, 3, L) = ( - C2 - C3*V) *C1
      S(2, 4, L) = - 2.*A*C1
      S(2, 6, L) = C2*C1
      S(2, 11, L) = S(2, 2, L)/2.
      S(2, 12, L) = C3*V*C1
      S(3, 1, L) = B*(1. - V)*C1
      S(3, 2, L) = - A*(1. - V)*C1
      S(3, 3, L) = (1. - V)*C1
      S(3, 5, L) = - S(3, 2, L)
      S(3, 6, L) = - S(3, 3, L)
      S(3, 9, L) = S(3, 3, L)
      S(3, 10, L) = - S(3, 1, L)
      S(3, 12, L) = - S(3, 3, L)
      S(4, 1, L) = - S(1, 4, L)
      S(4, 3, L) = S(1, 6, L)
      S(4, 4, L) = - S(1, 1, L)
      S(4, 5, L) = S(1, 2, L)
      S(4, 6, L) = S(1, 3, L)
      S(4, 8, L) = S(1, 11, L)
      S(4, 9, L) = C3*C1
      S(5, 1, L) = - S(2, 4, L)
      S(5, 3, L) = C2*C1
      S(5, 4, L) = - S(2, 1, L)
      S(5, 5, L) = S(2, 2, L)
      S(5, 6, L) = S(2, 3, L)
      S(5, 8, L) = S(2, 11, L)
      S(5, 9, L) = S(2, 12, L)
      S(6, 2, L) = S(3, 2, L)
      S(6, 3, L) = S(3, 3, L)
      S(6, 4, L) = S(3, 1, L)
      S(6, 5, L) = S(3, 5, L)
      S(6, 6, L) = S(3, 12, L)
      S(6, 7, L) = S(3, 10, L)
      S(6, 9, L) = S(3, 3, L)
      S(6, 12, L) = S(3, 12, L)
      S(7, 5, L) = - S(1, 11, L)
      S(7, 6, L) = C3*C1
      S(7, 7, L) = - S(1, 1, L)
      S(7, 8, L) = - S(1, 2, L)
      S(7, 9, L) = S(1, 3, L)
      S(7, 10, L) = - S(1, 4, L)
      S(7, 12, L) = S(1, 6, L)
      S(8, 5, L) = - S(2, 11, L)
      S(8, 6, L) = S(2, 12, L)
      S(8, 7, L) = - S(2, 1, L)
      S(8, 8, L) = - S(2, 2, L)
      S(8, 9, L) = S(2, 3, L)
      S(8, 10, L) = - S(2, 4, L)
      S(8, 12, L) = C2*C1
      S(9, 3, L) = S(3, 3, L)
      S(9, 4, L) = S(3, 1, L)
      S(9, 6, L) = S(3, 6, L)
      S(9, 7, L) = S(3, 10, L)
      S(9, 8, L) = S(3, 5, L)
      S(9, 9, L) = S(3, 3, L)
      S(9, 11, L) = S(3, 2, L)
      S(9, 12, L) = S(3, 6, L)
      S(10, 2, L) = - S(1, 11, L)
      S(10, 3, L) = C3*C1
      S(10, 7, L) = S(1, 4, L)
      S(10, 9, L) = S(1, 6, L)
      S(10, 10, L) = S(1, 1, L)
      S(10, 11, L) = - S(1, 2, L)
      S(10, 12, L) = S(1, 3, L)
      S(11, 2, L) = - S(2, 11, L)
      S(11, 3, L) = S(2, 12, L)
      S(11, 7, L) = S(2, 4, L)
      S(11, 9, L) = C2*C1
      S(11, 10, L) = S(2, 1, L)
      S(11, 11, L) = - S(2, 2, L)
      S(11, 12, L) = S(2, 3, L)
      S(12, 1, L) = S(3, 1, L)
      S(12, 3, L) = S(3, 3, L)
      S(12, 6, L) = S(3, 6, L)
      S(12, 8, L) = S(3, 5, L)
      S(12, 9, L) = S(3, 9, L)
      S(12, 10, L) = S(3, 10, L)
      S(12, 11, L) = S(3, 2, L)
      S(12, 12, L) = S(3, 12, L)
      RETURN
      END
C     *****************************************************************
C     *     THIS SUBROUTINE CONVERTS INPUT DATA INTO REAL NUMBERS     *
C     *****************************************************************
      SUBROUTINE CNVRT(I1, I2)
      INTEGER*2 INPT(80), IN(5)/'0', '9', '.', ' ', '-'/
      COMMON /INPUT/INPT, FN(8)
      J = 10*I2 + 1
      IFL = I2 - I1 + 1
   10 FK = 0.
      K = 1
      L = 0
      DO 50 I = 1, 10
      J = J - 1
      IF (INPT(J) .NE. IN(3)) GO TO 20
      FK = FK/10.**L
      L = 0
      GO TO 50
   20 IF (INPT(J) .LT. IN(1) .OR. INPT(J) .GT. IN(2)) GO TO 30
      IJ = INPT(J)/256 + 15
      L = L + 1
      FK = FK + IJ*10.**L/10.
      GO TO 50
   30 IF (INPT(J) .NE. IN(5)) GO TO 40
      FK = - FK
      GO TO 50
   40 IF (INPT(J) .EQ. IN(4)) GO TO 50
      K = 0
   50 CONTINUE
   60 FN(IFL) = FK*K
      IFL = IFL - 1
      IF (IFL .GT. 0) GO TO 10
      RETURN
      END
```

A P P E N D I X   D

DYNAMIC AND EARTHQUAKE ANALYSIS

D.1 Mass Approximation

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 described 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 motion of each floor level, as previously described for the formulation of the structure stiffness matrix. Correspondingly, the mass of the building is lumped at each floor level. With this lumped parameter idealization, equilibrium of the structure is described by a set of ordinary second order differential equations.

D.2 Dynamic Equilibrium Equations

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

$$M\ddot{r}_a + C\dot{r} + Kr = P(t) \tag{D.1}$$

where M = mass matrix

C = damping matrix

K = stiffness matrix

$P(t)$ = applied load vector, which may be time varying

r = displacement vector of deformation relative to support motion

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

r and r_a are related in the following fashion

$$r_a = V_g + r \tag{D.2a}$$

where V_g is the vector of peseudo-static displacements due to support movement. Also

$$\ddot{r}_a = \ddot{V}_g + \ddot{r} \qquad \text{(D.2a)}$$

These vectors have the following form for a typical floor, of a building shown in Figure D.1 below

$$\begin{Bmatrix} r_{xa} \\ r_{ya} \\ r_{\theta a} \end{Bmatrix}_n = \begin{Bmatrix} V_{gx} \\ V_{gy} \\ V_{g\theta} \end{Bmatrix} + \begin{Bmatrix} r_{xn} \\ r_{yn} \\ r_{\theta n} \end{Bmatrix} = \begin{Bmatrix} \sin \\ \cos \\ 0 \end{Bmatrix} V_g + \begin{Bmatrix} r_{xn} \\ r_{yn} \\ r_{\theta n} \end{Bmatrix} \qquad \text{(D.3a)}$$

and

$$\begin{Bmatrix} \ddot{r}_{xa} \\ \ddot{r}_{ya} \\ \ddot{r}_{\theta a} \end{Bmatrix}_n = \begin{Bmatrix} \sin\beta \\ \cos\beta \\ 0 \end{Bmatrix} \ddot{V}_g + \begin{Bmatrix} \ddot{r}_{xn} \\ \ddot{r}_{yn} \\ \ddot{r}_{\theta n} \end{Bmatrix} \qquad \text{(D.3b)}$$

i.e. $\quad r_{na} = b\ddot{V}_g + r_n \qquad$ (D.3c)

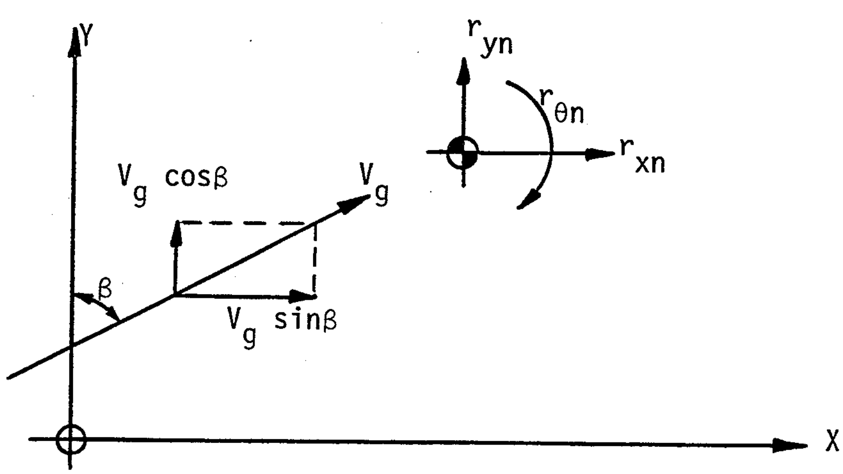


FIGURE D1 GROUND AND STRUCTURE DISPLACEMENTS

Or, for all floors

$$r_a = B\ddot{v}_g + r \tag{D.3d}$$

where

$$B = \begin{Bmatrix} b_1 \\ b_2 \\ b_3 \\ \cdot \\ \cdot \\ b_N \end{Bmatrix} \quad ; b_1 = b_2 \text{ etc.} \tag{D.3e}$$

In the case of seismic analysis, there are no externally applied loads; i.e. $P(t) = 0$. Then equation (D.1) may be written as

$$M(\ddot{r} + B\,\ddot{V}_g) + C\dot{r} + Kr = 0 \tag{D.4a}$$

or

$$M\ddot{r} + C\dot{r} + Kr = -MB\ddot{v}_g \tag{D.4b}$$

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 eigenvectors or mode shapes of the system.

D.3 Mode Shapes and Frequencies

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

$$M\ddot{r} + Kr = 0 \tag{D.5}$$

The eigenvalue problem to be solved is written as

$$K\phi = W^2 M\phi \tag{D.6}$$

where

ϕ = mode shapes

W = frequencies

The mode shapes are normalized such that

$$\phi^T M\phi = I \tag{D.7a}$$

then also

$$\phi^T K\phi = W^2 \tag{D.7b}$$

Also, it is assumed that the damping matrix C is of a form that is uncoupled by the mode shapes; specifically it is assumed that

$$\phi^T C\phi = [2\lambda_m W_m] \tag{D.7c}$$

so that λ_m represents the damping of the mth mode.

The actual displacements, r, are now expressed as a linear combination of the mode shapes.

$$r = [\phi_1 \quad \phi_2 \quad \phi_3 \quad . \quad . \quad . \quad \phi_N] \begin{Bmatrix} Z_1(t) \\ Z_2(t) \\ . \\ . \\ . \\ Z_N(t) \end{Bmatrix} \tag{D.8a}$$

i.e. $$r = \phi Z \tag{D.8b}$$

also $$\dot{r} = \phi \dot{Z} \tag{D.8c}$$

and $$\ddot{r} = \phi \ddot{Z} \tag{D.8d}$$

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

D.4 Dynamic Response Analysis

Using equation (D.8), equation (D.4b) may be rewritten as

$$M\phi\ddot{Z} + C\phi\dot{Z} + K\phi Z = -MB\ddot{V}_g \qquad (D.9)$$

Premultiplication by ϕ^T yields the uncoupled set of second order equations.

$$M^*\ddot{Z} + C^*\dot{Z} + K^*Z = P^*\ddot{V}_g \qquad (D.10)$$

where

$$M^* = \phi^T M\phi = I \qquad (D.11a)$$

$$C^* = \phi^T C\phi = [2\lambda_m W_m] \qquad (D.11b)$$

$$K^* = \phi^T K\phi = [W^2_m] \qquad (D.11c)$$

$$P^*\ddot{V}_g = \phi^T MB\ddot{V}_g \qquad (D.11d)$$

to find the form of P*, consider

$$MB = \begin{bmatrix} m_1 & & & & & & & & \\ & m_2 & & & & & & & \\ & & J_1 & & & & & & \\ & & & m_2 & & & & & \\ & & & & m_2 & & & & \\ & & & & & J_2 & & & \\ & & & & & & \cdot & & \\ & & & & & & & \cdot & \\ & & & & & & & & \cdot \\ & & & & & & & & & J_N \end{bmatrix} \begin{Bmatrix} \sin\beta \\ \cos\beta \\ 0 \\ \sin\beta \\ \cos\beta \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{Bmatrix} \qquad (D.12a)$$

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

$$\ddot{V}_g = A + Bt \qquad (D.15a)$$

where A and B are computed from the end values as shown on Figure D.2.

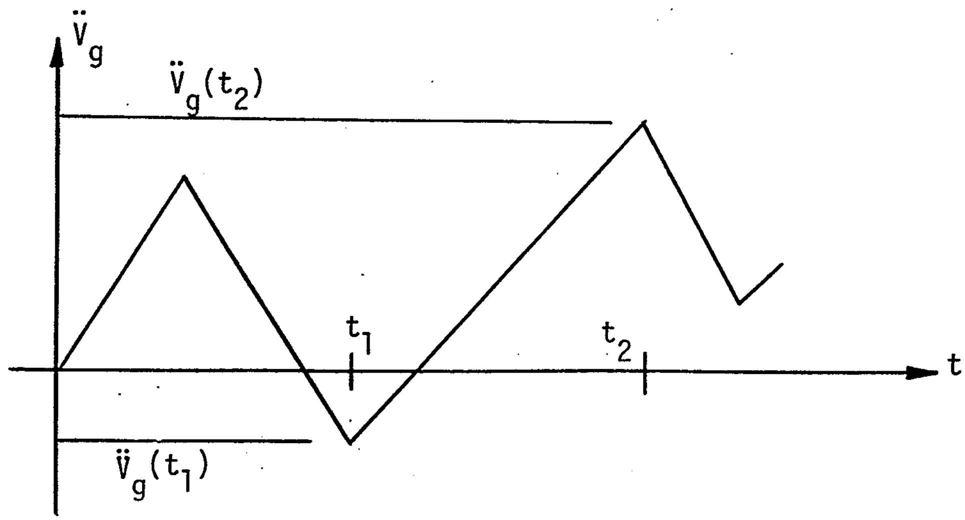


FIGURE D.2 GROUND ACCELERATION

on (t_1,t_2) ; $\quad A = V_g(t_1)$ (D.15b)

$$B = \frac{\ddot{V}_g(t_2)-\ddot{V}_g(t_1)}{t_2-t_1} \qquad \text{(D.15c)}$$

where m_1 = mass of storey 1

J_1 = rotational mass moment of inertia of storey 1

i.e.

$$MB = \left\{ \begin{matrix} m_1 \sin\beta \\ m_1 \cos\beta \\ 0 \\ m_2 \sin\beta \\ m_2 \cos\beta \\ 0 \\ \cdot \\ \cdot \\ \cdot \end{matrix} \right\} \qquad \text{(D.12b)}$$

So, a typical term of P^* has the form

$$P^*_m = \phi^T_m MB \tag{D.13a}$$

or

$$p^*_m = [\phi_{1x}\ \phi_{1y}\ \phi_{1\theta}\ \phi_{2x}\ \phi_{2y}\ \phi_{2\theta}\ \cdot\ \cdot\ \cdot\ ] \begin{Bmatrix} m_1 \sin\beta \\ m_1 \cos\beta \\ 0 \\ m_2 \sin\beta \\ m_2 \cos\beta \\ 0 \\ \cdot \\ \cdot \\ \cdot \end{Bmatrix} \tag{D.13b}$$

$$P^*_m = \sum_{n=1}^{N} m_n \{ \sin\beta\phi_{nx} + \cos\beta\phi_{ny}\} \tag{D.13c}$$

Now a typical equation governing the response in the mth mode has the form

$$\ddot{Z}_m + 2\lambda_m W_m \dot{Z}_m + W_m^2 Z_m = P^*_m \ddot{V}_g \tag{D.14}$$

Now on (t_1,t_2) equation (D.14) becomes

$$\ddot{Z}_m + 2\lambda_m W_m \dot{Z}_m + W_m^2 Z_m = P^*_m (A + bt) \tag{D.16}$$

The solution to equation (3.16) on (t_1,t_2) is given by

$$Z_m(t) = P^*_m e^{-\lambda_m W_m t} \quad Z_m(t_1) - \frac{A}{W_m^2} + \frac{2\lambda_{mB}}{W_m^3} \quad \cos W_{Dm} t$$

$$+ \frac{1}{W_{DM}} \dot{Z}_m(t_1) + \lambda_m W_m Z_m(t_1) - \frac{\lambda m A}{W_m} + \frac{B(2\lambda_m^2-1)}{W_m^2} \sin W_{Dm} t$$

$$+ P_m^* \frac{A}{W_m^2} - \frac{2\lambda_m B}{W_m^3} + \frac{Bt}{W_m^2} \qquad \text{(D.17a)}$$

where

$$W_{Dm} = W_m \, (1-\lambda_m^2)^{1/2} \qquad \text{(D.17b)}$$

and $Z_m\,(t_1)$, $\dot{Z}_m(t_1)$ are the initial conditions for this linear portion. Differentiation of equation (D.17a) gives the modal velocity

$$\dot{Z}_m(t) = P_m^* \, e^{-\lambda_m W_m t} \left\{ \left[\dot{Z}_m(t_1) - \frac{B}{W_m^2} \right] \cos W_{DM} t \right.$$

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

$$+ P_m^* \frac{B}{W_m^2} \qquad \text{(D.18)}$$

At rest initial conditions are used for the first linear portion. Equation (D.17) and (D.18) are used to compute the end values which become the initial conditions for the second linear portion. Repetition gives the complete solution over the required time interval. With solutions for each mode, equation (D.8) is used to give the structure displacements r as a function of time. Member forces follow as in section 4.3 of Chapter IV.

D.5 Spectrum Analysis

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

The response of a unit mass system with damping λ, and frequency W, is governed by the equation

$$\ddot{u}(t) + 2\lambda W\dot{u}(t) + W^2u(t) = \ddot{V}_g(t) \qquad (D.19)$$

Let u_{max} be the maximum value that u(t) attains. Then, three spectral quantities are defined by

i) spectral displacement: $S_d\ (W,\lambda) \equiv u_{max}$

ii) spectral velocity: $S_v\ (W,\lambda) \equiv wu_{max}$

iii) spectral acceleration: $S_a\ (W,\lambda) \equiv W^2u_{max}$

So, for a specific earthquake, for a series of damping values, either spectral quantity may be evaluated and plotted against frequency or period. Although spectral displacement is the most directly useful, spectral acceleration is generally used as it gives a measure of effective acceleration and may be expressed as a dimensionless fraction of gravity.

Recalling equation (D.14) for the mth mode, in terms of spectrum acceleration, the maximum response is given by

$$Z_m^{(max)} = \frac{P_m^* S_a\ (W_m,\lambda_m)}{W_m^2} \qquad (D.20)$$

This implies a set of actual displacements

$$r_m = Z_m^{(max)}\ \phi_m \qquad (D.21)$$

and a corresponding set of member forces.

The maximum in each mode will generally occur at different times. A good estimation of the maximum displacements and member forces is made by calculating the root-mean-square of the maximum modal values.

D.6 Computer Program Dynamic Options

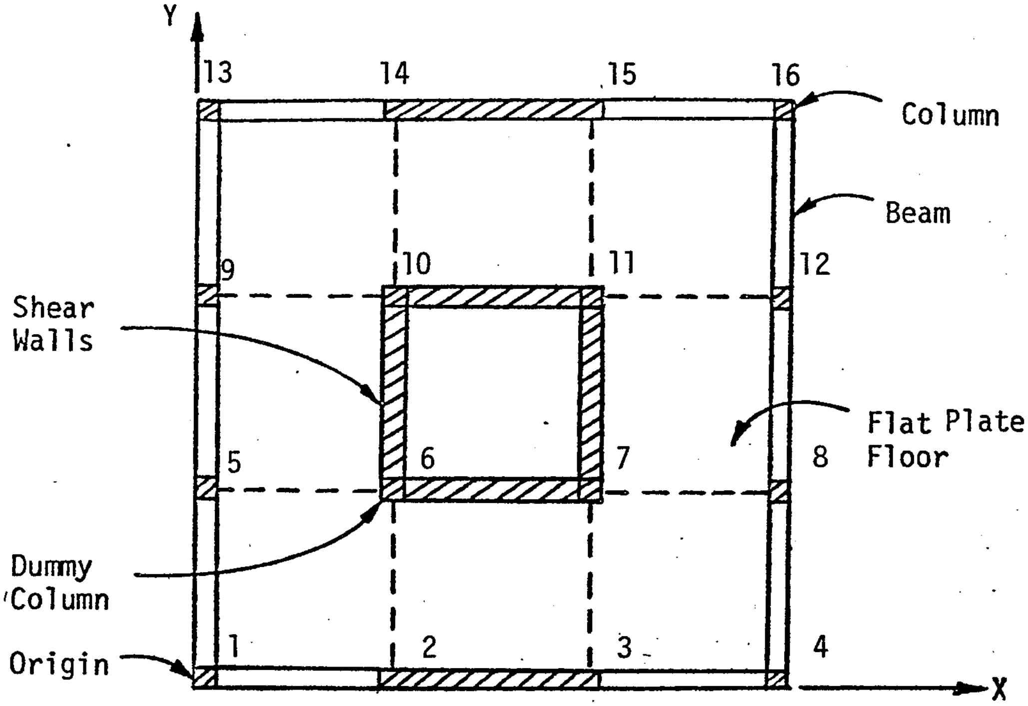
The options available in the program are:

1. Calculation of mode shapes and periods (frequencies).
2. Response spectrum analysis for any acceleration spectrum supplied by the user with
   (a) Root-Mean-Square Modal Combination
   (b) Sum of absolute value modal combinations
3. Time history analysis for any ground motion supplied by the user.

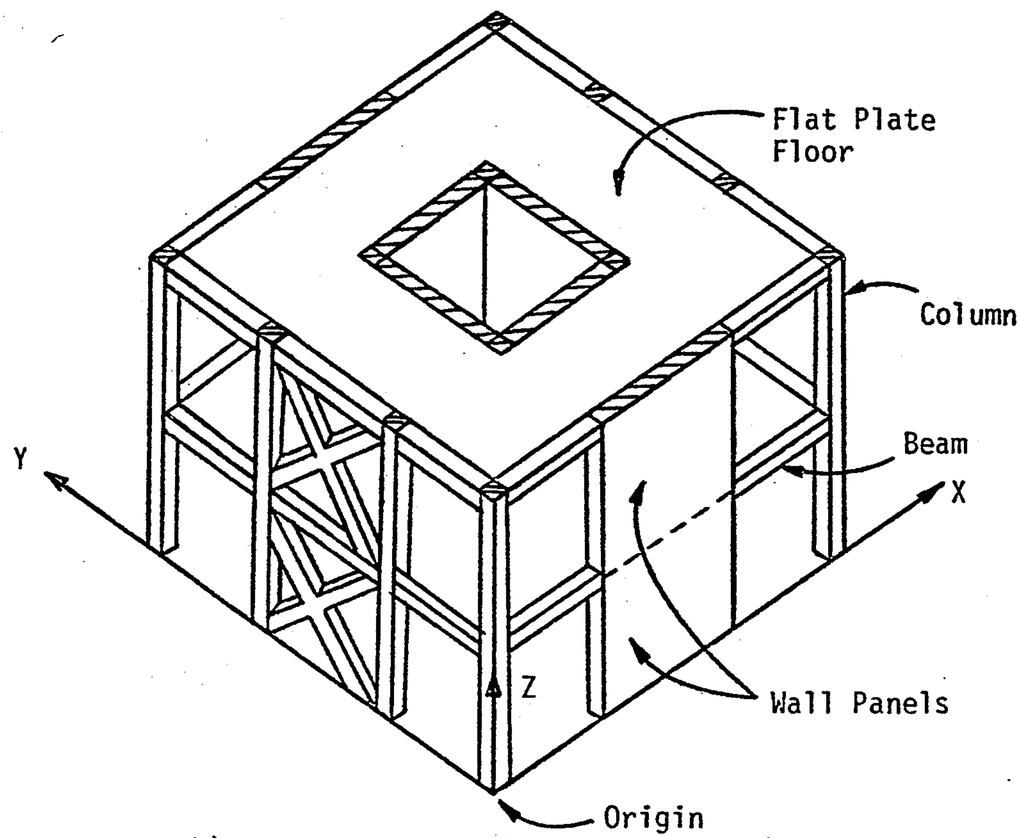
Option 2b is supplied as a matter of interest to give an upper bound on the maximum values. Either dynamic analysis may be combined with any static load case.

A P P E N D I X E

EXAMPLE INPUT AND OUTPUT



a) Floor Plan



b) Three-dimensional View

FIGURE E1 SKETCH OF TYPICAL BUILDING

INPUT DATA

```
TWO-STOREY SHEAR WALL-FRAME BUILDING
2          3          1          16
3          3          8          1          8          0          2
2          10
1000                             60         1000                  30         30
1          10
                                            500                   30         30
20         20         20
20         20         20
1          1          1          1          0          1          1          1
1
.5         576000     .2
16         1          2          1          6          2          8
1          576000     1          .84        .84        0          .0835      .0835
0          0
1          576000     1.25       1          .27
0          0
2          576000                           1000000
0          0
1          576000     10         333.33     8          230000
2          576000     5          41.7       .42        230000
1                                                                 1.2
1          1          2          1          1          1
2          2          3          2          1
3          3          4          1          1          1
4          13         14         1          1          1
5          14         15         2          1
6          15         16         1          1          1
7          1          5          1          1          1
8          5          9          1          1          1
9          9          13         1          1          1
10         4          8          1          1          1
11         8          12         1          1          1
12         12         16         1          1          1
13         6          7          2          1
14         10         11         2          1
15         6          10         2          1
16         7          11         2          1
1          1          1
2          0          1
3          0          1
4          1          1
5          1          1
6          0          1
7          0          1
8          1          1
9          1          1
10         0          1
11         0          1
12         1          1
13         1          1
14         0          1
15         0          1
16         1          1
1          2          3          1          1
2          14         15         1          1
3          6          7          2          1
4          10         11         2          1
5          6          10         2          1
6          7          11         2          1
2          9          5          576000     .25
2          5          9          576000     .25
2          12         8          576000     .25
2          8          12         576000     .25
1          9          5          576000     .25
1          5          9          576000     .25
1          12         8          576000     .25
1          8          12         576000     .25
                                 1
                                            1
1                                           1
2          3          3
10         40         60
10         30         50
```

LATERAL AND/OR GRAVITY LOAD ANALYSIS OF SHEAR WALL-FRAME STRUCTURE WITH FLAT PLATE FLOORS

JOB TITLE : TWO-STOREY SHEAR WALL-FRAME BUILDING

TYPE OF ANALYSIS : STATIC ANALYSIS ONLY

.... DESCRIPTION OF STRUCTURE ....

STORY DATA

LEVEL NO	STORY HEIGHT
2	10.00
1	10.00

TYPICAL FLOOR PLAN
(SHOWING FLOOR PANEL AND COLUMN NUMBERING)

```
PANEL THICKNESS T =          0.50
ELASTIC MODULUS E =     576000.00
POSION RATIO    V =          0.20
COLUMNS ....................... ----5----
```

```
            Y
            I
            I
            I
            13----------------14----------------15----------------16
            I                 I                 I                 I
            I                 I                 I                 I
    20.00   I        6        I        7        I        8        I
            I                 I                 I                 I
            I                 I                 I                 I
            9-----------------10----------------11----------------12
            I                 I                 I                 I
            I                 I                 I                 I
    20.00   I        4        I                 I        5        I
            I                 I                 I                 I
            I                 I                 I                 I
            5-----------------6-----------------7-----------------8
            I                 I                 I                 I
            I                 I                 I                 I
    20.00   I        1        I        2        I        3        I
            I                 I                 I                 I
            I                 I                 I                 I
            1-----------------2-----------------3-----------------4 ------X

                  20.00             20.00             20.00
```

FRAME DATA

COLUMN PROPERTIES

COLUMN TYPE	ELASTIC MODULUS	CROSS-SECT AREA	SHEAR AREA Y	SHEAR AREA X	TORSIONAL INERTIA	MOMENT OF INERTIA X-X	MOMENT OF INERTIA Y-Y	DEPTH OF FINITE JOINT COL TOP	DEPTH OF FINITE JOINT COL BTM
1	576000.00	1.00	0.84	0.84	0.0	0.08	0.08	0.0	0.0

BEAM PROPERTIES

BEAM TYPE	ELASTIC MODULUS	SHEAR AREA	TORSIONAL INERTIA	FLEXURAL INERTIA	BEAM STIFFNESS COEFICIENTS KII	KJJ	KIJ=KJI	WIDTH OF FINITE JOINT END I	END J
1	576000.00	1.25	1.00	0.27	4.00	4.00	2.00	0.0	0.0
2	576000.00	0.0	0.0	1000000.00	4.00	4.00	2.00	0.0	0.0

WALL PANEL PROPERTIES

WALL TYPE	ELASTIC MODULUS	CROSS-SECT AREA	MOMENT INERTIA	SHEAR AREA	SHEARING MODULUS
1	576000.00	10.00	333.33	8.00	230000.00
2	576000.00	5.00	41.70	0.42	230000.00

BEAM LOCATIONS

LEVEL	BM TYPE	COL I	COL J	GRAVITY LOADING TYPE (SEE BELOW) I	II	III
2	1	1	2	1	0	0
2	2	2	3	0	0	0
2	1	3	4	1	0	0
2	1	13	14	1	0	0
2	2	14	15	0	0	0
2	1	15	16	1	0	0
2	1	1	5	1	0	0
2	1	5	9	1	0	0
2	1	9	13	1	0	0
2	1	4	8	1	0	0
2	1	8	12	1	0	0
2	1	12	16	1	0	0
2	2	6	7	0	0	0
2	2	10	11	0	0	0
2	2	6	10	0	0	0
2	2	7	11	0	0	0
1	1	1	2	1	0	0
1	2	2	3	0	0	0
1	1	3	4	1	0	0
1	1	13	14	1	0	0
1	2	14	15	0	0	0
1	1	15	16	1	0	0
1	1	1	5	1	0	0
1	1	5	9	1	0	0
1	1	9	13	1	0	0
1	1	4	8	1	0	0
1	1	8	12	1	0	0
1	1	12	16	1	0	0
1	2	6	7	0	0	0
1	2	10	11	0	0	0
1	2	6	10	0	0	0
1	2	7	11	0	0	0

COLUMN LOCATIONS

LEVEL	COLUMN	COL TYPE
2	1	1
2	4	1
2	5	1
2	8	1
2	9	1
2	12	1
2	13	1
2	16	1
1	1	1
1	4	1
1	5	1
1	8	1
1	9	1
1	12	1
1	13	1
1	16	1

WALL PANEL LOCATIONS

LEVEL	WALL TYPE	COL I	COL J
2	1	2	3
2	1	14	15
2	2	6	7
2	2	10	11
2	2	6	10
2	2	7	11
1	1	2	3
1	1	14	15
1	2	6	7
1	2	10	11
1	2	6	10
1	2	7	11

BRACING ELEMENT LOCATIONS AND PROPERTIES

LEVEL	TOP COL	BTM COL	ELASTIC MODULUS	CROSS-SECT AREA
2	9	5	576000.00	0.25
2	5	9	576000.00	0.25
2	12	8	576000.00	0.25
2	8	12	576000.00	0.25
1	9	5	576000.00	0.25
1	5	9	576000.00	0.25
1	12	8	576000.00	0.25
1	8	12	576000.00	0.25

.... LOADING INFORMATION ....

TWO LATERAL LOADING CASES DESIGNATED A AND B,
THREE GRAVITY LOADING CASES DESIGNATED I, II AND III, PERMITTED

STORY LATERAL LOADS ...CASES A AND B

LEVEL NO	LATERAL LOAD CASE A F-X	F-Y	LATERAL LOAD CASE B F-X	F-Y	LOCATION OF POINT OF APPLICATION XA	YA	XB	YB
2	1000.00	0.0	1000.00	0.0	0.0	60.00	30.00	30.00
1	0.0	0.0	500.00	0.0	0.0	0.0	30.00	30.00

BEAM GRAVITY LOADING INFORMATION
(SEE BEAM LOCATION TABLE FOR LOADING LOCATIONS)

LOAD TYPE	FIXED END FORCES-END I MOMENT	SHEAR	FIXED END FORCES-END J MOMENT	SHEAR	UNIFORMLY DIST. LOAD
1	0.0	0.0	0.0	0.0	1.200

.... STRUCTURAL DISPLACEMENTS FOR FIVE BASIC LOADING CASES ....

LEVEL NO	DIRECTION	GRAVITY LOADING CASE I	II	III	LATERAL LOADING CASE A	B
2	X	0.000000	0.0	0.0	0.000413	0.012914
2	Y	0.000000	0.0	0.0	0.010334	0.000000
2	ROTN	-0.000000	0.0	0.0	-0.000344	-0.000000
1	X	0.000000	0.0	0.0	0.000149	0.005978
1	Y	0.000000	0.0	0.0	0.004186	0.000000
1	ROTN	-0.000000	0.0	0.0	-0.000140	-0.000000

.... LOADING COMBINATIONS ....

COMBINATION	GRAVITY LOADING TYPE I	II	III	LATERAL LOADING TYPE A	B
1	0.0	0.0	0.0	1.00	0.0
2	0.0	0.0	0.0	0.0	1.00
3	1.00	0.0	0.0	0.0	1.00

.... LATERAL FRAME DISPLACEMENTS FOR LOADING COMBINATION 1 2 3 ETC ....

LEVEL	DIRECTION	1	2	3
2	X	0.0004132	0.0129143	0.0129143
2	Y	0.0103337	0.0000000	0.0000000
2	ROTN	-0.0003445	-0.0000000	-0.0000000
1	X	0.0001491	0.0059779	0.0059779
1	Y	0.0041856	0.0000000	0.0000000
1	ROTN	-0.0001395	-0.0000000	-0.0000000

.... MEMBER FORCES LEVEL NO. 1 ....

COLUMN FORCES

COLUMN	LOADING COMBINATION	TORSIONAL MOMENT	X-X AXIS TOP MOMENT	X-X AXIS BTM MOMENT	AXIAL FORCE	Y-Y AXIS TOP MOMENT	Y-Y AXIS BTM MOMENT	Y-Y AXIS SHEAR	X-X AXIS SHEAR
1	1	0.0	6.4482	9.1518	2.8960	0.5724	0.0664	1.5600	0.0639
1	2	0.0	-0.3812	-0.1865	5.8430	-10.9563	-13.9254	-0.0568	-2.4882
1	3	0.0	-8.6974	-4.2560	-36.9267	-3.5778	-10.3148	-1.2953	-1.3893
CCLUMN 2 IS A DUMMY COLUMN									
COLUMN 3 IS A DUMMY COLUMN									
4	1	0.0	-6.4482	-9.1518	-2.8960	0.5724	0.0664	-1.5600	0.0639
4	2	0.0	0.3812	0.1865	-5.8430	-10.9563	-13.9254	0.0568	-2.4882
4	3	0.0	-7.9351	-3.8830	-48.6126	-18.3348	-17.5360	-1.1818	-3.5871
5	1	0.0	8.0585	9.9398	36.5358	-1.4052	-4.8989	1.7998	-0.6304
5	2	0.0	-0.0193	-0.0094	0.6226	-5.5852	-11.2971	-0.0029	-1.6882
5	3	0.0	1.2130	0.5936	-48.2887	-4.6003	-10.8152	0.1807	-1.5415
COLUMN 6 IS A DUMMY COLUMN									
CCLUMN 7 IS A DUMMY COLUMN									
8	1	0.0	-8.0585	-9.9398	-36.5358	-1.4052	-4.8989	-1.7998	-0.6304
8	2	0.0	0.0193	0.0094	-0.6226	-5.5852	-11.2371	0.0029	-1.6882
8	3	0.0	1.2516	0.6124	-49.5339	-6.5701	-11.7791	0.1864	-1.8349
9	1	0.0	8.0927	9.9565	-35.4460	-4.6377	-10.4783	1.8049	-1.5116
9	2	0.0	0.0193	0.0094	0.6226	-5.5852	-11.2971	0.0029	-1.6882
9	3	0.0	-1.2130	-0.5936	-48.2887	-4.6003	-10.8152	-0.1907	-1.5415
COLUMN 10 IS A DUMMY COLUMN									
COLUMN 11 IS A DUMMY COLUMN									
12	1	0.0	-8.0927	-9.9565	35.4460	-4.6377	-10.4783	-1.8049	-1.5116
12	2	0.0	-0.0193	-0.0094	-0.6226	-5.5852	-11.2971	-0.0029	-1.6882
12	3	0.0	-1.2516	-0.6124	-49.5339	-6.5701	-11.7791	-0.1864	-1.8349
13	1	0.0	7.0935	9.4676	7.3146	-15.5072	-19.7948	1.6561	-3.5302
13	2	0.0	0.3812	0.1865	5.8430	-10.9563	-13.9254	0.0568	-2.4982
13	3	0.0	8.6974	4.2560	-36.9267	-3.5778	-10.3148	1.2953	-1.3893
CCLUMN 14 IS A DUMMY COLUMN									
COLUMN 15 IS A DUMMY COLUMN									
16	1	0.0	-7.0935	-9.4676	-7.3146	-15.5072	-19.7948	-1.6561	-3.5302
16	2	0.0	-0.3812	-0.1865	-5.8430	-10.9563	-13.9254	-0.0568	-2.4982
16	3	0.0	7.9351	3.8830	-48.6126	-18.3348	-17.5360	1.1818	-3.5871

BEAM FORCES

COL I	COL J	LOADING COMBINATION	TORS MOMENT	I MOMENT	J MOMENT
1	2	1	1.6950	-1.9982	-1.3825
1	2	2	-1.9829	-24.1072	-25.3579
1	2	3	0.5095	2.3545	-72.8044
2	3	1	0.0	1.7204	1.7204
2	3	2	0.0	29.6631	29.6631
2	3	3	0.0	78.1442	-18.8180
3	4	1	1.6950	-1.3825	-1.9982
3	4	2	-1.9829	-25.3579	-24.1072
3	4	3	-4.4752	22.0886	-50.5688
13	14	1	5.0739	-38.3250	-41.5441
13	14	2	1.9829	-24.1072	-25.3579
13	14	3	-0.5095	2.3545	-72.8044
14	15	1	0.0	48.5145	48.5145
14	15	2	0.0	29.6631	29.6631
14	15	3	0.0	78.1442	-18.8180
15	16	1	5.0739	-41.5441	-38.3250
15	16	2	1.9829	-25.3579	-24.1072
15	16	3	4.4752	22.0886	-50.5688
1	5	1	3.7034	-12.9550	-11.6255
1	5	2	3.3998	-0.4295	-0.1308
1	5	3	-0.6472	27.2351	-44.5827
5	9	1	2.9090	-6.0868	-6.0587
5	9	2	0.0000	-0.0159	0.0159
5	9	3	0.0000	41.0015	-41.0015
9	13	1	-1.9249	-11.4130	-12.2379
9	13	2	-3.3998	0.1308	0.4295
9	13	3	0.6472	44.5827	-27.2351
4	8	1	3.7034	12.9550	11.6255
4	8	2	3.3998	0.4295	0.1308
4	8	3	7.4467	28.0941	-44.3212
8	12	1	2.9090	6.0868	6.0587
8	12	2	0.0000	0.0159	-0.0159
8	12	3	0.0000	41.0333	-41.0333
12	16	1	-1.9249	11.4130	12.2379
12	16	2	-3.3998	-0.1308	-0.4295
12	16	3	-7.4467	44.3212	-28.0941
6	7	1	0.0	1.4793	1.4793
6	7	2	0.0	2.1704	2.1704
6	7	3	0.0	3.1909	1.1499
10	11	1	0.0	2.0851	2.0851
10	11	2	0.0	2.1704	2.1704
10	11	3	0.0	3.1909	1.1499
6	10	1	0.0	0.2844	3.2012
6	10	2	0.0	-1.6964	1.6964
6	10	3	0.0	-0.5367	0.5367
7	11	1	0.0	-0.2844	-3.2012
7	11	2	0.0	1.6964	-1.6964
7	11	3	0.0	2.8560	-2.8560

WALL PANEL FORCES

FLEXURAL PANELS

COL I	COL J	LOADING COMBINATION	TOP-MOMENT	BTM-MOMENT	AXIAL-FORCE	SHEAR-FORCE
2	3	1	-178.1087	321.3942	-0.0000	14.3286
2	3	2	-4034.9326	10905.8808	-0.0000	687.0948
2	3	3	-4034.9326	10905.8808	-52.8946	687.0948
14	15	1	-8053.7458	17010.8079	-0.0000	895.7062
14	15	2	-4034.9326	10905.8808	-0.0000	687.0948
14	15	3	-4034.9326	10905.8808	-52.8946	687.0948
6	7	1	79.7329	184.7980	0.0000	26.4531
6	7	2	204.8545	340.6692	0.0000	54.5524
6	7	3	204.8545	340.6692	-0.7781	54.5524
10	11	1	196.6646	326.2901	-0.0000	52.2955
10	11	2	204.8545	340.6692	-0.0000	54.5524
10	11	3	204.8545	340.6692	-0.7781	54.5524
6	10	1	71.9407	59.6605	70.3509	13.1601
6	10	2	0.0000	-0.0000	81.4237	0.0000
6	10	3	0.0000	-0.0000	80.6456	0.0000
7	11	1	-71.9407	-59.6605	-70.3509	-13.1601
7	11	2	-0.0000	0.0000	-81.4237	-0.0000
7	11	3	-0.0000	0.0000	-82.2018	-0.0000

BRACING ELEMENT FORCES

TOP COL	BTM COL	LOADING COMBINATION	AXIAL FORCE
9	5	1	22.3369
9	5	2	0.0311
9	5	3	-2.4144
5	9	1	-22.2824
5	9	2	0.0311
5	9	3	-2.4144
12	8	1	-22.3369
12	8	2	-0.0311
12	8	3	-2.4767
8	12	1	22.2824
8	12	2	-0.0311
8	12	3	-2.4767

.... MEMBER FORCES LEVEL NO. 2 ....

COLUMN FORCES

COLUMN	LOADING COMBINATION	TORSIONAL MOMENT	X-X AXIS TOP MOMENT	X-X AXIS BTM MOMENT	AXIAL FORCE	Y-Y AXIS TOP MOMENT	Y-Y AXIS BTM MOMENT	Y-Y AXIS SHEAR	X-X AXIS SHEAR
1	1	0.0	11.5671	10.4416	1.2421	0.4852	0.6126	2.2009	0.1098
1	2	0.0	-0.9385	-0.7491	2.7923	-16.9417	-13.8056	-0.1688	-3.0747
1	3	0.0	-16.7578	-14.8151	-18.2272	-3.6202	-1.6752	-3.1573	-0.5295
COLUMN 2 IS A DUMMY COLUMN									
COLUMN 3 IS A DUMMY COLUMN									
4	1	0.0	-11.5671	-10.4416	-1.2421	0.4852	0.6126	-2.2009	0.1098
4	2	0.0	0.9385	0.7491	-2.7923	-16.9417	-13.8956	0.1688	-3.0747
4	3	0.0	-14.8808	-13.3168	-23.8119	-30.2633	-25.9360	-2.8198	-5.6199
5	1	0.0	12.8034	12.2713	13.2276	-3.9844	-2.9967	2.5075	-0.6981
5	2	0.0	0.0354	0.0026	0.1789	-8.2619	-5.4732	0.0038	-1.3735
5	3	0.0	3.0616	2.4207	-24.3830	-5.5493	-3.3968	0.5482	-0.8946
COLUMN 6 IS A DUMMY COLUMN									
CCLUMN 7 IS A DUMMY COLUMN									
8	1	0.0	-12.8034	-12.2713	-13.2276	-3.9844	-2.9967	-2.5075	-0.6981
8	2	0.0	-0.0354	-0.0026	-0.1789	-8.2619	-5.4732	-0.0038	-1.3735
8	3	0.0	2.9908	2.4154	-24.7407	-10.9744	-7.5497	0.5406	-1.8524
9	1	0.0	12.7410	12.2667	-12.8620	-11.7132	-9.1554	2.5008	-2.0869
9	2	0.0	-0.0354	-0.0026	0.1789	-8.2619	-5.4732	-0.0038	-1.3735
9	3	0.0	-3.0616	-2.4207	-24.3830	-5.5493	-3.3968	-0.5482	-0.8946
CCLUMN 10 IS A DUMMY COLUMN									
COLUMN 11 IS A DUMMY COLUMN									
12	1	0.0	-12.7410	-12.2667	12.8620	-11.7132	-9.1554	-2.5008	-2.0869
12	2	0.0	0.0354	0.0026	-0.1789	-8.2619	-5.4732	0.0038	-1.3735
12	3	0.0	-2.9908	-2.4154	-24.7407	-10.9744	-7.5497	-0.5406	-1.8524
13	1	0.0	13.2285	11.7454	3.8308	-31.7423	-27.1412	2.4974	-5.8884
13	2	0.0	0.9385	0.7491	2.7923	-16.9417	-13.8056	0.1688	-3.0747
13	3	0.0	16.7578	14.8151	-18.2272	-3.6202	-1.6752	3.1573	-0.5295
COLUMN 14 IS A DUMMY COLUMN									
COLUMN 15 IS A DUMMY COLUMN									
16	1	0.0	-13.2285	-11.7454	-3.8308	-31.7423	-27.1412	-2.4974	-5.8884
16	2	0.0	-0.9385	-0.7491	-2.7923	-16.9417	-13.8056	-0.1688	-3.0747
16	3	0.0	14.8808	13.3168	-23.8119	-30.2633	-25.9360	2.8198	-5.6199

BEAM FORCES

COL I	COL J	LOADING COMBINATION	TORS MOMENT	I MOMENT	J MOMENT
1	2	1	1.0059	-1.7177	-1.3799
1	2	2	-2.2578	-17.9838	-25.9399
1	2	3	1.2607	3.9392	-75.9994
2	3	1	0.0	1.6706	1.6706
2	3	2	0.0	30.6498	30.6498
2	3	3	0.0	82.1315	-20.8318
3	4	1	1.0059	-1.3799	-1.7177
3	4	2	-2.2578	-25.9399	-17.9838
3	4	3	-5.7764	24.1195	-39.9068
13	14	1	5.0443	-31.4592	-45.3745
13	14	2	2.2578	-17.9838	-25.9399
13	14	3	-1.2607	3.9392	-75.9994
14	15	1	0.0	53.5411	53.5411
14	15	2	0.0	30.6498	30.6498
14	15	3	0.0	82.1315	-20.8318
15	16	1	5.0443	-45.3745	-31.4592
15	16	2	2.2578	-25.9399	-17.9838
15	16	3	5.7764	24.1195	-39.9068
1	5	1	2.6371	-8.9951	-8.6250
1	5	2	3.8305	-0.8788	-0.2210
1	5	3	-0.9044	22.2831	-45.3576
5	9	1	0.9628	-2.5965	-2.6618
5	9	2	0.0000	0.0370	-0.0370
5	9	3	0.0000	42.0376	-42.0376
9	13	1	-4.4575	-8.2199	-7.4138
9	13	2	-3.8305	0.2210	0.8788
9	13	3	0.9044	45.3576	-22.2831
4	8	1	2.6371	8.9951	8.6250
4	8	2	3.8305	0.8788	0.2210
4	8	3	8.5654	24.0408	-44.9156
8	12	1	0.9628	2.5965	2.6618
8	12	2	-0.0000	-0.0370	0.0370
8	12	3	-0.0000	41.9636	-41.9636
12	16	1	-4.4575	8.2199	7.4138
12	16	2	-3.8305	-0.2210	-0.8788
12	16	3	-8.5654	44.9156	-24.0408
6	7	1	0.0	1.0642	1.0642
6	7	2	0.0	1.2299	1.2299
6	7	3	0.0	2.8769	-0.4172
10	11	1	0.0	1.3201	1.3201
10	11	2	0.0	1.2299	1.2299
10	11	3	0.0	2.8769	-0.4172
6	10	1	0.0	0.0935	4.2058
6	10	2	0.0	-2.3192	2.3192
6	10	3	0.0	-0.6447	0.6447
7	11	1	0.0	-0.0935	-4.2058
7	11	2	0.0	2.3192	-2.3192
7	11	3	0.0	3.9936	-3.9936

WALL PANEL FORCES

FLEXURAL PANELS

COL I	COL J	LOADING COMBINATION	TOP-MOMENT	BTM-MOMENT	AXIAL-FORCE	SHEAR-FORCE
2	3	1	5.7312	183.2258	-0.0000	18.9957
2	3	2	128.4014	4165.3537	-0.0000	429.3755
2	3	3	128.4014	4165.3537	-26.7918	429.3755
14	15	1	226.2975	8269.0342	0.0000	849.5332
14	15	2	128.4014	4165.3537	0.0000	429.3755
14	15	3	128.4014	4165.3537	-26.7918	429.3755
6	7	1	174.1266	209.7475	0.0000	38.3874
6	7	2	284.8581	332.4219	0.0000	61.7280
6	7	3	284.8581	332.4219	-0.4734	61.7280
10	11	1	354.1101	406.4563	-0.0000	76.0566
10	11	2	284.8581	332.4219	-0.0000	61.7280
10	11	3	284.8581	332.4219	-0.4734	61.7280
6	10	1	101.5573	93.1947	26.3690	19.4752
6	10	2	0.0000	0.0000	28.5155	0.0000
6	10	3	0.0000	0.0000	28.0420	0.0000
7	11	1	-101.5573	-93.1947	-26.3690	-19.4752
7	11	2	-0.0000	-0.0000	-28.5155	-0.0000
7	11	3	-0.0000	-0.0000	-28.9889	-0.0000

BRACING ELEMENT FORCES

TOP COL	BTM COL	LOADING COMBINATION	AXIAL FORCE
9	5	1	31.1709
9	5	2	0.0089
9	5	3	-1.2191
5	9	1	-31.1526
5	9	2	0.0089
5	9	3	-1.2191
12	8	1	-31.1709
12	8	2	-0.0089
12	8	3	-1.2370
8	12	1	31.1526
8	12	2	-0.0089
8	12	3	-1.2370

.... SLAB FORCES LEVEL NO. 2 ....

SLAB FORCE IN X DIRECTION , ALONG ORDINATE Y = 10.00

X-ORDINATE	LOADING COMBINATION	MOMENT-MX	MOMENT-MY	MOMENT-MXY
0.0	1	-0.0506	0.0003	0.0830
0.0	2	-0.1741	0.0297	0.0885
0.0	3	-0.3387	0.1017	0.0203
10.00	1	0.0268	0.0061	0.0152
10.00	2	-0.0312	0.0101	0.0965
10.00	3	0.0290	0.1404	0.1169
20.00	1	0.0000	0.0046	0.0067
20.00	2	-0.1791	0.0810	-0.1167
20.00	3	-0.2025	0.1254	-0.1153
30.00	1	0.0000	-0.0000	0.0101
30.00	2	0.0000	-0.0000	-0.1363
30.00	3	-0.0132	0.0755	-0.1363
40.00	1	-0.0000	-0.0046	0.0067
40.00	2	0.1791	-0.0810	-0.1167
40.00	3	0.1556	-0.0365	-0.1182
50.00	1	-0.0268	-0.0061	0.0152
50.00	2	0.0312	-0.0101	0.0965
50.00	3	0.0914	0.1202	0.0761
60.00	1	0.0506	-0.0003	0.0830
60.00	2	0.1741	-0.0297	0.0885
60.00	3	0.0095	0.0422	0.1567

SLAB FORCE IN X DIRECTION , ALONG ORDINATE Y = 40.00

X-ORDINATE	LOADING COMBINATION	MOMENT-MX	MOMENT-MY	MOMENT-MXY
0.0	1	0.8531	0.1780	-0.1716
0.0	2	0.5982	0.0302	-0.0584
0.0	3	0.8740	-0.0436	-0.0955
10.00	1	-0.0904	0.2198	-0.0556
10.00	2	-0.0315	0.1057	-0.0099
10.00	3	-0.0069	0.0802	-0.0443
20.00	1	-0.0537	0.2735	0.0435
20.00	2	-0.0788	0.1237	0.0292
20.00	3	-0.2357	-0.0232	0.0116
30.00	1	0.0000	0.0000	0.0283
30.00	2	0.0000	0.0000	0.0150
30.00	3	-0.0224	0.0541	0.0150
40.00	1	0.0537	-0.2735	0.0435
40.00	2	0.0788	-0.1237	0.0292
40.00	3	-0.0782	-0.2706	0.0469
50.00	1	0.0904	-0.2198	-0.0556
50.00	2	0.0315	-0.1057	-0.0099
50.00	3	0.0562	-0.1312	0.0244
60.00	1	-0.8531	-0.1780	-0.1716
60.00	2	-0.5982	-0.0302	-0.0594
60.00	3	-0.3224	-0.1041	-0.0212

SLAB FORCE IN X DIRECTION , ALONG ORDINATE Y = 60.00

X-ORDINATE	LOADING COMBINATION	MOMENT-MX	MOMENT-MY	MOMENT-MXY
0.0	1	1.2538	-0.1670	0.0671
0.0	2	0.7518	0.1334	0.0971
0.0	3	1.7760	1.1606	0.6971
10.00	1	-0.1608	0.1560	-0.2178
10.00	2	-0.0868	0.0510	-0.1094
10.00	3	-0.0279	-0.1342	-0.0794
20.00	1	-1.1081	-0.7087	0.4113
20.00	2	-0.6244	-0.3227	0.2312
20.00	3	-0.7343	0.1475	0.2684
30.00	1	0.0000	-0.0000	0.1391
30.00	2	0.0000	-0.0000	0.0997
30.00	3	-0.0097	-0.1288	0.0997
40.00	1	1.1081	0.7087	0.4113
40.00	2	0.6244	0.3227	0.2312
40.00	3	0.5145	0.7929	0.1939
50.00	1	0.1608	-0.1560	-0.2178
50.00	2	0.0868	-0.0510	-0.1094
50.00	3	0.1457	-0.2361	-0.1395
60.00	1	-1.2538	0.1670	0.0671
60.00	2	-0.7518	-0.1334	0.0971
60.00	3	0.2723	0.8939	-0.5029

SLAB FORCE IN Y DIRECTION , ALONG ORDINATE X = 10.00

Y-ORDINATE	LOADING COMBINATION	MOMENT-MX	MOMENT-MY	MOMENT-MXY
0.0	1	0.0095	-0.0643	-0.0209
0.0	2	-0.0868	0.0510	0.1094
0.0	3	-0.0279	-0.1342	0.0794
10.00	1	0.0268	0.0061	0.0152
10.00	2	-0.0312	0.0101	0.0965
10.00	3	0.0290	0.1404	0.1169
20.00	1	0.0383	-0.0239	-0.0378
20.00	2	-0.0315	0.1057	0.0099
20.00	3	-0.0069	0.0802	0.0443
30.00	1	0.0967	0.0030	-0.0282
30.00	2	0.0982	0.0042	-0.0000
30.00	3	0.1224	-0.0397	-0.0000
40.00	1	-0.0904	0.2198	-0.0556
40.00	2	-0.0315	0.1057	-0.0099
40.00	3	-0.0069	0.0802	-0.0443
50.00	1	-0.0680	0.0097	-0.1556
50.00	2	-0.0312	0.0101	-0.0965
50.00	3	0.0290	0.1404	-0.1169
60.00	1	-0.1608	0.1560	-0.2178
60.00	2	-0.0868	0.0510	-0.1094
60.00	3	-0.0279	-0.1342	-0.0794

SLAB FORCE IN Y DIRECTION , ALONG ORDINATE X = 30.00

Y-ORDINATE	LOADING COMBINATION	MOMENT-MX	MOMENT-MY	MOMENT-MXY
0.0	1	-0.0000	-0.0000	-0.0389
0.0	2	-0.0000	-0.0000	-0.0997
0.0	3	-0.0097	-0.1288	-0.0997
10.00	1	0.0000	-0.0000	0.0101
10.00	2	0.0000	-0.0000	-0.1363
10.00	3	-0.0132	0.0755	-0.1363
20.00	1	-0.0000	0.0000	0.0012
20.00	2	-0.0000	0.0000	-0.0150
20.00	3	-0.0224	0.0541	-0.0150
40.00	1	0.0000	0.0000	0.0283
40.00	2	0.0000	0.0000	0.0150
40.00	3	-0.0224	0.0541	0.0150
50.00	1	0.0000	-0.0000	0.2529
50.00	2	-0.0000	0.0000	0.1363
50.00	3	-0.0132	0.0755	0.1363
60.00	1	0.0000	-0.0000	0.1391
60.00	2	0.0000	-0.0000	0.0997
60.00	3	-0.0097	-0.1288	0.0997

SLAB FORCE IN Y DIRECTION , ALONG ORDINATE X = 50.00

Y-ORDINATE	LOADING COMBINATION	MOMENT-MX	MOMENT-MY	MOMENT-MXY
0.0	1	-0.0095	0.0643	-0.0209
0.0	2	0.0868	-0.0510	0.1094
0.0	3	0.1457	-0.2361	0.1395
10.00	1	-0.0268	-0.0061	0.0152
10.00	2	0.0312	-0.0101	0.0965
10.00	3	0.0914	0.1202	0.0761
20.00	1	-0.0383	0.0239	-0.0378
20.00	2	0.0315	-0.1057	0.0099
20.00	3	0.0562	-0.1312	-0.0244
30.00	1	-0.0967	-0.0030	-0.0282
30.00	2	-0.0982	-0.0042	-0.0000
30.00	3	-0.0741	-0.0480	-0.0000
40.00	1	0.0904	-0.2198	-0.0556
40.00	2	0.0315	-0.1057	-0.0099
40.00	3	0.0562	-0.1312	0.0244
50.00	1	0.0680	-0.0097	-0.1556
50.00	2	0.0312	-0.0101	-0.0965
50.00	3	0.0914	0.1202	-0.0761
60.00	1	0.1608	-0.1560	-0.2178
60.00	2	0.0868	-0.0510	-0.1094
60.00	3	0.1457	-0.2361	-0.1395