UC SESM 71 - 19

STRUCTURES AND MATERIALS RESEARCH
DEPARTMENT OF CIVIL ENGINEERING

SOLID SAP

A STATIC ANALYSIS PROGRAM FOR THREE DIMENSIONAL SOLID STRUCTURES

by

EDWARD L. WILSON

SEPTEMBER 1971
REVISED
DECEMBER 1972

STRUCTURAL ENGINEERING LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY CALIFORNIA

SOLID SAP

A STATIC ANALYSIS PROGRAM
FOR THREE DIMENSIONAL SOLID STRUCTURES

by

Edward L. Wilson

Report to

Denver Mining Research Center
U.S. Department of the Interior
Bureau of Mines

September 1971

REVISED REPORT

This report is a revised edition of "Solid Sap", Report UC SESM 71-19 of September 1971. The revision includes clarification of instructions for use of the program and correction of typographical errors. Minor changes in the program code are reflected in the new listing of Appendix D.

ABSTRACT

A general computer program is given for the linear elastic static analysis of complex structural systems. New finite elements are developed which are useful in the analysis of two and three dimensional solids. These elements have special advantages for the analysis of massive underground structures such as mine structures and above ground structures such as arch dams. Examples are given which illustrate the accuracy of the elements. A description on the use of the program and a FORTRAN IV listing are given in the Appendices.

ACKNOWLEDGEMENT

In addition to the author, the following people participated in the computer program development: Lindsay R. Jones programmed parts of the main program and the beam element subroutines. Peter G. Smith participated in the initial organization of the program and in the development of the two-dimensional plane element. Te-ming Hsueh incorporated Dr. Carlos A. Felippa's quadrilateral shell element into the program. H. H. Dovey modified the three-dimensional solid element initially programmed by Kenneth T. Kavanagh. William P. Doherty programmed parts of the equation solver, beam element, plane stress element and axisymmetric solid element. J. Ghaboussi developed the 16 node thick shell element.

During the past several years many organizations have sponsored various phases of the development of this computer program. The most recent additions to the program which are described in this report have been sponsored by the U. S. Department of the Interior, Bureau of Mines, under Contract No. H0110231 with the University of California, Berkeley. Mr. Fun-Den Wang of the Denver Mining Research Center served as the Contracting Officer's representative. Work at the University of California was conducted under the direction of Professors R. W. Clough and E. L. Wilson.

TABLE OF CONTENTS

I. INTRODUCTION

The purpose of this computer program is to perform static, linear, elastic analyses of three dimensional structural systems. The structural systems to be analysed may be composed of combinations of a number of structural element types. The present version contains the following element types:

1) three dimensional truss
2) three dimensional beam
3) plane stress and plane strain
4) two dimensional axisymmetric solid
5) three dimensional solid
6) plate and shell
7) boundary
8) thick shell elements

Since several of these elements have not been published it will be necessary to present their development in this report. Only an outline and the unique characteristics of the program will be given; no attempt is made to present internal documentation of the program .

Systems composed of large numbers of joints and members may be analysed. The capacity of the program depends mainly on the total number of joints in the system. There is practically no restriction on the number of elements, number of load cases, or the "bandwidth" of the equations to be solved. Note, that while the program has the capacity to analyse very large systems, there is no loss of efficiency in the solution of smaller problems as compared to several special purpose programs presently available. The program is machine independent and is coded in standard FORTRAN IV.

The program presented in this report is an extensive modification of a previous computer program [8]. New elements have been introduced which have special advantages for the analysis of massive underground structures such as mine structures and above ground structures such as arch dams. Also, new coding techniques have been utilized which improve the speed of the program and reduce low speed storage requirements. In addition, the dynamic options have been removed in order that the program can be used with a minimum of effort for static analyses. Finally, this report contains a more complete discussion of the use of incompatible displacement modes within solid elements and of the static condensation algorithm.

II. EQUILIBRIUM EQUATIONS FOR COMPLEX STRUCTURAL SYSTEMS

2.1 The Direct Stiffness Method

The governing joint equilibrium equations for a structural system can be derived by several different approaches. All methods yield a set of linear equations of the following form:

$$\underline{K}\ \underline{u} = \underline{R} \tag{2.1}$$

These equations set the sum of the internal element forces, $\underline{K}\ \underline{u}$, expressed in terms of joint displacements, $\underline{u}$, to the generalized loads, $\underline{R}$, acting at the joints. The matrix $\underline{u}$ contains all the joint displacements (degrees of freedom) of the system. The stiffness matrix $\underline{K}$ can be formed by the direct addition of element stiffness matrices; or

$$\underline{K} = \Sigma\ \underline{K}_m \tag{2.2}$$

For a typical element $\underline{m}$ the element stiffness matrix is given by

$$\underline{K}_m = \int_{Vol} \underline{a}_m^T\ \underline{c}_m\ \underline{a}_m\ d\,V_m \tag{2.3}$$

The stress-strain relationship for the element is of the form

$$\underline{\sigma}_m = \underline{c}_m\ \underline{\varepsilon}_m + \underline{\tau}_m \tag{2.4}$$

where $\underline{\varepsilon}_m$ are the element strains produced by the displacements $\underline{u}$ and $\underline{\tau}_m$ are the initial stresses in the element before deformation.

Within each element the strains are expressed (approximately) by the following equation:

$$\underline{\varepsilon}_m = \underline{a}_m \, \underline{u} \tag{2.5}$$

Note that $\underline{a}_m$ appear to be a very large matrix since $\underline{u}$ contains all degrees of freedom of the system. However, within the computer program only the non-zero columns of $\underline{a}_m$ are stored and their column numbers are stored as a separate identification array. The advantage of this notation is that the "direct" addition of element stiffness matrices as implied by equation (2.2) is correct.

The generalized loads, $\underline{R}$, are given by

$$\underline{R} = \underline{P} + \underline{T} - \underline{F} \tag{2.6}$$

where $\underline{P}$ is a matrix of concentrated joint loads and $\underline{T}$ is a matrix of generalized loads due to distributed surface stresses and is given by a summation of boundary element forces, or

$$\underline{T} = \Sigma \, \underline{T}_m \tag{2.7}$$

in which

$$\underline{T}_m = \int_{Area} b_m^T \, \underline{t}_m \, ds_m \tag{2.8}$$

The surface stresses are $\underline{t}_m$ and the relationship between surface displacements $\underline{u}_m$ and joint displacements $\underline{u}$ is

$$\underline{u}_m \, (s) = \underline{b}_m \, \underline{u} \tag{2.9}$$

$\underline{F}$ is a matrix of generalized loads due to the initial stresses $\underline{\tau}_m$ and is given by a summation of element forces, or

$$\underline{F} = \Sigma \; \underline{F}_m \tag{2.10}$$

in which

$$\underline{F}_m = \int_{Vol} \underline{d}_m^T \; \underline{\tau}_m \; dV_m \tag{2.11}$$

The matrix $\underline{d}_m$ is the basic displacement field approximation within the element:

$$\underline{u}_m \; (x, y, z) = \underline{d}_m \; \underline{u} \tag{2.12}$$

2.2 <u>Boundary Conditions</u>

Equation (2.1) represents the relationship between all joint forces and all joint displacements and can be rewritten in partitioned form as:

$$\underline{K}_{aa} \; \underline{u}_a + \underline{K}_{ab} \; \underline{u}_b = \underline{R}_a \tag{2.13}$$

$$\underline{K}_{ba} \; \underline{u}_a + \underline{K}_{bb} \; \underline{u}_b = \underline{R}_b \tag{2.14}$$

where $\underline{R}_a$ = the specified joint loads

$\underline{R}_b$ = the unknown joint reactions

$\underline{u}_a$ = the unknown joint displacements

$\underline{u}_b$ = the specified joint displacements

The normal approach to the solution to this problem is to rewrite Eq. (2.13) in the following form:

$$\underline{K}_{aa}\ \underline{u}_a = \underline{R}_a - \underline{K}_{ab}\ \underline{u}_b = \underline{R}_a^* \qquad (2.15)$$

Since $\underline{R}_a^*$ can be calculated directly, Eq. (2.15) can be solved for the unknown displacement.

In this report another approach is used which has certain programming advantages. If a displacement component, u_b, is zero, the stiffness coefficients $\underline{K}_{ab}$, $\underline{K}_{ba}$ and K_{bb} are not added to the total stiffness matrix and that particular degree of freedom is disregarded in the equilibrium equations. If, however, a non-zero displacement is to be specified, $u_b = x$, Eq. (2.14) is modified by the addition of an equation of the following form:

$$k\ u_b = kx \qquad (2.16)$$

where k is an arbitrary number. The resulting equation is:

$$\underline{K}_{ba}\ \underline{u}_a + (K_{bb} + k)\ u_b = R_b + kx \qquad (2.17)$$

If k is selected to be several orders of magnitude greater than the stiffness coefficient K_{bb} the solution of this equation will be $u_b = x$. This may also be interpreted physically as adding a spring of large stiffness k to the structure; a large load kx is then applied; therefore, the relatively flexible structure will move along with the spring in order to produce the displacement x.

This technique of adding a stiff spring to the structure may also be used to specify skew boundary conditions. It may also be used to obtain support reactions by specifying zero displacement at the support node.

III. THREE DIMENSIONAL TRUSS ELEMENT

The three-dimensional truss element will be explained in detail in order to illustrate the calculation of the stiffness matrix for a typical element.

A typical truss element connected to joints i and j is shown below. All dimensions are assumed positive; however, the development is correct for elements of different orientation.

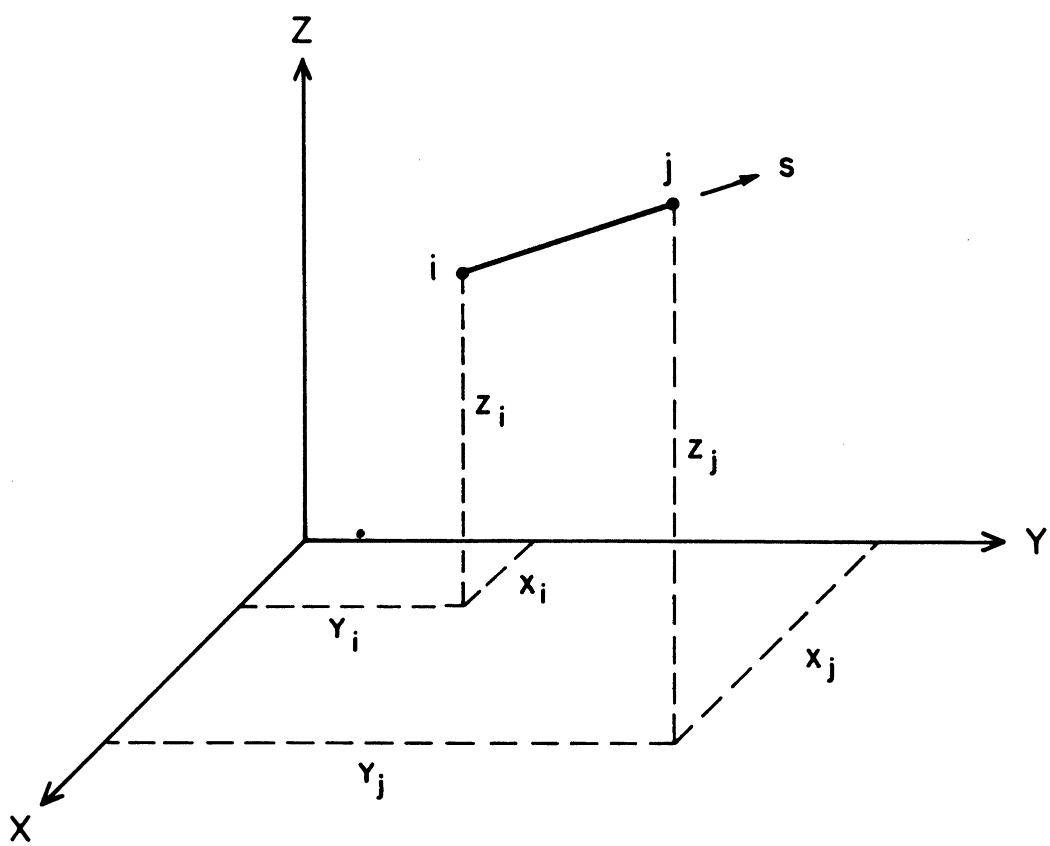


FIGURE 3-1 TYPICAL TRUSS ELEMENT

The length of the element is given by

$$L = \sqrt{L_x^2 + L_y^2 + L_z^2} \tag{3.1}$$

where

$$\begin{aligned} L_x &= x_j - x_i \\ L_y &= y_j - y_i \\ L_z &= z_j - z_i \end{aligned} \tag{3.2}$$

The axial displacement in the s-direction is assumed to be linear (constant strain).

$$u_s = u_{si} + \frac{s}{L}(u_{sj} - u_{si}) \tag{3.3}$$

where s equals zero at joint i. Therefore, the axial strain is

$$\varepsilon_s = \frac{\partial u_s}{\partial s} = \frac{1}{L}(u_{sj} - u_{si}) \tag{3.4}$$

The axial displacement u_s is given in terms of the global displacements u_x, u_y, and u_z by

$$u_s = \frac{L_x}{L} u_x + \frac{L_y}{L} u_y + \frac{L_z}{L} u_z \tag{3.5}$$

The evaluation of equation (3.5) at joints i and j and the substitution into equation (3.4) yields the following expression for element strain:

$$\varepsilon_s = \frac{1}{L^2} [- L_x - L_y - L_z \; L_x \; L_y \; L_z] \begin{bmatrix} u_{xi} \\ u_{yi} \\ u_{zi} \\ u_{xj} \\ u_{yj} \\ u_{zj} \end{bmatrix} \tag{3.6}$$

Therefore, the strain-displacement matrix for the truss element is a 1 x 6 matrix and is given by

$$\underline{a} = \frac{1}{L^2} [- L_x - L_y - L_z \; L_x \; L_y \; L_z] \tag{3.7}$$

The axial stress is expressed in terms of axial strain by

$$\sigma_s = E \, \varepsilon_s \tag{3.8}$$

Therefore, the stress-strain relationship is a 1 x 1 matrix, or

$$\underline{c} = [E] \tag{3.9}$$

where E is the modulus of elasticity of the truss material.

From equation (2.3) the element stiffness can be calculated directly. Since the volume of the element is equal to the cross-sectional area of the element, A, times the length of the element, L, the element stiffness is of the form:

$$K = \frac{AE}{L^3} \begin{bmatrix} -L_x \\ -L_y \\ -L_z \\ L_x \\ L_y \\ L_z \end{bmatrix} [\, -L_x \; -L_y \; -L_z \; L_x \; L_y \; L_z \,] \tag{3.10}$$

From equations (3.8) and (3.6) the axial stress in terms of global displacements is

$$\sigma_s = \frac{E}{L^2} [\, -L_x \; -L_y \; -L_z \; L_x \; L_y \; L_z \,] \begin{bmatrix} u_{xi} \\ u_{yi} \\ u_{zi} \\ u_{xj} \\ u_{yj} \\ u_{zj} \end{bmatrix} \tag{3.11}$$

Within the computer program the stress-displacement relationship is always calculated at the same time as the element stiffness is evaluated; it is then placed on tape storage and is used later in the determination of element stresses after the joint displacements are determined.

IV. THREE DIMENSIONAL BEAM ELEMENT

The beam element included in this program considers torsion, bending about two axes, axial and shearing deformations. The element is prismatic and the development of its stiffness properties is standard and is given in many modern texts on structural analysis. Only the unique characteristics will be discussed in this section.

4.1 Definition of Principal Axes

The geometric location of a typical element is defined by joint numbers i and j. The place which locates the principal axis of the beam is defined by a third joint number k as shown in Figure 4.1. The relationship between the local coordinate system, s_1, s_2 and s_3 is most conveniently developed by the use of vector notation. The unit vectors in the $\hat{s}_1$ and $\hat{g}$ directions are given by

$$\hat{s}_1 = S_{1x}\,\hat{x} + S_{1y}\,\hat{y} + S_{1z}\,\hat{z}$$

$$\hat{g} = g_x\,\hat{x} + g_y\,\hat{y} + g_z\,\hat{z}$$

where $\hat{x}$, $\hat{y}$, and $\hat{z}$ are unit vectors in the x, y and z directions, respectively. The direction cosines are

$$S_{1x} = \frac{L_x}{L}\ ;\ S_{1y} = \frac{L_y}{L}\ ;\ S_{1z} = \frac{L_z}{L}$$

$$q_x = \frac{G_x}{G}\ ;\ g_y = \frac{G_y}{G}\ ;\ g_z = \frac{G_z}{G}$$

in which

$$L_x = x_j - x_i \; ; \; L_y = y_j - y_i \; ; \; L_z = z_j - z_i$$

$$G_x = x_k - x_i \; ; \; G_y = y_k - y_i \; ; \; G_z = z_k - z_i$$

$$L = \sqrt{L_x^2 + L_y^2 + L_z^2} \quad \text{and} \quad G = \sqrt{G_x^2 + G_y^2 + G_z^2}$$

The unit vector in the $\hat{s}_3$ direction is given by the vector product of $\hat{s}_1$ and $\hat{g}$ divided by the length of the vector in that direction

$$\hat{s}_3 = \frac{\hat{s}_1 \times \hat{g}}{\left|\hat{s}_1 \times \hat{g}\right|} = S_{3x}\,\hat{x} + S_{3y}\,\hat{y} + S_{3z}\,\hat{z}$$

where the vector product is by definition the evaluation of the determinant

$$\underline{a} \times \underline{b} = \begin{bmatrix} \hat{x} & \hat{y} & \hat{z} \\ a_x & a_y & a_z \\ b_x & b_y & b_z \end{bmatrix}$$

The unit vector in the $\hat{s}_2$ direction is given by the vector product

$$\hat{s}_2 = \hat{s}_3 \times \hat{s}_1 = S_{2x}\,\hat{x} + S_{2y}\,\hat{y} + S_{2z}\,\hat{z}$$

The three unit vectors may be summarized by the following matrix equations:

$$\begin{bmatrix} \hat{s}_1 \\ \hat{s}_2 \\ \hat{s}_3 \end{bmatrix} \begin{bmatrix} S_{1x} & S_{1y} & S_{1z} \\ S_{2x} & S_{2y} & S_{2z} \\ S_{3x} & S_{3y} & S_{3z} \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix}$$

Within the computer program local displacements, forces and moments are transformed to the global system by this 3 x 3 matrix.

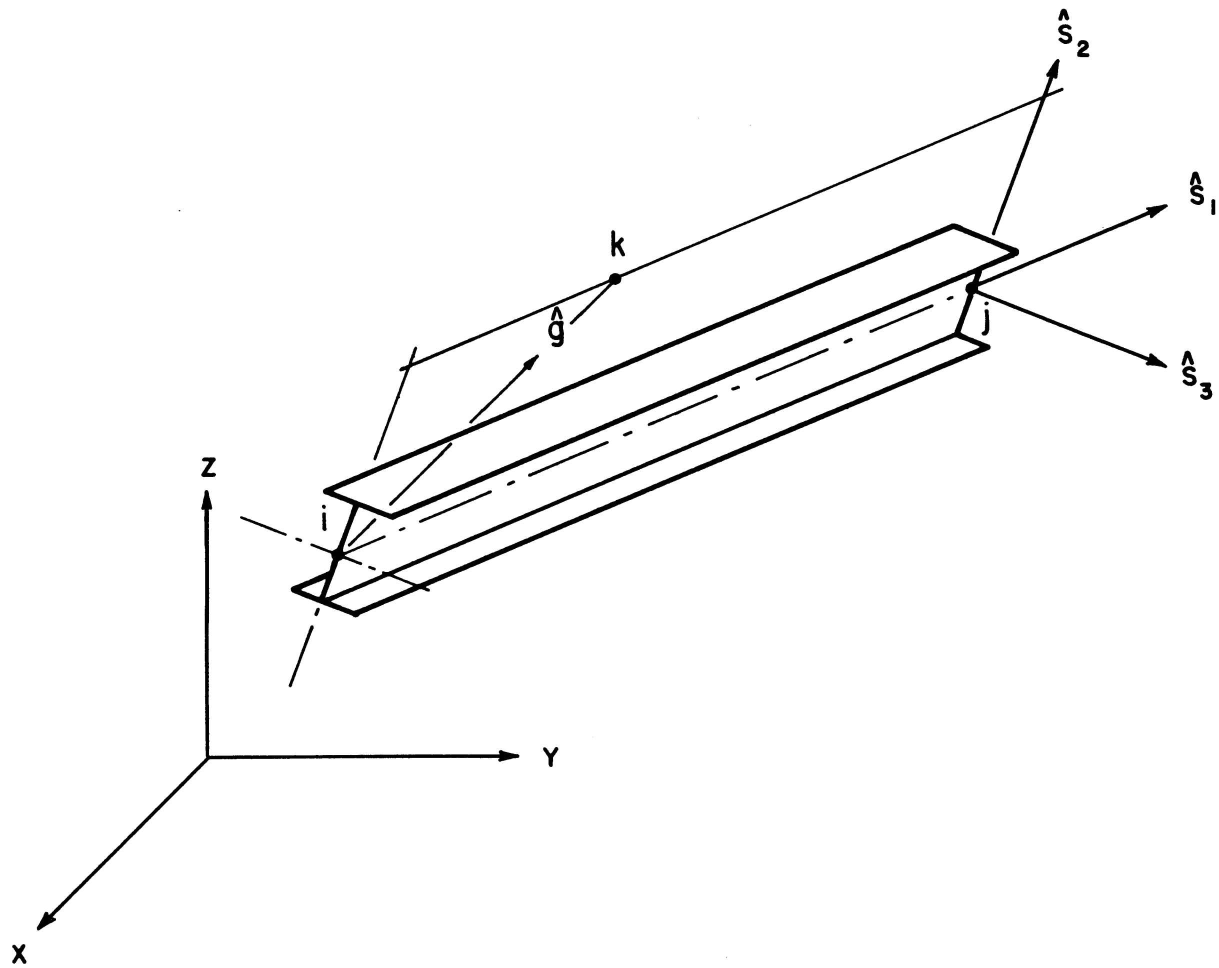


FIGURE 4-1 THREE DIMENSIONAL BEAM ELEMENT

4.2 Master and Slave Degrees of Freedom

The joints of three dimensional beam elements can be connected to slave degrees of freedom. Slave degrees of freedom are eliminated from the formulation and replaced by the degrees of freedom of the master joint. This technique reduces the total number of joint equilibrium equations, in the system and greatly reduces the possibility of numerical sensitivities in many types of structures.

The geometry of the master and slave joints is shown in Figure 4.2. A beam node may be connected directly to either a master or a slave joint. Any one of the six degrees of freedom of the slave joint may be eliminated. If all six degrees of freedom at a given beam node are made into slaves then the physical effect is that the master and beam joint are connected with a rigid link.

If the x displacement of the slave joint is defined as a slave of the master joint the displacements will be transformed as follows:

$$u_{xs} = u_{xm} + (z_s - z_m)\ \theta_{ym} - (y_s - y_m)\ \theta_{zm}$$

For the y-displacement

$$u_{ys} = u_{ym} - (z_s - z_m)\ \theta_{xm} + (x_s - x_m)\ \theta_{zm}$$

For the z-displacement

$$u_{zs} = u_{zm} + (y_s - y_m)\ \theta_{xm} - (x_s - x_m)\ \theta_{ym}$$

and for the rotations

$$\theta_{xs} = \theta_{xm}$$

$$\theta_{ys} = \theta_{sm}$$

$$\theta_{zs} = \theta_{zm}$$

For the beam elements those transformations automatically take place for all elements connected to slave degrees of freedom. The computer program allows a joint to be a slave to more than one master; however, this is an incorrect use of the option.

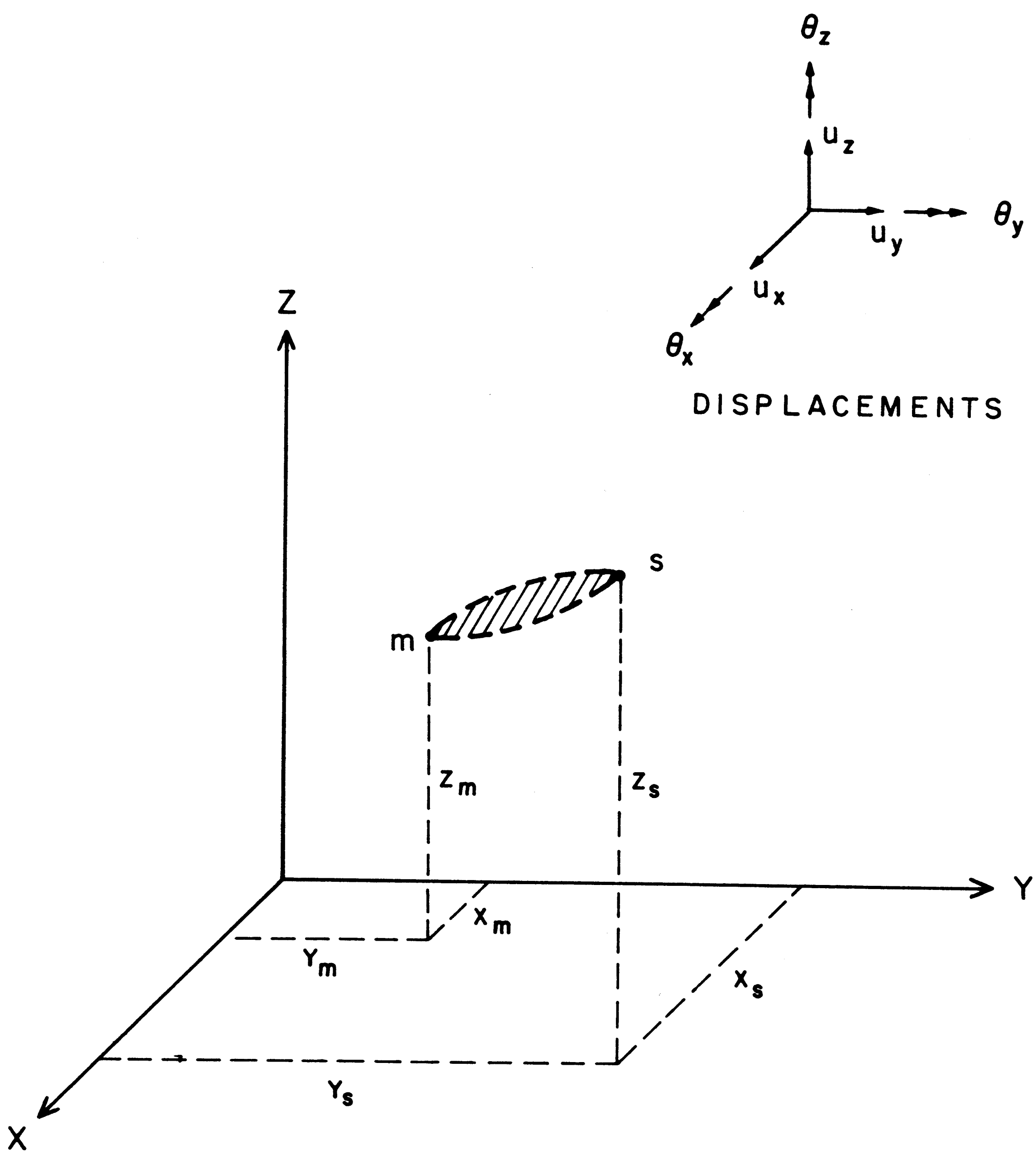


FIGURE 4-2 MASTER AND SLAVE JOINTS

V. SOLID FINITE ELEMENTS

The following type of solid finite elements are included in SOLID SAP:

1. Two-dimensional membrane plane stress elements of specified thickness and located in an arbitrary plane.
2. Two-dimensional plane stress, plane strain and axisymmetric elements located in the Y-Z plane.
3. Eight node three-dimensional solid elements.
4. Sixteen node three-dimensional thick shell elements.

All of these elements are based on an isoparametric formulation with the addition of incompatible displacement modes. Since this type of element has not been explained previously it is necessary to present the basic method in this report.

5.1 Introduction

One of the most significant developments in the numerical solution of solid structures was the introduction of isoparametric finite elements [1]. As a result, many elements with a high degree of accuracy have been developed [2]. In addition, the technique has been extended to curved shell elements [3]. The purpose of this paper is to present a modification of this approach which results in a further improvement in accuracy.

Other attempts have been made to improve the basic accuracy of these elements. In general they involve the use of approximate integration techniques which disregard part of the shear strain energy associated with pure bending modes [4], [5], [6]. However, these methods have been limited to idealized geometries and isotropic materials. Also, convergence may not be assured.

The method presented in this paper formally introduces incompatible displacement modes at the element level in order to improve the element accuracy. These unknowns are eliminated by requiring that the total strain energy within the element is minimum. Convergence of the solution is assured. Examples are presented for two and three dimensional solids and for thick shells.

5.2 Source of Errors

One of the main causes of inaccuracies in lower order finite elements is due to their inability to represent certain simple stress gradients. This is clearly illustrated by subjecting a simple rectangular element to a pure bending stress as shown in figure (1a). The exact displacements for this type of loading are illustrated in figure (1b) and are given by the following equations:

$$u_x = \alpha_1 \, xy \tag{5.1}$$

$$u_y = 1/2 \, \alpha_1 \, (a^2 - x^2) + \alpha_2 \, (b^2 - y^2) \tag{5.2}$$

It is clear that these displacements satisfy the pure bending condition of zero shear strain; or

$$\varepsilon_{xy} = \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} = 0 \tag{5.3}$$

The constant α_2 is a function of the material properties - for Poisson's ratio equal to zero $\alpha_2 = 0$.

For a finite element displacement model the only displacement activated for this type of loading is shown in figure (1c) and is given by

$$u_x = \beta_1 \, xy \tag{5.4}$$

Therefore, the form of the error in the solution is

$$u_y = \beta_2 \, (a^2 - x^2) + \beta_3 \, (b^2 - y^2) \tag{5.5}$$

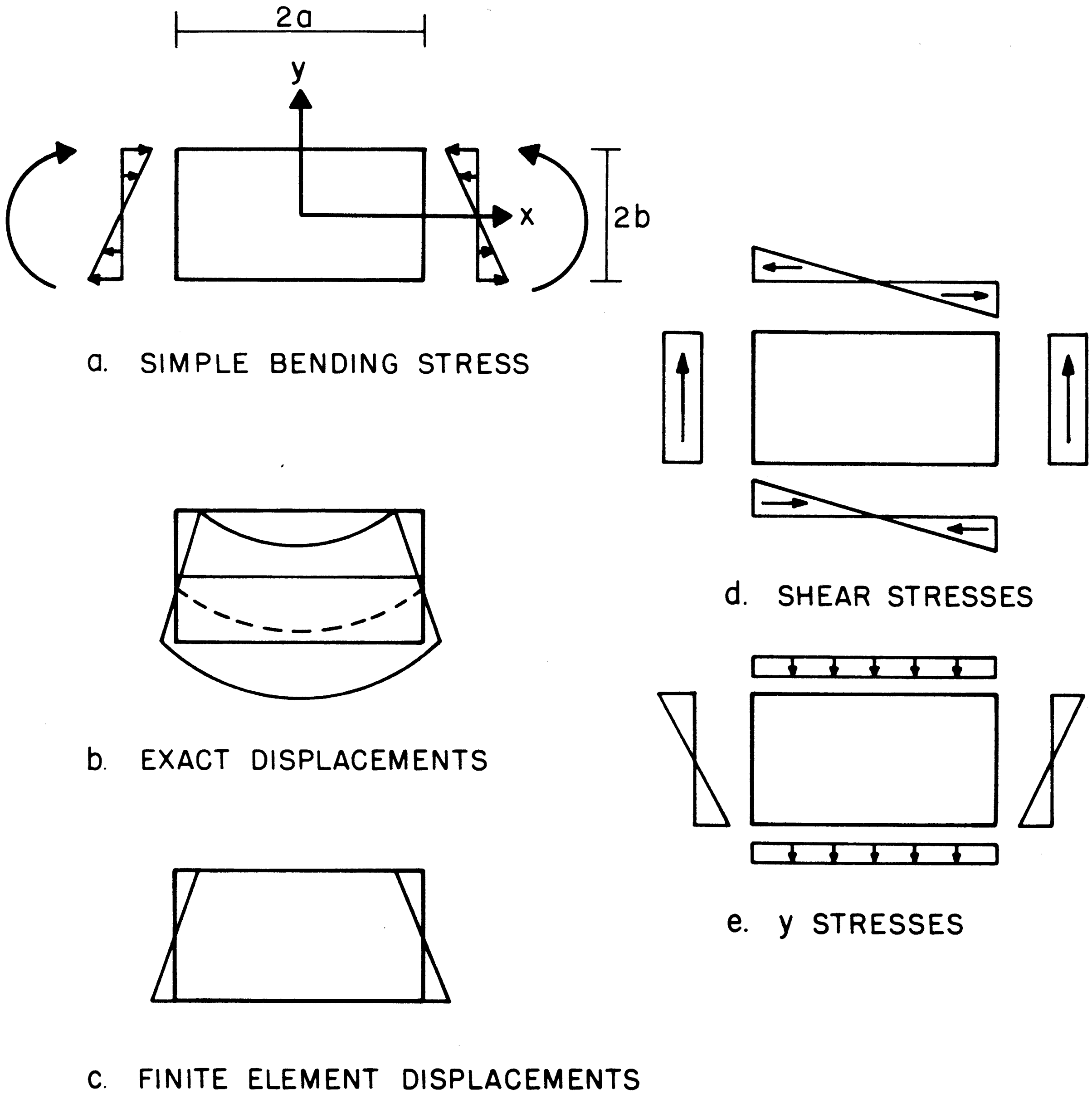


FIGURE 5-1 ERRORS DUE TO PURE BENDING STRESSES

The errors in stresses associated with equation (5) are shown in figures (1d) and (1e).

Previous attempts to reduce these errors have involved selecting integration formulas which disregard the strain energy associated with the stresses shown in figure (1d). One point integration applied at the center of the element will accomplish this. However, this technique can produce a stiffness matrix which has directional properties. For shell type structures the normal stresses shown in figure (1d) have been disregarded by making the assumption of plane stress in the stress-strain equations. It is clear that these approximate methods are difficult to apply in the general case of curved anisotropic elements.

In this paper the approach adopted to minimize these errors is to add extra displacement modes to the elements which have the same form as the errors in the simple displacement approximation. In general these extra displacement modes violate inter-element compatibility. The magnitudes of the modes are selected by requiring that the total strain energy of the element be a minimum. In the following sections of this report, the method will be presented as an addition to the basic isoparametric method. Specific examples to various types of elements will be given.

5.3 Addition of Incompatible Modes for Two-Dimensional Isoparametric Elements

A two dimensional isoparametric element will be used to illustrate the method in detail. For a general quadrilateral element, as shown in figure (2), the local and global coordinate systems are related by

$$x = \sum_{i=1}^{4} h_i \, x_i \tag{5.6a}$$

$$y = \sum_{i=1}^{4} h_i \, y_i \tag{5.6b}$$

where the interpolation functions are given by

$$h_1 = 1/4 \, (1-s) \, (1-t) \tag{5.7a}$$

$$h_2 = 1/4 \, (1+s) \, (1-t) \tag{5.7b}$$

$$h_3 = 1/4 \, (1+s) \, (1+t) \tag{5.7c}$$

$$h_4 = 1/4 \, (1-s) \, (1+t) \tag{5.7d}$$

Strain-Displacement Equations

In order to ensure rigid body displacement modes the same interpolation functions are used in the displacement approximation.

$$u_x \, (s,t) = \Sigma \, h_i \, u_{xi} \tag{5.8a}$$

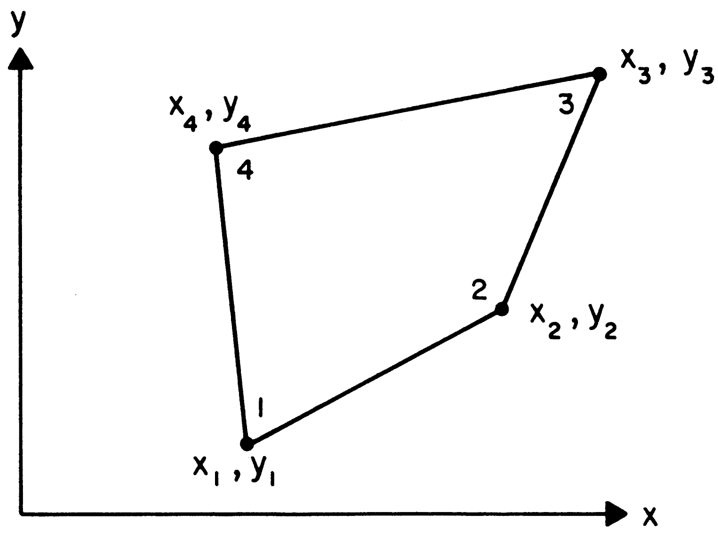
$$u_y \, (s,t) = \Sigma \, h_i \, u_{yi} \tag{5.8b}$$

For two dimensional analysis the strain-displacement equations are

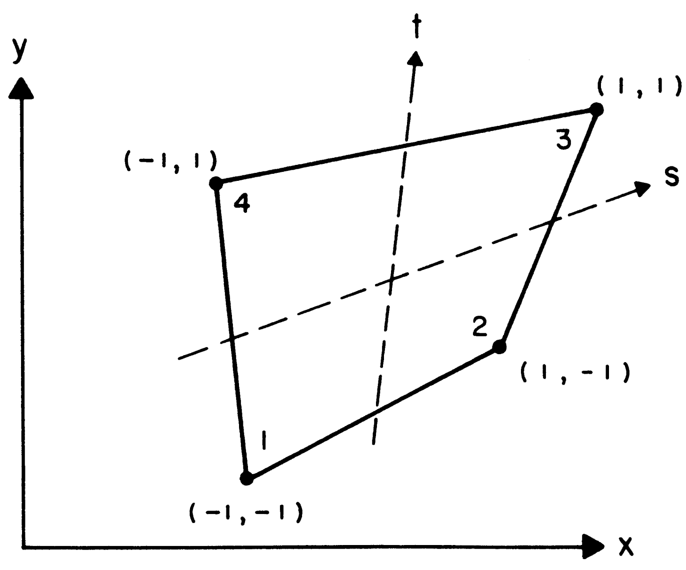
$$\varepsilon_{xx} = \frac{\partial u_x}{\partial x} = \Sigma \, h_{i,x} \, u_{xi} \tag{5.9a}$$

$$\varepsilon_{yy} = \frac{\partial u_y}{\partial y} = \Sigma \, h_{i,y} \, u_{yi} \tag{5.9b}$$

$$\varepsilon_{xy} = \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} = \Sigma \, h_{i,y} \, u_{xi} + \Sigma \, h_{i,x} \, u_{yi} \tag{5.9c}$$



a. GLOBAL SYSTEM



b. LOCAL SYSTEM

FIGURE 5-2 TWO-DIMENSIONAL ISOPARMETRIC ELEMENT

Or equation (9) can be written in matrix form as

$$\underline{\varepsilon} = \underline{a}\ (s,t)\ \underline{U} = \begin{bmatrix} \underline{H},x & 0 \\ 0 & \underline{H},y \\ \underline{H},y & \underline{H},x \end{bmatrix} \begin{bmatrix} \underline{U}x \\ \underline{U}y \end{bmatrix} \tag{5.10}$$

In this case the three strains are related to the eight nodal point displacements by a 3 x 8 matrix. The submatrices in equation (10) are given by

$$\underline{H},_x = [h_{1,x}\ h_{2,x}\ h_{3,x}\ h_{4,x}] \tag{5.11a}$$

$$\underline{H},_y = [h_{1,y}\ h_{2,y}\ h_{3,y}\ h_{4,y}] \tag{5.11b}$$

Since the functions h_i are in terms of s and t the chain rule is applied in order to compute the derivatives with respect to the global x - y system.

$$h_{i,x} = h_{i,s}\ s,_x + h_{i,t}\ t,_x \tag{5.12a}$$

$$h_{i,y} = h_{i,s}\ s,_y + h_{i,t}\ t,_y \tag{5.12b}$$

In general, the chain rule can be written as

$$\begin{bmatrix} \frac{\partial}{\partial s} \\ \frac{\partial}{\partial t} \end{bmatrix} = \begin{bmatrix} x,_s & y,_s \\ x,_t & y,_t \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \tag{5.13}$$

or inverted as

$$\begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} = \begin{bmatrix} s,_x & t,_x \\ s,_y & t,_y \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial s} \\ \frac{\partial}{\partial t} \end{bmatrix} \tag{5.14}$$

Therefore, the derivative required in equation (12) is given by

$$\begin{bmatrix} s,_x & t,_x \\ s,_y & t,_y \end{bmatrix} = \frac{1}{J} \begin{bmatrix} y,_t & -y,_s \\ -x,_t & x,_s \end{bmatrix} \tag{5.15}$$

where the Jacobian J is

$$J = x,_s\, y,_t - x,_t\, y,_s \tag{5.16}$$

From equations (6a) and(6b)

$$\begin{aligned} x,_s &= \Sigma\, h_{i,s}\, x_i \\ x,_t &= \Sigma\, h_{i,t}\, x_i \\ y,_s &= \Sigma\, h_{i,s}\, y_i \\ y,_t &= \Sigma\, h_{i,t}\, y_i \end{aligned} \tag{5.17}$$

For given numerical values of s and t the derivatives of the interpolating functions can be evaluated. Then from equations (17) and (15) all required derivatives for the numerical evaluation of the strain displacement matrix, equation (10) can be calculated.

Element Stiffness and Numerical Integration

For unit thickness the element stiffness matrix is given by

$$\underline{K} = \int_{Area} \underline{a}^T\, \underline{c}\, \underline{a}\; dA \tag{5.18}$$

in which $\underline{c}$ is the stress-strain matrix and the integration is carried out over the area of the element.

For the purpose of numerical integration, equation (18) is written in the s and t system as

$$\underline{K} = \int_{-1}^{1} \int_{-1}^{1} \underline{a}^T\, \underline{c}\, \underline{a}\; J\; ds\; dt \tag{5.19}$$

The direct application of one-dimensional numerical integration formulas [7] yields

$$\underline{K} = \sum_j \sum_k W_j \, W_k \, J \, \underline{a}^T(s_j, t_k) \, \underline{c} \, \underline{a} \, (s_j, t_k) \tag{5.20}$$

in which s_j and t_k are integration points and W_j and W_k are the appropriate weight functions.

Addition of Incompatible Modes

The basic method is the same when internal degrees of freedom are added at the element level. For the quadrilateral element the displacement approximation may be of the following form:

$$u_x = \Sigma \, h_i \, u_{xi} + h_5 \, \alpha_1 + h_6 \, \alpha_2$$
$$u_y = \Sigma \, h_i \, u_{yi} + h_5 \, \alpha_3 + h_6 \, \alpha_4$$

The functions h_5 and h_6 must be zero at the four nodes. The displacement amplitudes α_i are additional degrees of freedom; therefore, the resulting element stiffness will be 12 x 12. However, if the strain energy within the element is minimized with respect to α_i four additional equations can be generated and the additional displacements can be eliminated and a reduced 8 x 8 stiffness matrix developed. This is identical to the standard static condensation procedure. An alternate approach is to consider α_i Lagrange multipliers.

The functions h_5 and h_6 can be selected to be of the same form as the errors in the bending deformation which is given by equation (5). Or

$$h_5 = (1-s^2)$$
$$h_6 = (1-t^2)$$

If the element is rectangular only the y displacements would need to be modified. However, for the general quadrilateral these modes must be added to both components of displacements.

These incompatible modes are plotted in figure (3). It is apparent that the energy associated with these modes is large compared to the constant strain modes. It is also clear that they will not participate significantly in areas of low stress gradients since they are added to the basic constant strain modes. The net result of the addition of the incompatible modes is that microscopic equilibrium is better satisfied within the element.

Two-Dimensional Examples

A. Cantilever Beam - A cantilever beam with the dimensions shown in figure (4) will illustrate the accuracy of the element for plane stress structures. Results for two different loading conditions and for two different meshes are shown in Table 1. They are compared with exact solution and with a finite element solution without incompatible modes. For this case the improvement is significant. The Q6 is the element presented in this paper. The Q4 is the standard isoparametric quadrilateral element.

B. Axisymmetric Cylindrical Shell - The same basic expansion is used for the analysis of axisymmetric solids. An infinite cylindrical shell is idealized by 17 axisymmetric elements as shown in figure (5). In Table 2 results are compared with a theoretical solution and to two other types of finite elements. The Q4CST is composed of four constant strain triangles and the QM5 is quadrilateral with a restricted integration on the shear strains.

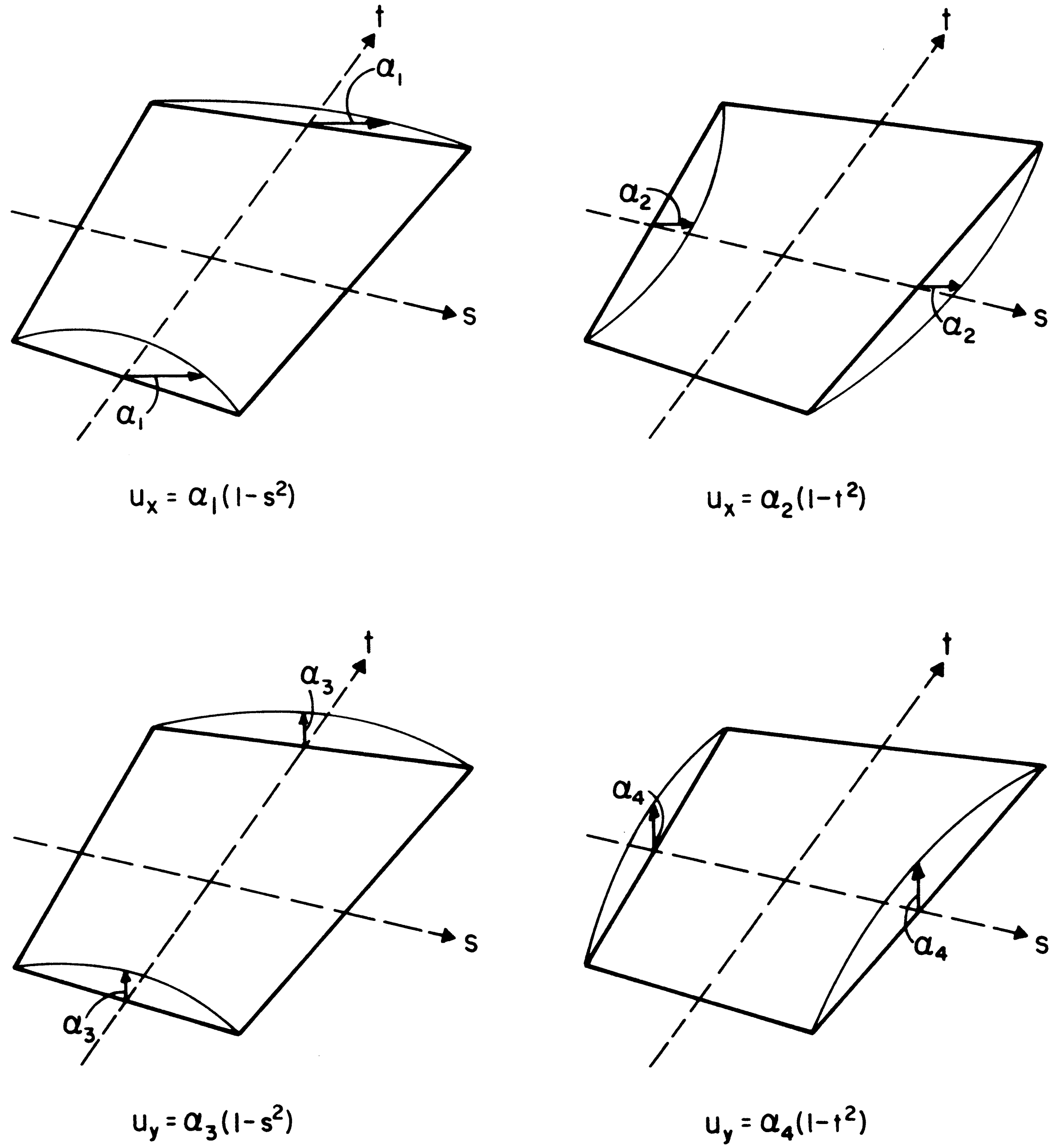


FIGURE 5-3 INCOMPATIBLE DISPLACEMENT MODES FOR GENERAL QUADRILATERAL ELEMENT

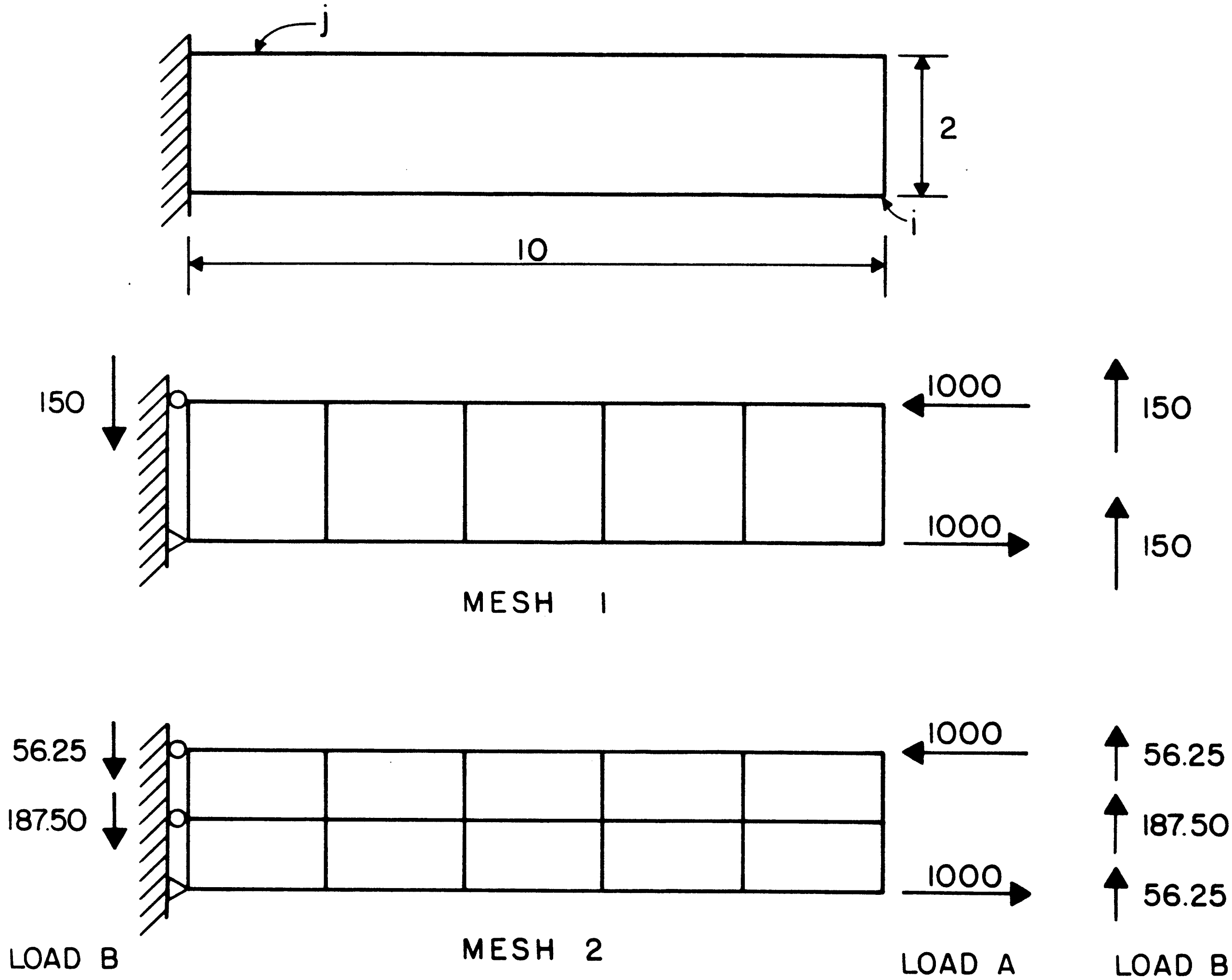


FIGURE 5-4 CANTILEVER BEAM-PLANE STRESS

	Displacement at i		Bending Stress at j	
	Load A	Load B	Load A	Load B
Beam Theory	10.00	103.0	300.0	4050
Q4 Mesh 1	6.81	70.1	218.2	2945
Q4 Mesh 2	7.06	72.3	218.8	2954
Q6 Mesh 1	10.00	101.5	300.0	4050
Q6 Mesh 2	10.00	101.3	300.0	4050

TABLE 5-1 RESULTS OF CANTILEVER BEAM ANALYSIS

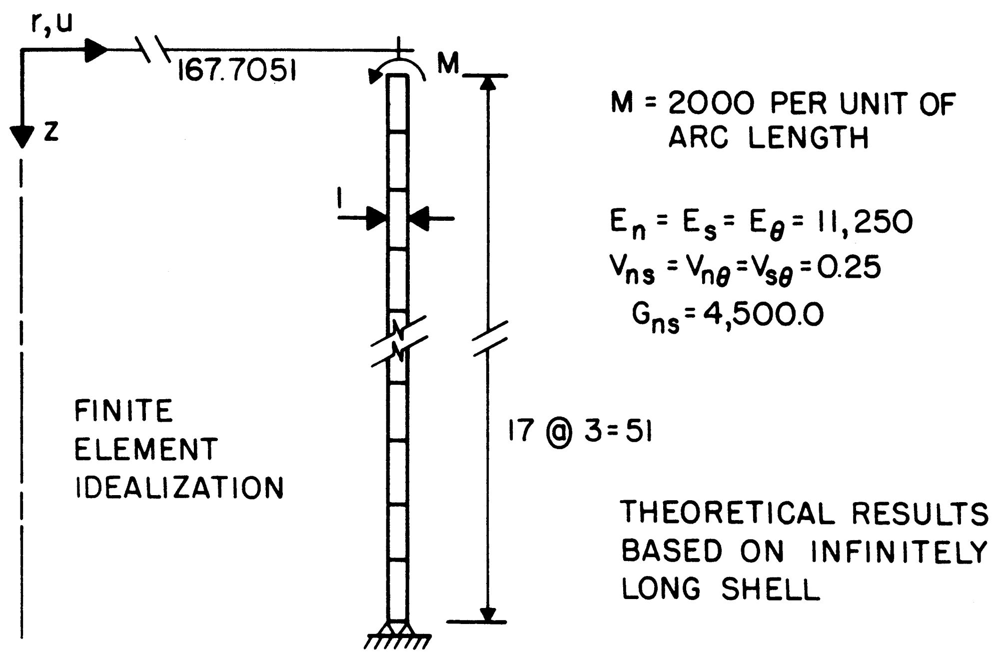


FIGURE 5-5 CYLINDRICAL SHELL ANALYSIS

LATERAL DISPLACEMENTS u					
Z	THEORY	Q4CST	QM5	Q4	Q6
0	100.00	39.97	98.56	46.47	100.01
3	48.88	26.04	47.87	29.17	48.98
6	14.31	14.98	13.49	15.69	14.19
9	-6.57	6.56	-7.29	5.69	-6.54
12	-17.16	0.47	-17.77	-1.31	-17.15
15	-20.68	-3.65	-21.17	-5.82	-20.70
18	-19.85	-6.16	-20.21	-8.35	-19.88
21	-16.75	-7.40	-16.97	-9.39	-16.83
24	-12.82	-7.68	-12.92	-9.33	-12.85
27	-8.95	-7.27	-8.98	-8.55	-9.00
30	-5.63	-6.40	-5.65	-7.32	-5.72
33	-3.06	-5.27	-3.12	-5.87	-3.23

HOOP STRESSES					
Z	THEORY	Q4CST	QM5	Q4	Q6
1.5	4846	2210	4903	2536	4837
4.5	1991	1369	2050	1507	1986
7.5	159	716	201	718	154
10.5	-868	230	-846	147	-873
13.5	-1316	-120	-1311	-238	-1320
16.5	-1386	-334	-1392	-475	-1390
19.5	-1240	-459	-1250	-595	-1243
22.5	-994	-510	-1006	-627	-996
25.5	-727	-505	-738	-599	-729
28.5	-483	-462	-493	-532	-487
31.5	-285	-395	-297	-442	-293
34.5	-138	-315	-155	-344	153

TABLE 5-2 RESULTS OF CYLINDRICAL SHELL ANALYSIS

THREE DIMENSIONAL ELEMENTS

The same basic method of introducing incompatible displacement modes in order to improve the bending properties can be used in three dimensions. For an arbitrary eight point brick element shown in figure (6) the appropriate displacement approximations are

$$u_x = \sum_{i=1}^{8} u_{xi} + h_9\,\alpha_{x1} + h_{10}\,\alpha_{x2} + h_{11}\,\alpha_{x3}$$

$$u_y = \sum_{i=1}^{8} u_{yi} + h_9\,\alpha_{y1} + h_{10}\,\alpha_{y2} + h_{11}\,\alpha_{y3}$$

$$u_z = \sum_{i=1}^{8} u_{zi} + h_9\,\alpha_{z1} + h_{10}\,\alpha_{z2} + h_{11}\,\alpha_{z3}$$

where

$$h_1 = 1/8\,(1 + \xi)\,(1 + \eta)\,(1 + \zeta)$$
$$h_2 = 1/8\,(1 - \xi)\,(1 + \eta)\,(1 + \zeta)$$
$$h_3 = 1/8\,(1 - \xi)\,(1 - \eta)\,(1 + \zeta)$$
$$h_4 = 1/8\,(1 + \xi)\,(1 - \eta)\,(1 + \zeta)$$
$$h_5 = 1/8\,(1 + \xi)\,(1 + \eta)\,(1 - \zeta)$$
$$h_6 = 1/8\,(1 - \xi)\,(1 + \eta)\,(1 - \zeta)$$
$$h_7 = 1/8\,(1 - \xi)\,(1 - \eta)\,(1 - \zeta)$$
$$h_8 = 1/8\,(1 + \xi)\,(1 - \eta)\,(1 - \zeta)$$
$$h_9 = (1 - \xi^2)$$
$$h_{10} = (1 - \eta^2)$$
$$h_{11} = (1 - \zeta^2)$$

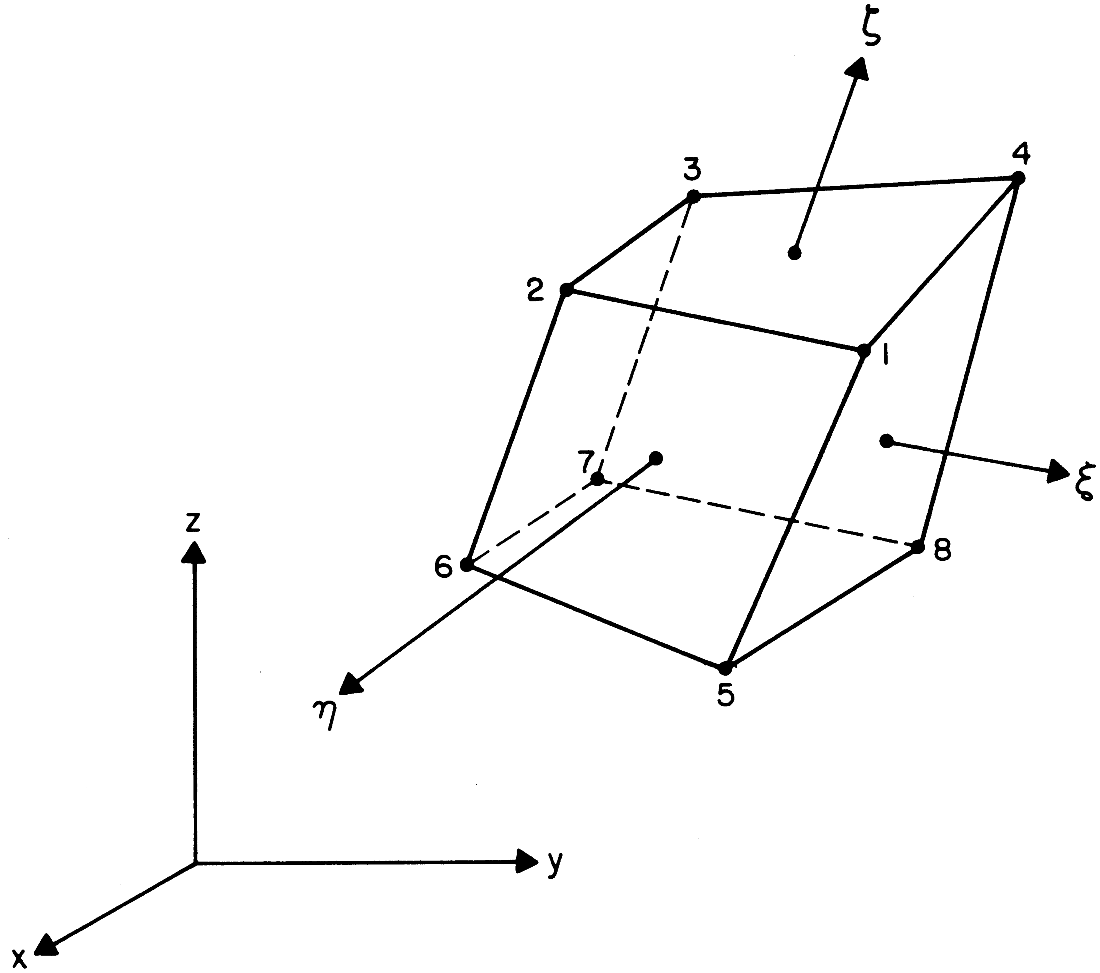


FIGURE 5-6 EIGHT POINT THREE DIMENSIONAL ELEMENT

The first eight are the standard compatible interpolation functions. The last three are incompatible and are associated with linear shear and normal strains. The nine incompatible modes are eliminated at the element stiffness level by static condensation.

Since the three-dimensional element degenerates to the same approximation as in the two-dimensional element the same general improvement in accuracy is obtained. Therefore, additional examples of its behavior will not be given here.

This element has been found to be extremely effective in the analysis of massive three-dimensional structures subjected to bending. One element in the thickness direction of arch dams or thick pipe joints have been found to be adequate. Hence, the three-dimensional analysis of this class of structure involves a reasonable amount of computer time.

Extension to 20 Node Elements

The modified eight node three-dimensional element has been found to be comparable in accuracy with the standard 20 node element. This is because the stresses associated with constant moment are included in the normal 20 node approximation. It is apparent that the addition of incompatible modes to the 20 node elements will improve its behavior. Three interpolation functions which are associated with linearly varying moments are

$$h_{21} = \xi\,(1 - \xi^2)$$
$$h_{22} = \eta\,(1 - \eta^2)$$
$$h_{23} = \zeta\,(1 - \zeta^2)$$

At this time the author does not have experience with this type of modification. It is possible that the increase in computational effort to form a 69 x 69 matrix and to reduce it to a 60 x 60 matrix will not be justified.

THICK SHELL ELEMENT

It is possible to use standard three-dimensional elements for the analysis of shell type structures. Practically, this has not been possible because of the following three problems:

1. Most three-dimensional solid elements have not had the ability to represent bending moments. (Elements with four nodes along each edge have this property).
2. Errors in the shear and normal strains cause the element to be very stiff.
3. Because of the relatively large stiffness coefficients in thickness direction numerical problems are introduced for thin shells.

The first two problems can be overcome by the introduction of incompatible modes. The third problem can be minimized by the use of a computer with high precision or by restricting the application of the element to thick shells.

The shell element presented in this paper is a 16 node curved solid element shown in figure (7). Each node has three unknown displacements. Therefore, if the shell is considered as a two-dimensional surface there are six unknowns per point. It is apparent that this type of formulation avoids the problems associated with the sixth degree of freedom - the normal rotation is set to zero when certain finite elements are used in the idealization of shells.

The locations of the nodes are defined by the orthogonal, right-handed coordinate system (x, y, z) which is referred to as a global

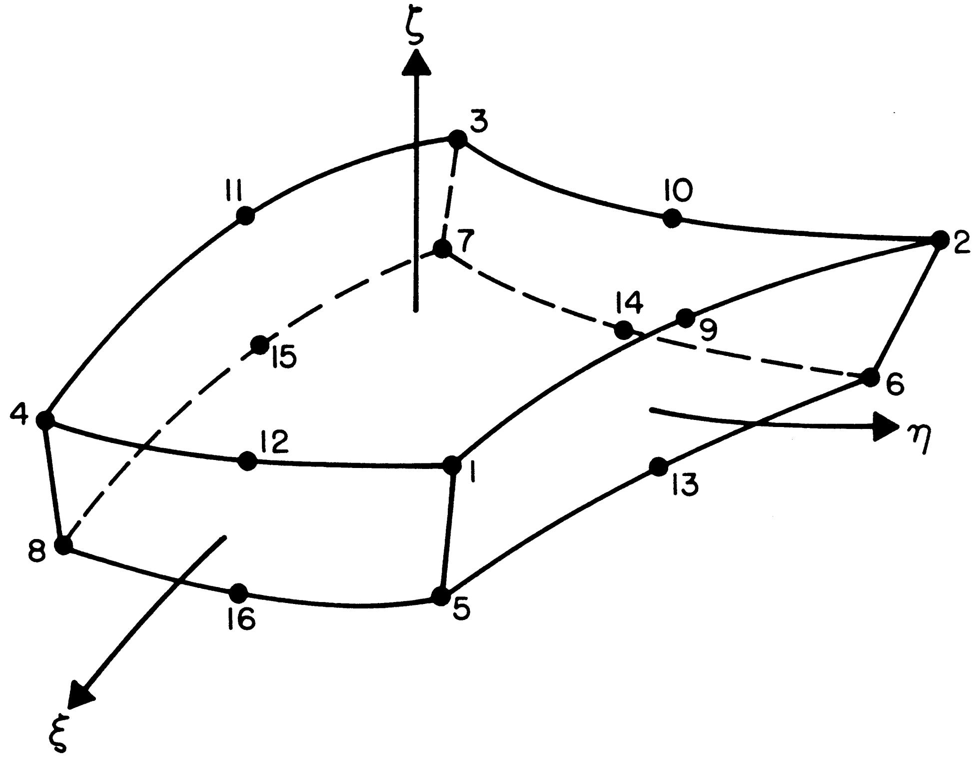


FIGURE 5-7 THREE DIMENSIONAL THICK-SHELL ELEMENT

system. Within the element a local coordinate system (ξ, η, ζ) has been chosen such that ξ, η, ζ vary from -1 to +1; (0, 0, 0) is located at the centroid of the element.

The local and global coordinate systems are related through a set of interpolating functions:

$$x = \sum_{i=1}^{16} h_i \; x_i$$

$$y = \sum_{i=1}^{16} h_i \; y_i$$

$$z = \sum_{i=1}^{16} h_i \; z_i$$

where

$$h_1 = 1/8\ (1 + \xi)\ (1 + \eta)\ (1 + \zeta)\ (\xi + \eta - 1)$$

$$h_2 = 1/8\ (1 - \xi)\ (1 + \eta)\ (1 + \zeta)(-\xi + \eta - 1)$$

$$h_3 = 1/8\ (1 - \xi)\ (1 - \eta)\ (1 + \zeta)(-\xi - \eta - 1)$$

$$h_4 = 1/8\ (1 + \xi)\ (1 - \eta)\ (1 + \zeta)\ (\xi - \eta - 1)$$

$$h_5 = 1/8\ (1 + \xi)\ (1 + \eta)\ (1 - \zeta)\ (\xi + \eta - 1)$$

$$h_6 = 1/8\ (1 - \xi)\ (1 + \eta)\ (1 - \zeta)(-\xi + \eta - 1)$$

$$h_7 = 1/8\ (1 - \xi)\ (1 - \eta)\ (1 - \zeta)(-\xi - \eta - 1)$$

$$h_8 = 1/8\ (1 + \xi)\ (1 - \eta)\ (1 - \zeta)\ (\xi - \eta - 1)$$

$$h_9 = 1/4\ (1 - \xi^2)\ (1 + \eta\ )\ (1 + \zeta)$$

$$h_{10} = 1/4\ (1 - \xi\ )\ (1 - \eta^2)\ (1 + \zeta)$$

$$h_{11} = 1/4\ (1 - \xi^2)\ (1 - \eta\ )\ (1 + \zeta)$$

$$h_{12} = 1/4\ (1 + \xi\ )\ (1 - \eta^2)\ (1 + \zeta)$$

$$h_{13} = 1/4\ (1 - \xi^2)\ (1 + \eta\ )\ (1 - \zeta)$$

$$h_{14} = 1/4\ (1 - \xi\ )\ (1 - \eta^2)\ (1 - \zeta)$$

$$h_{15} = 1/4\ (1 - \xi^2)\ (1 - \eta\ )\ (1 - \zeta)$$

$$h_{16} - 1/4\ (1 + \xi\ )\ (1 - \eta^2)\ (1 - \zeta)$$

The displacements within the element are assumed to be of the following form:

$$u_x = \sum_{i=1}^{16} h_i\, u_{xi} + h_{17}\,\alpha_{x1} + h_{18}\,\alpha_{x2} + h_{19}\,\alpha_{x3} + h_{20}\,\alpha_{x4} + h_{21}\,\alpha_{x5}$$

$$u_y = \sum_{i=1}^{16} h_i\, u_{yi} + h_{17}\,\alpha_{y1} + h_{18}\,\alpha_{y2} + h_{19}\,\alpha_{y3} + h_{20}\,\alpha_{y4} + h_{21}\,\alpha_{y5}$$

$$u_z = \sum_{i=1}^{16} h_i\, u_{zi} + h_{17}\,\alpha_{z1} + h_{18}\,\alpha_{z2} + h_{19}\,\alpha_{z3} + h_{20}\,\alpha_{z4} + h_{21}\,\alpha_{z5}$$

where;

$$h_{17} = \xi\,(1 - \xi^2)$$

$$h_{18} = \eta\,(1 - \eta^2)$$

$$h_{19} = (1 - \zeta^2)$$

$$h_{20} = \xi\eta\,(1 - \xi^2)$$

$$h_{21} = \eta\xi\,(1 - \eta^2)$$

The motivation for addition of the interpolation functions h_{17} to h_{21} is to increase the capability of the element in producing closer approximations to the exact displacements under simple loadings, thereby increasing the convergence to the exact solution. The incompatible interpolation functions h_{17} to h_{21} have zero values at the nodes and produce incompatibilities in the displacement field along the inter-element boundaries.

THICK SHELL EXAMPLES

The following examples are intended to demonstrate the range of applicability of this element. Two parameters were recognized to have significant effect on the behavior of the element, namely: the ratio of thickness to the length along the surface (t/a) and the ratio of the length along the surface to the radius of curvature ($\phi = a/R$). The effect of the first parameter is studied by the example of square simply supported plate and the effect of the second parameter is studied by a series of curved cantilever beam examples.

A. Square Simply Supported Plate -- The center deflection of a square simply supported plate for three different meshes and two values of thickness to length ratio is shown in figure (8). It is seen that this element is more appropriate for moderately thick to thick shell problems with thickness to length ratio of more than 1/20, although final convergence is assured for thin shell problems also.

It is worthwhile to mention that the finite element result in figure (8) exceeds that of Poisson-Kirchhoff which ignores the shear deformation. This is due to the fact that the thick shell element is capable of undergoing shear deformation.

B. Curved Cantilever Beam -- The results of these examples are shown in figure (9). It is seen from these examples that the accuracy is only slightly affected by curvature for moderately thick to thick shell problems, whereas the high curvature affects the accuracy of the thin shell problems rather drastically.

The effect of the curvature can be explained by the fact that the number of terms of polynomial expansion of the displacement field

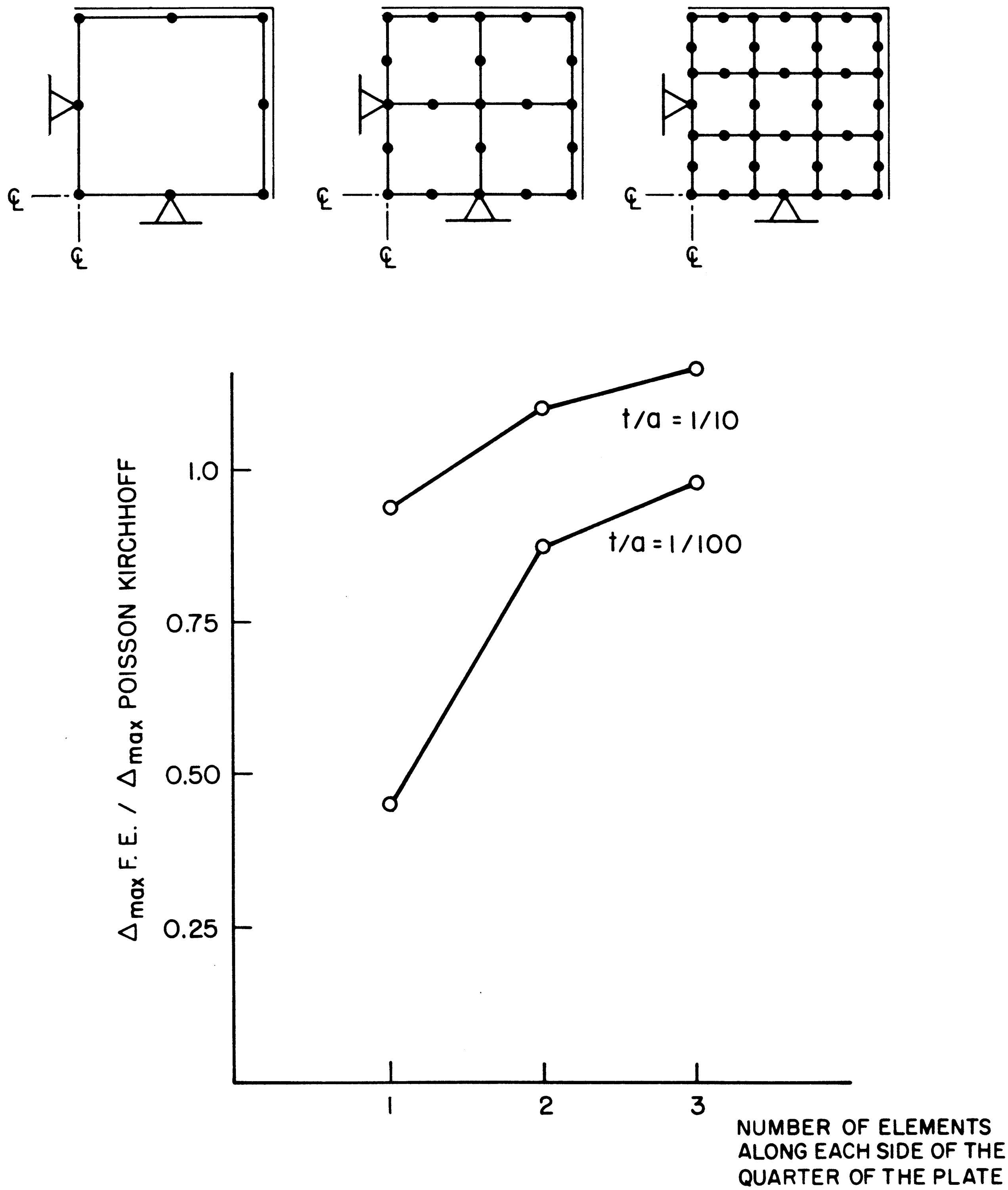


FIGURE 5-8 SIMPLY SUPPORTED SQUARE PLATE WITH A CONCENTRATED LOAD AT THE CENTER

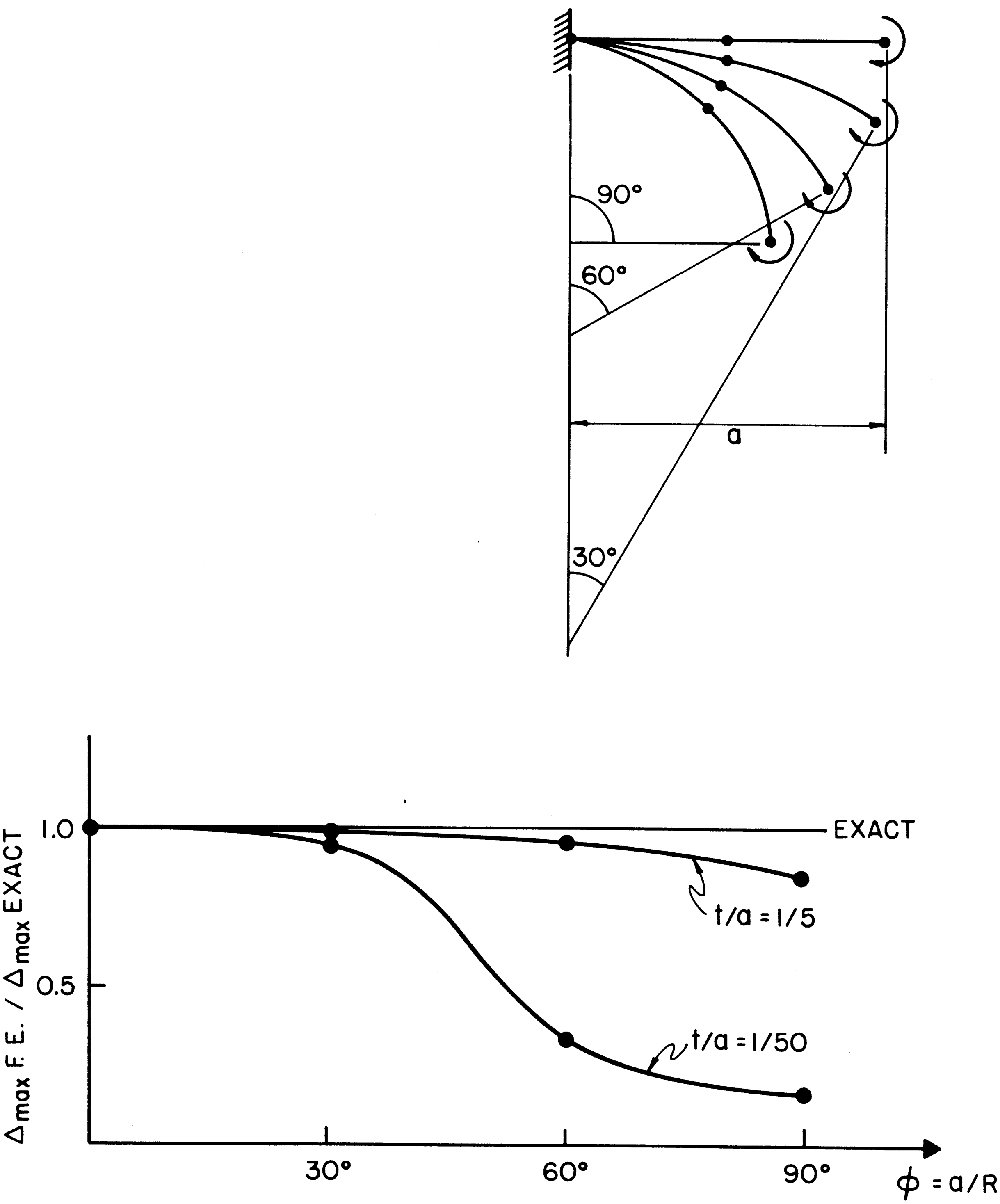


FIGURE 5-9 CURVED CANTILEVER BEAM

included in the displacement expansion of the element is limited, whereas the significance of the higher order terms in the polynomial expansion of the exact displacement increases with curvature. This effect is also reduced to zero at the limit as the mesh is refined, thereby assuring the convergence.

VI. PLATE AND THIN SHELL ELEMENT

The thin shell element included in this report is a quadrilateral of arbitrary geometry formed from four compatible triangles. The bending properties of this element are completely described in reference [9]. The element employs a partially restrained linear strain triangle to represent the membrane behavior. As shown in Figure 6-1, the central node is located at the average of the coordinates of the four courner nodes. The element has 17 interior degrees of freedom which are eliminated at the element level prior to assembling; therefore, the resulting quadrilateral element has 20 degrees of freedom, five per node, in the local element coordinate system.

For flat plates the stiffness associated with the rotation normal to the shell surface is not defined; therefore, the appropriate boundary condition must be enforced. For curved shells, the normal rotation can be included as an extra degree of freedom; or, it can be restrained by the addition of a "Boundary Element" which would add normal rotational stiffness to the node.

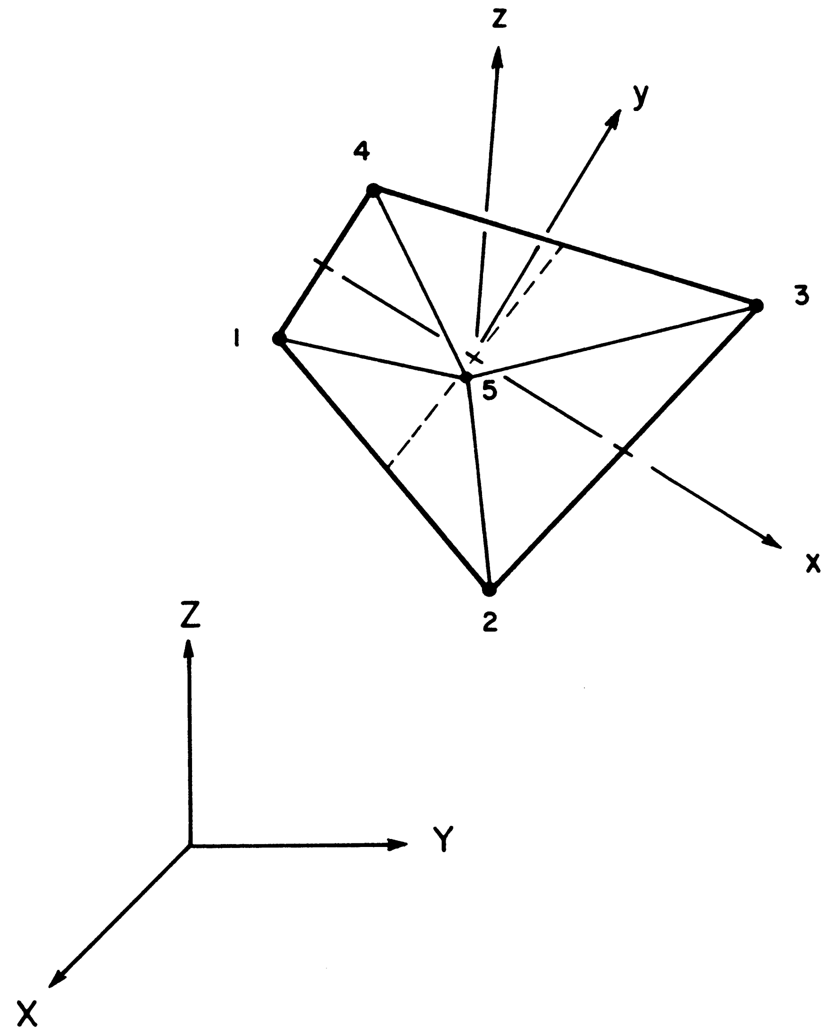


FIGURE 6-1 THIN SHELL ELEMENT

VII. BOUNDARY ELEMENT

The boundary element can be used for the following:

1. In the idealization of an external elastic support at a joint.
2. In the idealization of an inclined roller support.
3. To specify a joint displacement
4. To eliminate the numerical difficulty associated with the "sixth" degree of freedom in the analysis of shells.

The element is a one dimensional element with an axial and torsional stiffness. These element stiffness coefficients are added directly to the total stiffness matrix. If a displacement is to be specified a load must be applied in the direction of the stiffness. If the boundary element stiffness is large compared to the stiffness of the structures it is possible to apply a load to produce the desired displacement.

VIII. COMPUTER PROGRAM ORGANIZATION

The computer program is coded in standard FORTRAN IV and is practically machine independent. All storage is allocated at the time of execution; therefore, the minimum storage required will depend on the size of the structure. To increase the capacity of the program it is necessary to change two cards as described in Appendix C.

For static analysis the program is divided into four phases. A machine dependent overlay system is not used; instead, a COMMON storage area is used in each phase. These four are executed in the following sequence:

1. Data Input - Joint coordinates and loads are read or generated. As element properties are read or generated the element stiffness matrices are formed and placed on tape (or other low speed storage).

2. Formation of total stiffness is accomplished by reading the element stiffness tape and forming the joint equilibrium equations in blocks.

3. Equilibrium equations are solved for joint displacements, all load conditions are treated at the same time.

4. From the joint displacements, element stresses are calculated for all load conditions.

In the following sections these are explained in greater detail.

6.1 Solution of Equations

The computer program is built around a large capacity linear equation solver, USOL. The procedure used to solve the equations is not significantly different from the method developed by Gauss in 1827. The banded characteristics of the equations are recognized. Operations with zero coefficients are skipped. Data is transferred in and out of high speed storage in large blocks; therefore, a small amount of time is lost in the transfer of data.

The equilibrium equations (the stiffness matrix and loads) are stored in blocks on tape (or other low speed storage units). During the solution phase two blocks must be in high speed storage at any time. Therefore, the physical storage restriction is that there must be high speed storage available for at least two equations. For example, if the stiffness matrix has a band width of 250 and if there are 20,000 high speed storage locations available, the number of equations in a block will be 40. Hence, for this example all data transfer will be in blocks of 10,000. The block size is automatically determined at the time of solution. Therefore, storage is utilized in the most efficient manner for a particular structure.

6.2 Formation of Equilibrium Equations

Before the total stiffness matrix is formed the element stiffness matrices are calculated and stored in sequence on low speed storage. The total stiffness matrix is formed two blocks at a time by making a pass through the element stiffness matrices and adding in the appropriate coefficients. In order to minimize the effort in searching through all the element stiffnesses the element stiffness matrices for several blocks are transferred to another storage unit; therefore, in the formation of the next several blocks the time to search for the contributions to these blocks is reduced significantly.

6.3 Joint Input Data and Degrees of Freedom

The capacity of the program is controlled by the number of joints (nodal points) of the structural system. All joint data is retained in high speed storage during the formation of the element stiffness matrices. For each joint three coordinates and six boundary condition codes are required; therefore, the minimum required storage for a given problem is nine times the number of joints in the system.

Immediately after the joint data is supplied to the program a relationship between each joint degree of freedom and the corresponding equation number is established. Each of the six boundary condition codes for a given joint is replaced by the equation number for that degree of freedom. Restrained boundary conditions are identified by a zero equation number. Slave degrees of freedom (for beam elements) are identified by the negative joint number of the master node.

6.4 Calculation of Element Stiffness Matrices

After the coordinates of the joints are supplied and the equation numbers of the degrees of freedom established the stiffness and stress-displacement transformation matrices are calculated for each structural element in the system. Very little additional high speed storage is required for this phase since these matrices can be formed and placed on tape storage as the element properties are read. In addition to the element matrices the corresponding equation numbers are written on tape. After all element matrices are formed the joint coordinates and boundary condition information are not required; hence, this storage area can be used subsequently for storage for the two blocks of the equilibrium equations. It is now possible to form and solve these equations as previously described.

6.5 Evaluation of Element Stresses

After the joint displacements are evaluated a pass is made through the element stress-displacement matrix tape and the element stresses are calculated and printed.

REFERENCES

1. B. Irons, "Stress Analysis by Stiffnesses Using Numerical Integration," Rolls Royce Co. (Internal Report), June 1963.

2. O. C. Zienkiewicz, B. M. Irons, J. Ergatoudis, S. Ahmad, and F. C. Scott, "Isoparametric and Associated Element Families for Two and Three Dimensional Analysis," Ch. 13 of Finite Element Method in Stress Analysis (ed. I. Holand and K. Bell), Tapir, Trondheim, 1969.

3. S. Ahmad, B. M. Irons, and O. C. Zienkiewicz, "Curved Thick Shell and Membrane Elements with Particular Reference to Axisymmetric Problems," Proc. of 2nd Conf. on Methods in Struct. Mech., Wright Patterson AFB, Ohio, 1968.

4. W. P. Doherty, E. L. Wilson, and R. L. Taylor, "Stress Analysis of Axisymmetric Solids Utilizing Higher Order Quadrilateral Finite Elements," SESM Report 69-3, Structural Engineering Laboratory, University of California, Berkeley, California, 1969.

5. O. C. Zienkiewicz, R. L. Taylor, and J. M. Too, "Reduced Integration Techniques in General Analysis of Plates and Shells," Int. Journal for Numerical Methods in Engineering, Vol. 3, pp. 275-290, (1971).

6. S. F. Pawsey, "The Analysis of Moderately Thick to Thin Shells by the Finite Element Method," SESM Report 70-12, Structural Engineering Laboratory, University of California, Berkeley, California, p. 102, (1970).

7. O. C. Zienkiewicz, "The Finite Element Method in Structural and Continuum Mechanics," McGraw-Hill Ltd., London, 1967.

8. E.L. Wilson, "SAP, A General Structural Analysis Program," University of California Structural Engineering Laboratory Report No. UCSESM 70-20, September, 1970.

9. R. W. Clough and C. A. Felippa, "A Refined Quadrilateral Element for Analysis of Plate Bending," Proc. of 2nd Conf. on Matrix Methods in Structural Mechanics, Wright Patterson A. F. B., Ohio, 1968.

APPENDIX A - THE STATIC CONDENSATION ALGORITHM

The application of the static condensation method is discussed in general. The technique is presented as an extension of the Gauss elimination algorithm. An efficient Fortran subroutine is given which simultaneously reduces both the stiffness matrix and the stress-displacement transformation matrix.

A-1 Introduction

The static condensation method was initially used to eliminate the internal degree of freedom in a quadrilateral finite element constructed from four triangles [1]. The method is more general then the application at the element stiffness level and can be used to reduce the number of degrees of freedom of the complete structural system. In many cases, it is similar to the substructure technique, frontal solution method, or matrix partitioning. However, all of these methods appear to be an application of the basic Gaussian elimination procedure.

The elimination of internal degrees of freedom at the element stiffness level reduces the overall size and band width of the resulting set of equations. In the case of dynamic analysis the elimination of the massless degrees of freedom reduces the size of the resulting eigenvalue problem. The terminology "static" condensation was coined in the application of the method to dynamic analysis [2].

The purpose of this paper is to present the static condensation procedure as an efficient numerical algorithm. In addition, the procedure is extended to the condensation of the stress-displacement transformation matrix; therefore, the calculation of the eliminated degrees of freedom is not required in the subsequent evaluation of the element stresses.

A-2 Matrix Formulation Of The Static Condensation Method

The basic static condensation procedure can be illustrated by the use of matrix notation. The equilibrium equations can be written in matrix partitional forms as

$$\begin{bmatrix} \underline{K}_{aa} & \underline{K}_{ab} \\ \underline{K}_{ba} & \underline{K}_{bb} \end{bmatrix} \begin{bmatrix} \underline{u}_a \\ \underline{u}_b \end{bmatrix} = \begin{bmatrix} \underline{P}_a \\ \underline{P}_b \end{bmatrix} \quad \text{(A.1)}$$

where $\underline{u}_a$ indicates the degrees of freedom to be eliminated and $\underline{u}_b$ indicates the degrees of freedom which are associated with the reduced stiffness matrix. The solution of the first submatrix equation for $\underline{u}_a$ yields

$$\underline{u}_a = \underline{C} - \underline{T}\,\underline{u}_b \quad \text{(A.2)}$$

in which

$$\underline{C} = \underline{K}_{aa}^{-1}\,\underline{P}_a \quad \text{(A.3)}$$

$$\underline{T} = \underline{K}_{aa}^{-1}\,\underline{K}_{ab} \quad \text{(A.4)}$$

Substitution of Equation (2) into the second submatrix equation results in a set of equilibrium equations with respect to the "b" degrees of freedom.

$$\underline{K}^{*} \underline{u}_b = \underline{P}^{*} \tag{A.5}$$

where

$$\underline{K}^{*} = \underline{K}_{bb} - \underline{K}_{ba} \underline{T} \tag{A.6}$$

$$\underline{P}^{*} = \underline{P}_b - \underline{K}_{ba} \underline{C} \tag{A.7}$$

Physically, the term $\underline{K}_{ba} \underline{T}$ indicates the stiffness modification due to the release of the "a" degrees of freedom and $\underline{K}_{ba} \underline{C}$ represents the forces carried over from the "a" to the "b" degrees of freedom.

The stresses within the elements may be expressed by an equation of the form

$$\sigma = [\underline{A}_a \ \underline{A}_b] \begin{bmatrix} \underline{u}_a \\ \underline{u}_b \end{bmatrix} + \underline{\tau} \tag{A.8}$$

where $\underline{\tau}$ are the initial stresses before the system is subjected to the displacements $\underline{u}_a$ and $\underline{u}_b$. The displacement $\underline{u}_b$ may be eliminated from Equation (8) by the substitution of Equation (2). Or

$$\underline{\sigma} = \underline{A}^{*} \underline{u}_a + \underline{\tau}^{*} \tag{A.9}$$

where

$$\underline{A}^{*} = \underline{A}_b - \underline{A}_a \underline{T} \tag{A.10}$$

$$\underline{\tau}^{*} = \underline{\tau} + \underline{A}_a \underline{C} \tag{A.11}$$

As an example of the application of this procedure, consider a quadrilateral element formed by the combination of four six-node triangles as shown in Figure 1. In the case of two-dimensional stress problems this element has 13 nodes or 26 degrees of freedom; however, the 10 degrees of freedom associated with the interior points may be eliminated before the element stiffness is added to the total stiffness of the structure. If the stress matrices $\underline{A}^*$ and $\underline{\tau}^*$ are evaluated at the same time the reduced stiffness matrix $\underline{K}^*$ is formed, the internal displacements $\underline{u}_a$ need not be calculated. Of course, this will require that the matrices $\underline{A}^*$ and $\underline{\tau}^*$ be saved on a low speed storage unit until after the displacements $\underline{u}_b$ are evaluated.

A-3 The Static Condensation Algorithm

The matrix notation used in the previous section serves to illustrate the basic concept of the technique; however, the matrix operations of multiplication and inversion are not efficient within a computer program. Equation (1) can be written as

$$\underline{K}\,\underline{u} = \underline{P} \tag{A.12}$$

If this set of equations is for the element shown in Figure 1 and if the first ten unknowns are associated with the internal points, we can apply the Gauss elimination procedure directly to the system to eliminate these degrees of freedom. Also, if the stress-displacement equation is written as

$$\underline{\sigma} = \underline{A}\,\underline{u} + \underline{\tau} \tag{A.13}$$

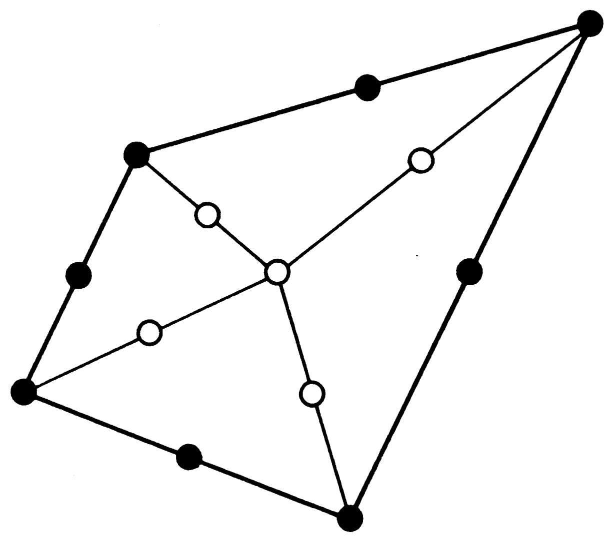


FIGURE A-1 QUADRILATERAL ELEMENT

the appropriate unknowns can be eliminated at the same time. If we consider three components of stress, the application of the Gauss algorithm to these equations can be summarized as follows:

for $n = 1, 10$

1. Solve for u_n

$$u_n = C_n - \sum_{j=n+1}^{26} T_{nj} u_j$$

where

$$C_n = P_n^* / K_{nn}^*$$

$$T_{nj} = K_{nj}^* / K_{nn}^*$$

2. Substitute into remaining equations will result in the modification of the coefficients

$$\left.\begin{aligned} K_{ij}^{**} &= K_{ij}^* - K_{in}^* \, T_{nj} \\ P_i^{**} &= P_i^* - K_{in}^* \, C_n^* \end{aligned}\right\} \quad \begin{aligned} i &= n+1, 26 \\ j &= n+1, 26 \end{aligned}$$

3. Substitute into the stress-displacement equations

$$\left.\begin{aligned} A_{ij}^{**} &= A_{ij}^* - A_{in}^* \, T_{nj} \\ \tau_i^{**} &= \tau_i^* - A_{in}^* \, C_n \end{aligned}\right\} \quad \begin{aligned} i &= 1, 3 \\ j &= n+1, 26 \end{aligned}$$

The $*$ indicates the repeated modification of the coefficients for each value of n. Within the computer program these coefficients are modified and stored at the same storage location.

The Fortran statements which apply to the elimination of these ten degrees of freedom are

```
         DO  500  N=1,10
         NN=N+1
         C=P(N)/S(N,N)
         DO  300  J=NN,26
         P(J)=P(J)-S(J,N)*C
         T=S(N,J)/S(N,N)
         DO  200  I=NN,26
   200   S(I,J)=S(I,J)-S(I,N)*T
         DO  300  I=1,3
   300   A(I,J)=A(I,J)-A(I,N)*T
         DO  400  I=1,3
   400   TAU(I)=TAU(I)-A(I,N)*C
   500   CONTINUE
```

where the stiffness term K_{ij} is indicated in Fortran as S(I,J). In this case, it is apparent that the Fortran statements are a very efficient summary of the algorithm. The results of the reduction are stored in the same area as the original matrices, with the first ten columns of the S and A array and the first ten rows of the S array eliminated.

A more convenient form for computer programming is to store the coefficients to be eliminated in sequence after the coefficients which are to be modified; then, the subscripts for the reduced stiffness start with one and rearrangement of the reduced coefficients is not required.

APPENDIX B - DESCRIPTION OF INPUT DATA

B-1 Introduction

The purpose of this computer program is to perform linear, elastic analyses of three dimensional structural systems. The structural systems to be analysed may be composed of combinations of a number of structural element types. The present version contains the following element types:

(1) Three-dimensional Truss Elements

A uniform temperature change and inertia loads in three directions can be considered as the basic element loads. Axial forces and stresses are computed.

(2) Three-dimensional Beam Elements

Beam elements are straight, prismatic beam members. Inertia loading (e.g. gravity) in three directions and specified fixed end forces form the element load cases. Forces (axial and shear) and moments (bending and torsion) are calculated in the beam local co-ordinate system.

(3) Plane Stress and Plane Strain Elements

An arbitrary quadrilateral (or triangular) element is used. The plane of the element is arbitrary with respect to a three dimensional coordinate system. Gravity, inertia, and temperature loadings may be considered. Stresses may be computed at the center of the element and at the center of each side.

(4) Axisymmetric Quadrilateral Elements

An arbitrary quadrilateral (or triangular) element is used. The element is axisymmetric about the global Z-axis and the Y direction is considered radial. Temperature, surface pressure and inertia (Z direction) loading are included. Stresses may be computed at the center of the element and at the center of each side.

(5) Three-dimensional Solid Elements

A general 8 nodal point "brick" element, with 3 translational degrees of freedom per nodal point, is used. Isotropic material properties are assumed, and element loading consists of temperature, surface pressure and inertia loads in three directions. Stresses (6 components) may be computed at the center of the element and at the center of each face.

(6) Plate and Shell Elements

An arbitrary quadrilateral element is used. Gravity, inertia, pressure and temperature loadings may be considered. Stresses are computed at the center of the element.

(7) Boundary Elements

This element is used to impose displacement boundary conditions and to compute support reactions.

(8) Thick Shell Elements

A 16 node curved solid element can be used for the analysis of thick shells or plates.

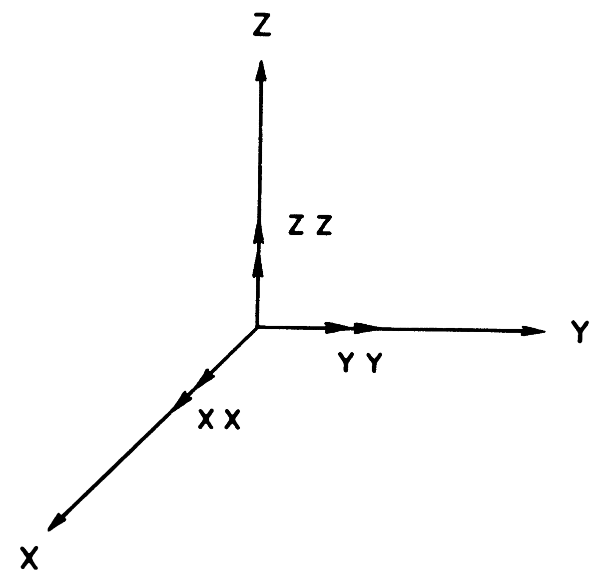
Systems composed of large numbers of joints and members may be analysed. The capacity of the program depends mainly on the total number of joints in the system. There is practically no restriction on the number of elements, number of load cases, or the "bandwidth" of the equations to be solved. Note, that while the program has the capacity to analyse very large systems, there is no loss of efficiency in the solution of smaller problems as compared to several special purpose programs presently available. For restrictions of program capacity see APPENDIX C.

B-2 Joints

Each joint in the system may have from 0 to 6 degrees of freedom as required. The user must ensure that the degrees of freedom specified for a given joint are compatible with the element-types which are adjacent to it.

Optimum solution efficiency is obtained by minimizing the number of degrees of freedom of the system. Also, joints connected only to Beam elements may use a special Slave-Master geometric constraint option to eliminate unnecessary degrees of freedom. (see Beam section)

A right-handed orthogonal co-ordinate system, shown below, is used to describe the geometry of the structure. All joint loads and displacements are defined with reference to this system. A local co-ordinate system is used for each element type.



GLOBAL COORDINATE SYSTEM

B-3 Loading

Loads may be applied by means of both point loads acting at the joints and by element loading (e.g. gravity, temperature). Each element has four different load cases called A, B, C, D. Loading for one solution consists of specified joint loads plus a linear combination of element load cases A, B, C and D. The types of loading which make up the element load cases A, B, C and D are defined in the description of each individual element type. Imposed displacement loading is possible by means of a special boundary element.

B-4 Input Data

The geometry of the joints, the boundary conditions, and the joint concentrated loads are numerically defined by a sequence of punched cards. The properties of the different structural elements are described separately.

I. Heading Card (12A6)

Columns	1 - 72	Contain information to be printed with output

II. Control Card (4I5)

Columns	1 - 5	Number of joints in system
	6 - 10	Number of element groups
	11 - 15	Number of load conditions
	16 - 20	Number of frequencies (= 0 for static analysis)

III. Joint Data (7I5,3F10.0,I5,F10.0)

The following information must be given for each joint in the system:

Columns	1 - 5	Joint Number	
	6 - 10	X-direction	Boundary Condition Codes: Zero or blank indicates that the joint is free to move in that direction and loads may be applied. One indicates that the joint is fixed in that direction.
	11 - 15	Y-direction	
	16 - 20	Z-direction	
	21 - 25	Rotation about X-axis	
	26 - 30	Rotation about Y-axis	
	31 - 35	Rotation about Z-axis	
	36 - 45	X-ordinate	
	46 - 55	Y-ordinate	
	56 - 65	Z-ordinate	
	66 - 70	KN	
	71 - 80	Joint Temperature	

Joint cards need not be in joint-order sequence. If cards are omitted, the joint data for a series of joints is generated. KN is a mesh generation parameter on the last card of a mesh generation sequence. KN is the increment to be added to the previous nodal point number. The intermediate joints are located at equal intervals along the straight line. The boundary condition codes for the generated joint data are set equal to the boundary condition codes on the first joint card in the series.

If a particular degree of freedom is fixed for a series of cards, this may be indicated by a boundary condition code of -1 on the first joint card in the series and +1 on the last joint card in the series. See element descriptions for determination of temperature dependent material properties and thermal loads from joint temperatures.

IV. Element Data

A sequence of cards is required for each type of element in the structure. The form of this data for each type is described in section B-5.

V. Concentrated Load Data (2I5,6F10.0)

One card per load case for each joint which has nonzero concentrated loads or moments applied. The cards must be in joint-number sequence.

Columns	1 - 5	Joint number
	6 - 10	Load condition number
	11 - 20	Load X-direction
	21 - 30	Load Y-direction
	31 - 40	Load Z-direction
	41 - 50	Moment X-axis
	51 - 60	Moment Y-axis
	61 - 70	Moment Z-axis

This sequence of cards (if any) must be terminated with ONE BLANK CARD.

VI. Element Load Multipliers (4F10.0)

Four different types of loads associated with the element are possible. These element loads are referred to as load cases A, B, C and D. By the use of "Element Load Multipliers," it is possible to add fractions of the basic element loads to any of the concentrated load conditions.

One card must be supplied for each load condition which contains the following information:

Columns	1 - 10	Multiplier for element load A
	11 - 20	Multiplier for element load B
	21 - 30	Multiplier for element load C
	31 - 40	Multiplier for element load D

These cards must be in load-order sequence. The definitions of the element loads associated with a particular element type are discussed in detail under the section "Element Data".

B-5 Element Data

Type 1 - Three-Dimensional Truss Members

Truss elements are identified by the number 1. Axial forces and stresses are calculated for each member. A uniform temperature change and inertia loads in three directions can be considered as the basic member load conditions. The truss members are described by the following sequence of cards:

A. Control Card (3I5)

Columns	1 - 5	The number 1
	6 - 10	Number of truss members
	11 - 15	Number of members with different properties

B. Member Property Cards (I5,5F10.0)

One card is required for each member which has a different cross-section or different material properties.

Columns	1 - 5	Material identification number
	6 - 15	Modulus of elasticity
	16 - 25	Coefficient of thermal expansion
	26 - 35	Blank
	36 - 45	Cross-sectional area
	46 - 55	Weight per unit length (used to calculate gravity loads)

C. Element Load Factors (4F10.0) Four cards

Three cards specifying the fraction of gravity (in each of the three global coordinate directions) to be added to each element load case.

Card 1: Multiplier of gravity load in the +X direction

Columns	1 - 10	Element load case A
	11 - 20	Element load case B
	21 - 30	Element load case C
	31 - 40	Element load case D

Card 2: As above for gravity in the +Y direction

Card 3: As above for gravity in the +Z direction

Card 4: This indicates the fraction of the thermal load to be added to each of the element load cases.

D. Member Data Cards (4I5,F10,0,I5)

One card per member in increasing numerical order starting with one.

Columns	1 - 5	Member number (n)
	6 - 10	Joint number I
	11 - 15	Joint number J
	16 - 20	Member identification number
	21 - 30	Reference temperature for zero stress
	31 - 35	Optional parameter K causing automatic generation of number data.

If a series of elements exist such that the member number, N_i, is one greater than the previous member number (i.e. $N_i = N_{i-1} + 1$) and the joint number can be given by

$$I_i = I_{i-1} + K$$

$$J_i = J_{i-1} + K$$

Then only the first element in the series need be provided. The member identification number and the temperature for the generated elements are set equal to the values on the first card. If K is input as zero it is set to 1 by the program.

The member temperature increase ΔT used to calculate thermal loads is given by

$$\Delta T = (T_i + T_j)/2.0 - T_r$$

where $(T_i + T_j)/2.0$ is the average of the nodal temperatures specified on the joint data cards for nodes i and j; and T_r is the zero stress reference temperature specified on the element card. For truss elements it is generally more convenient to set $T_i = T_j = 0.0$ such that $\Delta T = - T_r$ (note the minus sign). Other types of member loadings can be specified using an equivalent ΔT. If a truss member has an initial lack of fit by an amount d (positive if too long) then $\Delta T = d/(\alpha L)$ If an initial prestress force P (positive if tensile) is applied to the member ends that is released after the member is connected to the rest of the structure then $\Delta T = - P/(\alpha A E)$. In the above formulas A = cross section area, L = member length and α = coefficient of thermal expansion.

Type 2 - Three-Dimensional Beam Elements

Beam elements are identified by the number 2. Forces (axial and shear) and moments (bending and torsion) are calculated (in the beam local coordinate system) for each beam. Gravity loadings in each coordinate direction and specified fixed end forces form the basic member load conditions.

The beam members are described by the following sequence of cards:

A. Control Card (5I5)

Columns	1 - 5	The number 2
	6 - 10	Number of beam elements
	11 - 15	Number of geometric property cards
	16 - 20	Number of fixed end force sets
	21 - 25	Number of different materials

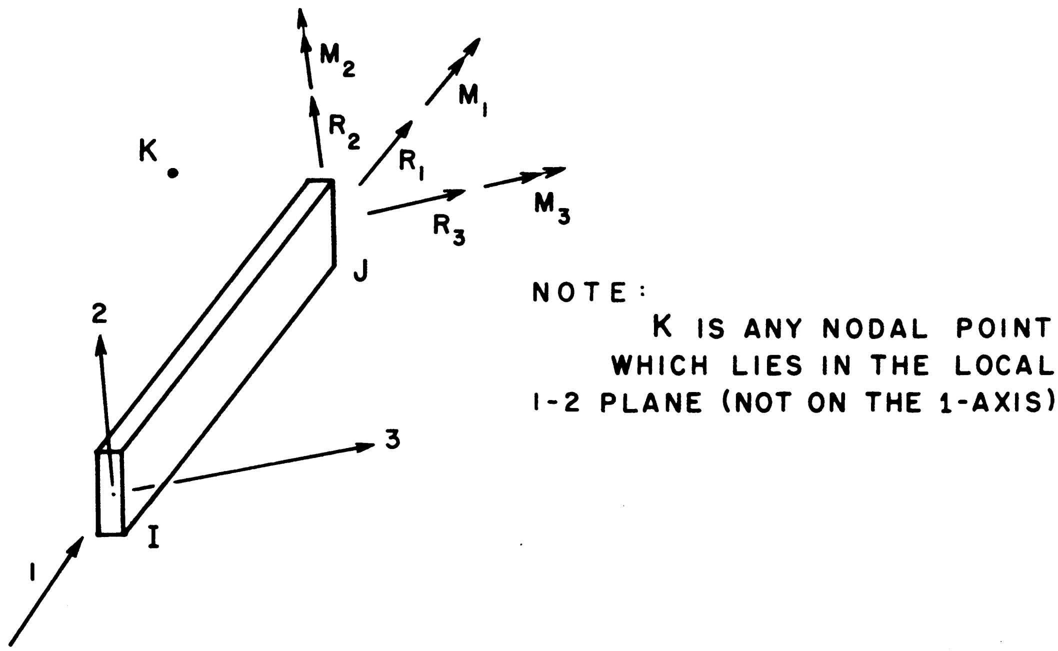
B. Material Property Cards (I5,3F10.0)

Columns	1 - 5	Material identification number
	6 - 15	Young's modulus
	16 - 25	Poisson's ratio
	26 - 35	Weight per unit length (to be used in gravity load calculations)

C. Geometric Property Cards (I5,6F10.0)

Columns	1 - 5	Geometric property number
	6 - 15	Axial area
	16 - 25	Shear area associated with shear forces in local 2-direction
	26 - 35	Shear area associated with shear forces in local 3-direction
	36 - 45	Torsional inertia
	46 - 55	Flexural inertia about local 2-axis
	56 - 65	Flexural inertia about local 3-axis

One card is required for each unique set of properties. Shear area is included only if shear deformations are to be included in the analysis.



LOCAL COORDINATE SYSTEM FOR BEAM ELEMENT

D. Element Load Factors (4F10.0)

Three cards specifying the fraction of gravity (in each of the three global coordinate directions) to be added to each element load case.

Card 1: Multiplier of gravity load in the +X direction

Columns	1 - 10	Element load case A
	11 - 20	Element load case B
	21 - 30	Element load case C
	31 - 40	Element load case D

Card 2: As above for gravity in the +Y direction

Card 3: As above for gravity in the +Z direction

Fixed-End Forces are not computed within the program for gravity loads

E. Fixed-End Forces (I5,6F10.0/I5,6F10.0)

Two cards are required for each unique set of fixed-end forces occurring in the analysis. Distributed loads and thermal loads are input using fixed end forces.

Card 1:

Columns	1 - 5	Fixed-end force number
	6 - 15	Fixed-end force in local 1-direction at Node I
	16 - 25	Fixed-end force in local 2-direction at Node I
	26 - 35	Fixed-end force in local 3-direction at Node I
	36 - 45	Fixed-end moment about local 1-direction at Node I
	46 - 55	Fixed-end moment about local 2-direction at Node I
	56 - 65	Fixed-end moment about local 3-direction at Node I

Card 2:

Columns	1 - 5	Blank
	6 - 15	Fixed-end force in local 1-direction at Node J
	16 - 25	Fixed-end force in local 2-direction at Node J
	26 - 35	Fixed-end force in local 3-direction at Node J
	36 - 45	Fixed-end moment about local 1-direction at Node J
	46 - 55	Fixed-end moment about local 2-direction at Node J
	56 - 65	Fixed-end moment about local 3-direction at Node J

Note that values input are literally fixed-end values. Corrections due to hinges and rollers are performed within the program. Directions 1, 2 and 3 indicate principal directions in the local beam coordinates

F. Beam Data Cards (10I5,2I6,I8)

Columns	
1 - 5	Identification - beam number
6 - 10	Node I number
11 - 15	Node J number
16 - 20	Node K number - see Figure 4-1.
21 - 25	Material number
26 - 30	Geometric property number
31 - 35	A — Fixed-end force identification for element load cases A, B, C, and D respectively
36 - 40	B
41 - 45	C
46 - 50	D
51 - 56	End release code - Node I
57 - 62	End release code - Node J
63 - 70	Optional parameter k used for automatic generation of element data. This option is described below under a separate heading. If the option is not used, the field is left blank.

The end release code at each node is a six digit number of ones and/or zeros. The 1st, 2nd, ... 6th digits respectively correspond to the force components R1, R2, R3, M1, M2, M3 at each node.

If any one of the above member end forces is known to be zero (hinge or roller), the digit corresponding to that component is a one.

Automatic Element Data Generation

If a series of elements occurs in which each element number NE_i is one greater than the previous number NE_{i-1}

i.e., $$NE_i = NE_{i-1} + 1$$

only the element data card for the first element in the series need be given as input.

IF The end nodal point numbers $NI_i = NI_{i-1} + k$

$$NJ_i = NJ_{i-1} + k$$

AND THE
a) material identification number
b) geometric property identification number
c) fixed-end force identification numbers for each element load case
d) element code
e) orientation of local 2-axis

are the same for each element in the series.

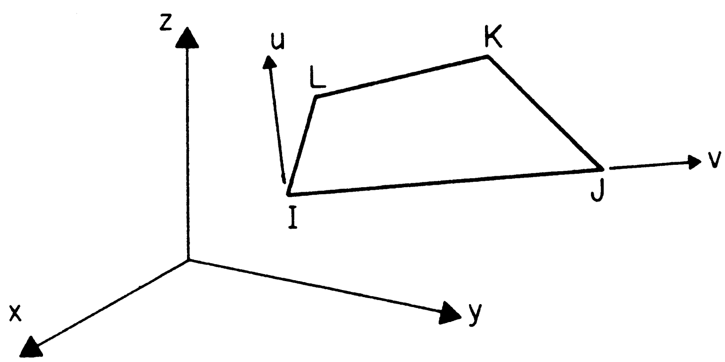
The generator option is the value of k and if left blank is taken to be one. The element data card for the last element in the structure must always be given.

*Where successive beam elements have the same stiffness, orientation and element loading, the program automatically skips recomputation of the stiffness. Note this when numbering the beams to obtain maximum efficiency.

Type 3 - Plane Stress Membrane Elements

Quadrilateral and triangular elements can be used for plane stress membrane elements of specified thickness which are oriented in an arbitrary plane. All elements have temperature dependent orthotropic material properties. Incompatible displacement modes can be included at the element level in order to improve the bending properties of the elements.

A general quadrilateral element is shown below:



A local element coordinate system is defined by a U-V system. The v-axis coincides with IJ side of the element. The u axis is normal to the v-axis and is in the plane defined by nodal points I, J and L. Node K must be in the same plane if the element stiffness calculations are to be correct. The following sequence of cards define the input data for a set of TYPE 3 elements.

A. Control Card (6I5)

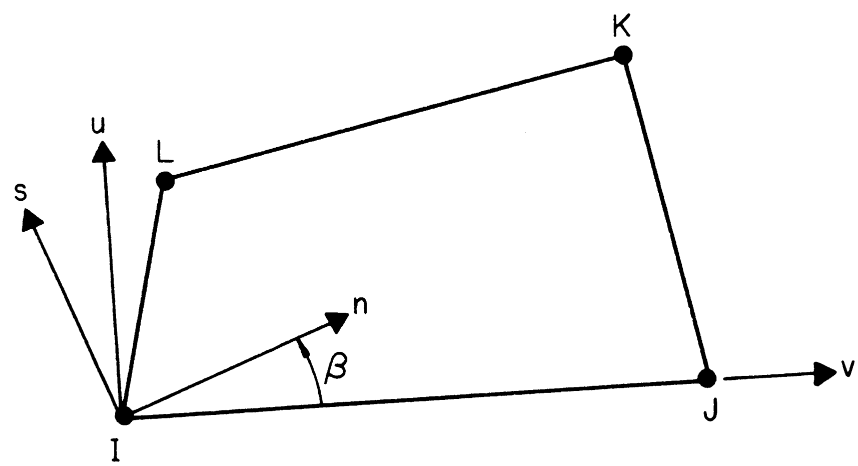
Columns	1 - 5	Number 3
	6 - 10	Number of elements
	11 - 15	Number of different materials
	16 - 20	Maximum number of temperature cards for any one material see Section B below.
	30	Non-zero numerical punch will suppress the introduction of incompatible displacement modes.

B. Material Property Information

Orthotropic, temperature dependent material properties are possible. For each different material the following group of cards must be supplied.

1. Material Identification Card (2I5,3F10.0)

Columns	1 - 5	Material identification number
	6 - 10	Number of different temperatures for which properties are given. If this field is left blank the number is taken as one.
	11 - 20	Weight density of material (for gravity loads only)
	21 - 30	Blank
	31 - 40	Angle β in degrees measured counter-clockwise from the v-axis to the n-axis.



The n-s axes are the principal axes for the orthotropic material. Weight and mass densities are listed only if gravity and inertia loads are to be considered.

2. Material Property Cards - Two cards for each temperature.

Card 1: (8F10.0)

Columns	1 - 10	Temperature
	11 - 20	Modulus of Elasticity - En
	21 - 30	Modulus of Elasticity - Es
	31 - 40	Modulus of Elasticity - Et
	41 - 50	Strain Ratio - ν_{ns}
	51 - 60	Strain Ratio - ν_{nt}
	61 - 70	Strain Ratio - ν_{st}
	71 - 80	Shear Modulus - Gns

Card 2: (3F10.0)

Columns	1 - 10	Coefficient of Thermal expansion - α_n
	11 - 20	Coefficient of Thermal expansion - α_s

For plane stress analysis the constitutive relation is modified by the program for normal stress $(\tau_{tt}) = 0.0$.

C. Element Load Factors

Four cards are used to define the element load cases A, B, C and D as fraction of the basic thermal, pressure and acceleration loads.

First card, load case A: Second card, load case B, etc.

Columns	1 - 10	Fraction of thermal load
	11 - 20	Fraction of pressure load
	21 - 30	Fraction of gravity in X-direction
	31 - 40	Fraction of gravity in Y-direction
	41 - 50	Fraction of gravity in Z-direction

D. Element Cards (6I5,2F10.0,2I5,F10.0)

One card per element must be supplied (or generated) with the following information:

Columns 1 - 5 Element number

6 - 10 Node I

11 - 15 Node J

16 - 20 Node K

21 - 25 Node L (Node L must equal Node K for triangular elements)

26 - 30 Material identification number

31 - 40 Reference temperature for zero stresses within element

41 - 50 Normal pressure on I-J side of element

51 - 55 Stress evaluation option "n."

56 - 60 Element data generator "k."

61 - 70 Element thickness

Element Data Generation - Element cards must be in element number sequence. If cards are omitted data for the omitted element data will be generated. The nodal numbers will be generated with respect to the first card in the series as follows:

$$I_n = I_{n-1} + k$$
$$J_n = J_{n-1} + k$$
$$K_n = K_{n-1} + k$$
$$L_n = L_{n-1} + k$$

All other element information will be set equal to the information on the last card. The mesh generation parameter "k" is specified on the last card.

Stress Print Option - See element type 4

Thermal Data - See element type 4

Type 4 - Two Dimensional Finite Elements

Quadrilateral and triangular elements can be used for the following purposes:

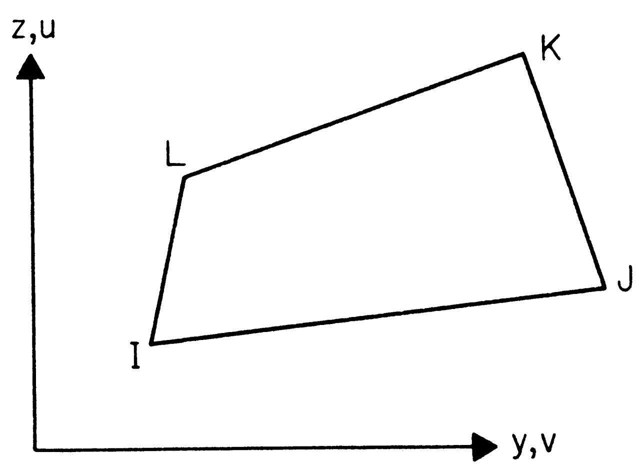
(i) Axisymmetric solid elements symmetrical about the Z-axis. The radial direction is specified as the Y-axis. Care must be exercised in combining this element with other types of elements.

(ii) Plane strain elements of unit thickness in the Y-Z plane.

(iii) Plane stress elements of specified thickness in the Y-Z plane.

All elements have temperature dependent orthotropic material properties. Incompatible displacement modes can be included at the element level in order to improve the bending properties of the element. However, for axisymmetric analysis incompatible modes should only be used for shell type structures where the radius to thickness ratio is large.

A general quadrilateral element is shown below:



A. Control Card (6I5)

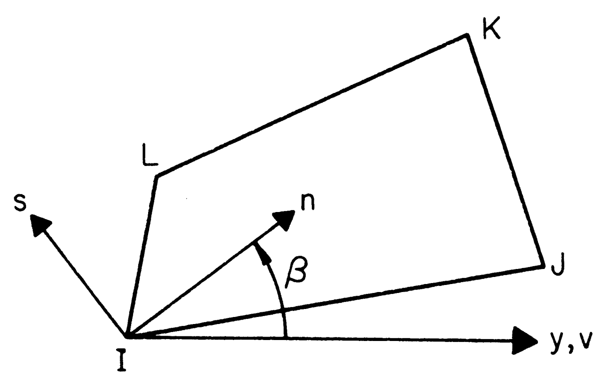
Columns	1 - 5	Number 4
	6 - 10	Number of elements
	11 - 15	Number of different materials
	16 - 20	Maximum number of temperature cards for any one material - see Section B below.
		0 axisymmetric analysis
	25	1 plane strain analysis
		2 plane stress analysis
	30	Non-zero numerical punch will suppress the introduction of incompatible displacement modes. Incompatible modes cannot be used for triangular elements and are automatically suppressed.

B. Material Property Information

Orthotropic, temperature dependent material properties are possible. For each different material the following group of cards must be supplied.

1. Material Identification Card (2I5,3F10.0)

Columns	1 - 5	Material identification number
	6 - 10	Number of different temperatures for which properties are given. If this field is left blank the number is taken as one.
	11 - 20	Weight density of material (for gravity loads only)
	21 - 30	Mass density of material (for rotational loads)
	31 - 40	Angle β in degrees measured counter-clockwise from the v-axis to the n-axis.



PRINCIPAL MATERIAL AXES

The n-s axes are the principal axes for the orthotropic material. Weight density is listed only if gravity and inertia loads are to be considered.

2. Material Property Cards - Two cards for each temperature.

Card 1: (8F10.0)

Columns	1 - 10	Temperature
	11 - 20	Modulus of elasticity - En
	21 - 30	Modulus of elasticity - Es
	31 - 40	Modulus of elasticity - E_t
	41 - 50	Strain ratio - ν_{ns}
	51 - 60	Strain ratio - ν_{nt}
	61 - 70	Strain ratio - ν_{st}
	71 - 80	Shear modulus - Gns

Card 2: (3F10.0)

Columns	1 - 10	Coefficient of thermal expansion - α_n
	11 - 20	Coefficient of thermal expansion - α_s
	21 - 30	Coefficient of thermal expansion - α_t

For plane stress analysis the constitutive relation is modified by the program for normal stress (τ_{tt}) = 0.0 .

C. Element Load Factors

Four cards are used to define the element load cases A, B, C and D as fraction of the basic thermal, pressure and acceleration loads.

First card, load case A: Second card, load case B; etc.

Columns	1 - 10	Fraction of thermal load
	11 - 20	Fraction of pressure load
	21 - 30	Angular velocity in radians/second (used to compute rotational loads only)
	31 - 40	Fraction of gravity in Y-direction (used for plane stress or plane strain only)
	41 - 50	Fraction of gravity in Z-direction

D. Element Cards (6I5,2F10.0,2I5,F10.0)

One card per element must be supplied (or generated) with the following information:

Columns	1 - 5	Element number
	6 - 10	Node I
	11 - 15	Node J
	16 - 20	Node K
	21 - 25	Node L (Node L must equal Node K for triangular elements)
	26 - 30	Material identification number
	31 - 40	Reference temperature for zero stresses withing element
	41 - 50	Normal pressure on I-J side of element
	51 - 55	Stress evaluation option "n"
	56 - 60	Element data generation "k"
	61 - 70	Element thickness (For plain strain set equal to 1.0 by program - For axisymmetric solids one radian is used)

Element Data Generation - Element cards must be in element number sequence. If cards are omitted the omitted element data will be generated. The nodal numbers will be generated with respect to the first card in the series as follows:

$$I_n = I_{n-1} + k$$

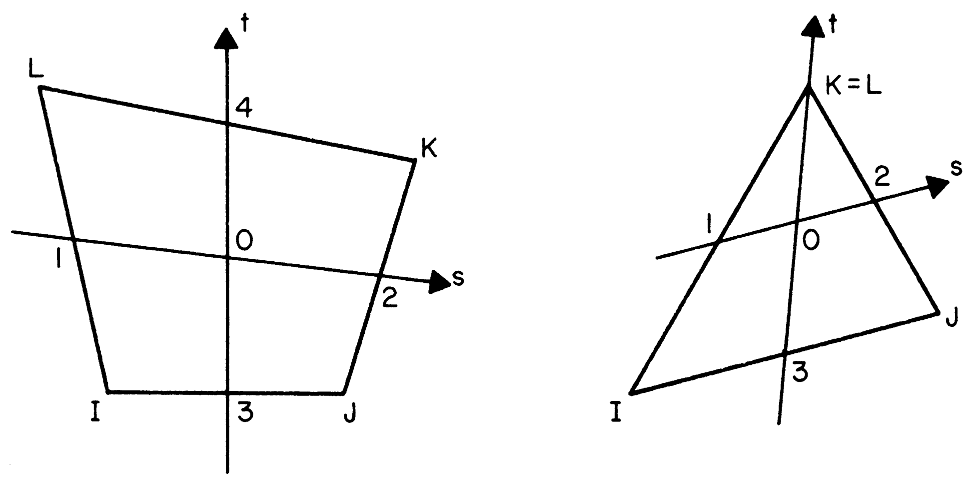
$$J_n = J_{n-1} + k$$

$$K_n = K_{n-1} + k$$

$$L_n = L_{n-1} + k$$

All other element information will be set equal to the information on the last card. The data generation k is given on the last card in the sequence.

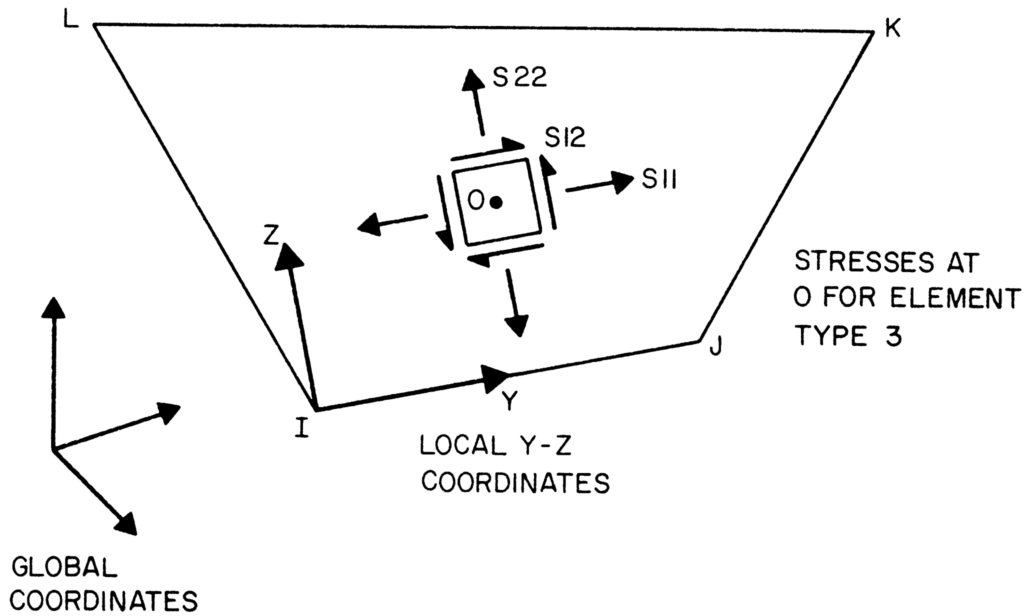
Stress Print Option - The following description of the stress print option applies to both element types 3 and 4. The value of the stress print option "n" can be given as 1, 0, 8, 16 or 20.

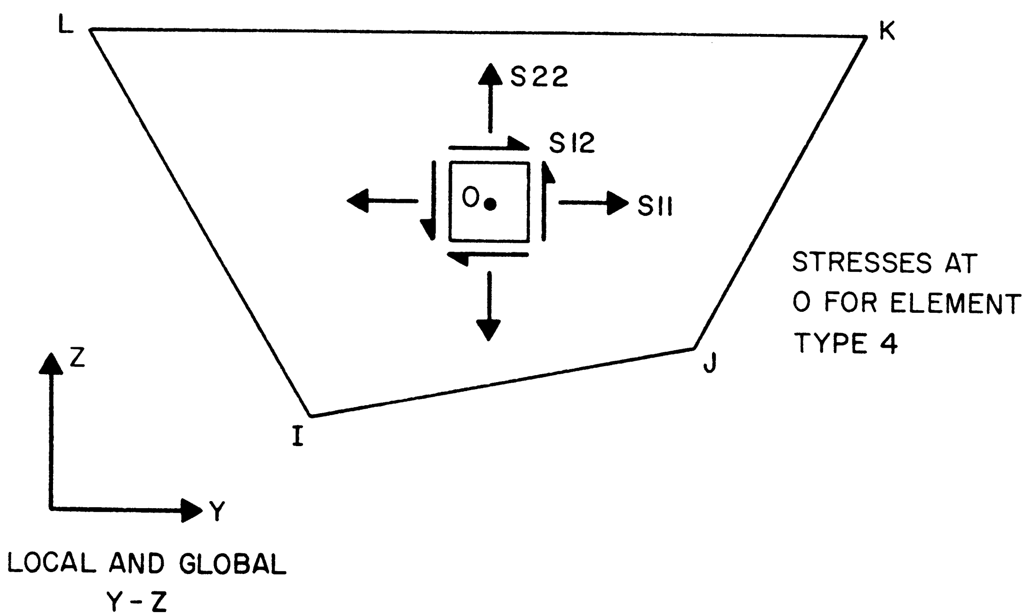


0 = origin of natural s-t coordinates (Figure 5-2). Points 1, 2, 3 and 4 are midpoints of sides. The points at which stresses are output depend on the value of n as described in the following table.

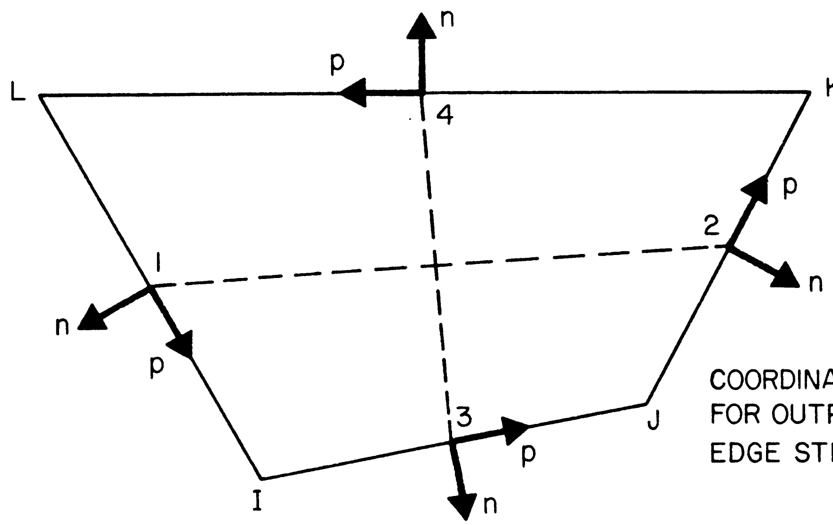
n	Stresses output at
1	None
0	0
8	0, 1
16	0, 1, 2, 3
20	0, 1, 2, 3, 4

The stresses at 0 are printed in a local Y-Z coordinate system. For element type 3, side I-J defines the local Y-Z axes in the plane of the element. For element type 4 the local Y-Z axes are parallel to the global Y-Z axes.

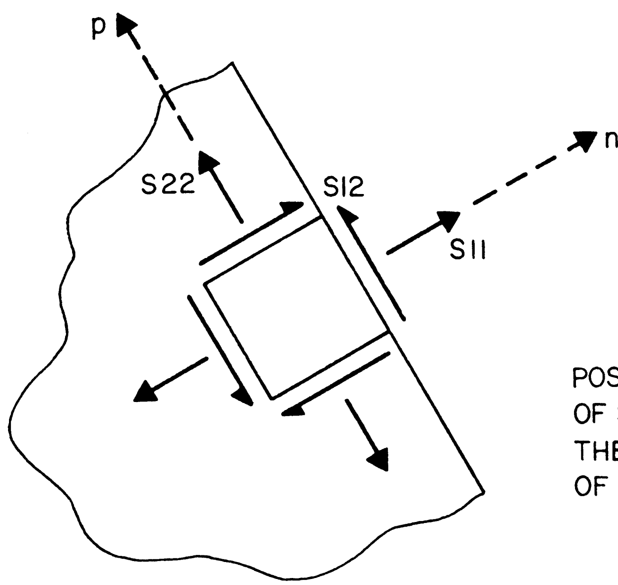




For both element types 3 and 4 the stresses at each midpoint are output in a rectangular n-p coordinate system defined by the outward normal to the edge (n axis) and the edge (p axis). The positive p axis for points 1, 2, 3 and 4 is from L to I, J to K, I to J and K to L respectively (counterclockwise positive about element).



COORDINATE SYSTEMS FOR OUTPUT OF EDGE STRESSES



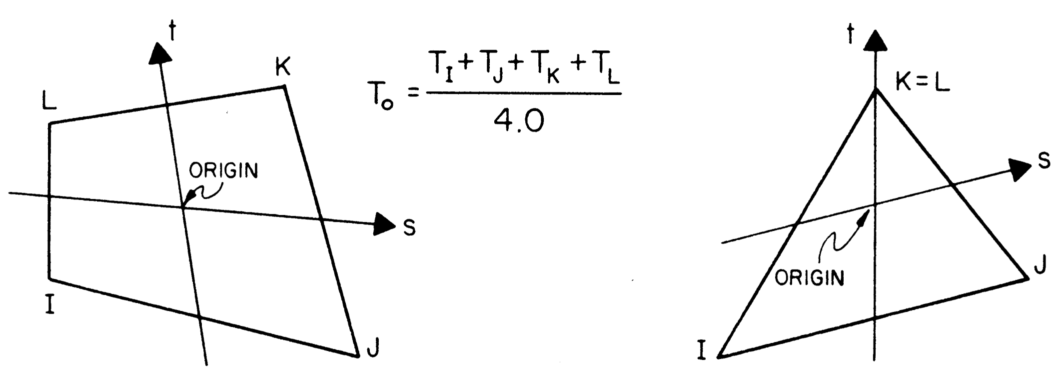
POSITIVE STATE OF STRESS AT THE MIDPOINT OF A SIDE

The stresses for an element are output under the following headings: S11, S22, S12, S33, S-MAX, S-MIN, ANGLE. The normal stresses S11 and S22 and the shear stress S12 are as described above. S-MAX and S-MIN are the principal stresses <u>in</u> the plane of the element and S33 is the third principal stress acting <u>on</u> the plane of the element. ANGLE is the angle in degrees from (1) the local Y axis at point 0, or (2) the n axis at the midpoints, to the axis of the algebraically largest principal stress.

For triangular elements the stress print option is as described above except that n=20 is not valid. If n=20 is input, n will be set to 16 by the program.

<u>Thermal Data</u> - Nodal temperatures as specified on the joint data cards are used by element types 3 and 4 in the following two ways:

(1) Temperature dependent material properties are approximated by interpolating (or extrapolating) the input material properties at the temperature T_0 corresponding to the origin of the local s-t coordinate system (see Figure 5.2 for description of local element coordinates). The material properties throughout the element are assumed constant corresponding to this temperature.



(2) For computation of nodal loads due to thermal strains in the element a bilinear interpolation expansion for the temperature change $\Delta T\ (s,t)$ is used.

$$\Delta T\ (s,t) = \sum_{i=1}^{4} h_i\ (s,t)\ T_i - T_r$$

where T_i are the nodal temperatures specified on the joint data cards, T_r is the reference stress free temperature and $h_i\ (s,t)$ are the interpolation functions given by equation 5.7.

Type 5 - Three Dimensional Solid Elements: 8 Nodal Brick

General three-dimensional, 8 node, isoparametric elements with three translational degrees of freedom per node are identified by the number 5. Isotropic material properties are assumed. The element load cases (A, B, C and D) are defined as a combination of surface pressure, hydrostatic loads, inertia loads in three directions and thermal loads. The six components of stress and three principal stresses are computed at the center of each element. Also, surface stresses are evaluated. Nine incompatible displacement modes are assumed in the formation of element stiffness matrices.

A. Control Card (4I5) --

Columns	1 - 5	The number 5
	6 - 10	Number of 8-node solid elements
	11 - 15	Number of different materials
	16 - 20	Number of element distributed load sets

B. Material Property Cards (I5,4F10) -- One card for each different material

Columns	1 - 5	Material identification number
	6 - 15	Modulus of elasticity (only elastic, isotropic materials are considered)
	16 - 25	Poisson's ratio
	26 - 35	Weight density of material
	36 - 45	Coefficient of thermal expansion

C. Distributed Surface Loads (2I5,2F10,2I5) -- One card is required for each unique set of uniformly distributed surface loads and for each reference fluid level for hydrostatically varying pressure loads. See notes IV and V for sign convention.

Columns	1 - 5	Load set identification number
	6 - 10	LT (load type) LT = 1 if this card specifies a uniformly distributed load. LT = 2 if this card specifies a hydrostatically varying pressure.
	11 - 20	P If LT = 1, P is the magnitude of the uniformly distributed load If LT = 2, P is the weight density of the fluid causing the hydrostatic pressure
	21 - 30	Y If LT = 1, leave blank If LT = 2, Y is the global Y coordinate of the surface of fluid causing hydrostatic <u>pressure</u> loading
	31 - 35	Element face number on which surface load acts. Face numbers are from 1 to 6 as described in note IV for uniformly distributed loads and can be only faces 2, 4 or 6 for hydrostatically varying <u>pressures</u>.

D. <u>One Blank Card</u>

E. <u>Element Load Case Multipliers</u> (5 cards of 4F10) -- Multipliers on the element load cases are scaling factors in order to provide flexibility in modifying applied loads.

Card 1: Columns	1 - 10	PA	Pressure load multipliers
	11 - 20	PB	
	21 - 30	PC	
	31 - 40	PD	

PA is a factor used to scale the complete set of distributed surface loads. This scaled set of loads is assigned to element load case A. Note that zero is a valid multiplier. PB, PC and PD are similar to PA except that scaled loads are assigned to element load cases B, C and D respectively. For the majority of applications these factors should be 1.0

Card 2: Columns	1 - 10	TA	Thermal load multipliers
	11 - 20	TB	
	21 - 30	TC	
	31 - 40	TD	

TA is a factor used to scale the complete set of thermal loads. The scaled set of loads are then assigned to element load case A. TB, TC and TD are similar and refer to element load cases B, C and D respectively.

Card 3: Columns	1 - 10	GXA	Gravity load multipliers for + X global direction
	11 - 20	GXB	
	21 - 30	GXC	
	31 - 40	GXD	
Card 4: Columns	1 - 10	GYA	Gravity load multipliers for + Y global direction
	11 - 20	GYB	
	21 - 30	GYC	
	31 - 40	GYD	
Card 5: Columns	1 - 10	GZA	Gravity load multipliers for + Z global direction
	11 - 20	GZB	
	21 - 30	GZC	
	31 - 40	GZD	

Gravity loads are computed from the weight density of the material and from the geometry of the element. GXA is a multiplier which reflects the location of the gravity axis and any load factors used. The program computes the weight of the element, multiplies it by GXA and assigns the resulting loads to the + X direction of element load case A. Consequently GXA is the product of the component of gravity along the + X global axis (from - 1.0 to 1.0) and any desired load factor. GXB, GXC and GXD are similar to GXA and refer to element load cases B, C and D respectively. GYA and GZA refer to the global Y and Z directions respectively.

F. Element Cards (12I5,4I2,2I1,F10)

Columns	Description	Node
1 - 5	Element number	
6 - 10	Global node point numbers corresponding to element nodes (See note III)	1
11 - 15		2
16 - 20		3
21 - 25		4
26 - 30		5
31 - 35		6
36 - 40		7
41 - 45		8
46 - 50	Integration Order	
51 - 55	Material Number	
56 - 60	Generation Parameter (INC)	
61 - 62	LSA	LSA is the distributed surface load set identification number of the distributed load acting on this element to be assigned to element load case A. LSB, LSC and LSD refer to element load cases B, C and D respectively
63 - 64	LSB	
65 - 66	LSC	
67 - 68	LSD	
69 - 70	Face numbers for stress output	
71 - 80	Stress free element temperature	

G. Notes

I. Element Generation

1. Element cards must be in ascending order
2. Generation is possible as follows:

   If a series of element cards are omitted

   a. Nodal point numbers are generated by adding INC to those of preceding element. (If omitted, INC is set equal to 1.)

   b. Same material properties are used as for the preceding element.

   c. Same temperature is used for succeeding elements.

   d. If on first card for the series the integration order is:

      >0 Same value is used for succeeding elements

      $=0$ A new element stiffness is not formed. Element stiffness is assumed to be identical to that of the preceding element.

      <0 Absolute value is used for the first element of the series, and the same element stiffness is used for succeeding elements.

   e. If on first card for the series, the distributed load number (for any load case) is:

      >0 Same load is applied to succeeding elements

      <0 The load case is applied to this element but not to succeeding elements in the series.

3. Element card for the last element must be supplied.

II. Integration Order

Computation time (for element stiffness) increases with the cube of the integration order. Therefore, the smallest satisfactory order should be used. This is found to be:

2 for rectangular element

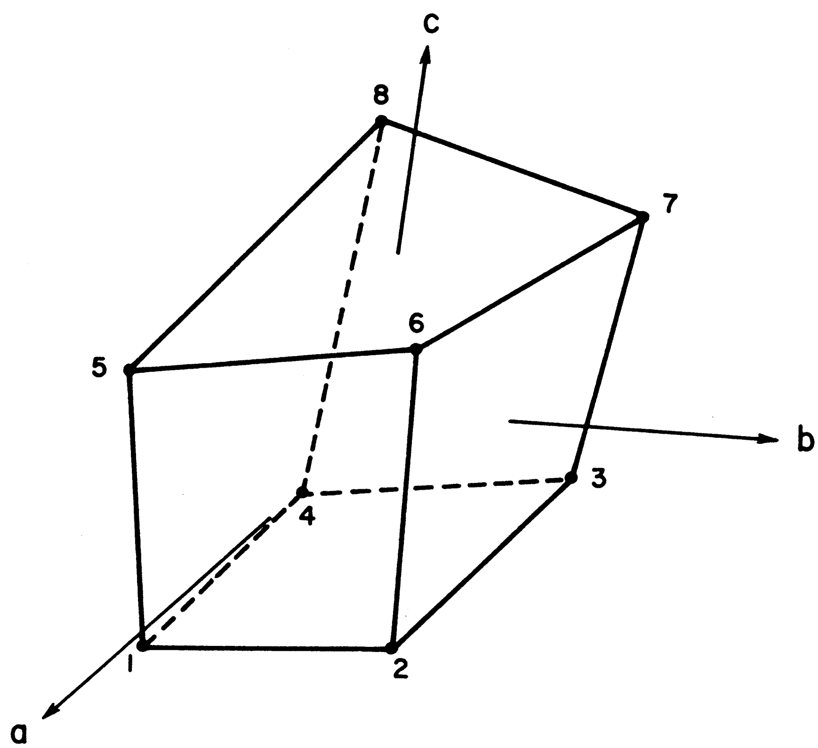
3 for skewed element

4 may be used if element is extremely distorted in shape, but not recommended.

Mesh should be selected to give "regular" elements as far as possible.

III. Element Coordinate System

Local element coordinate system is a natural system for this element in which the element maps onto a cube. Local element numbering is shown in Figure 1.



IV. Identification of Element Faces

Element faces are numbered as follows:

Face 1 corresponds to + a direction
2 corresponds to - a direction
3 corresponds to + b direction
4 corresponds to - b direction
5 corresponds to + c direction
6 corresponds to - c direction
0 corresponds to the center of the element

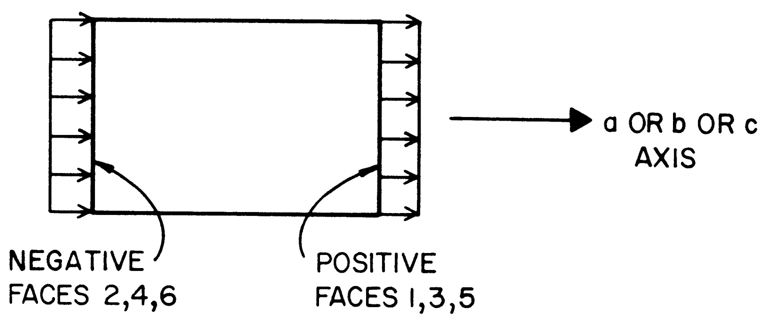
Faces 1,3,5 are positive faces

Faces 2,4,6 are negative faces

V. Distributed Surface Loads

Two types of surface loadings may be specified; load type 1 (LT = 1), uniformly distributed surface load and load type 2 (LT = 2), hydrostatically varying surface pressure (but not surface tension). Both loading types are for loads normal to the surface and do not include surface shears. Surface loadings that do not fall into these categories must be input as consistent nodal loads on the concentrated load data cards (see Section B4).

(1) LT = 1: A positive surface load acts in the direction of the outward normal of a positive element face and along the inward normal of a negative element face as shown in the following diagram.



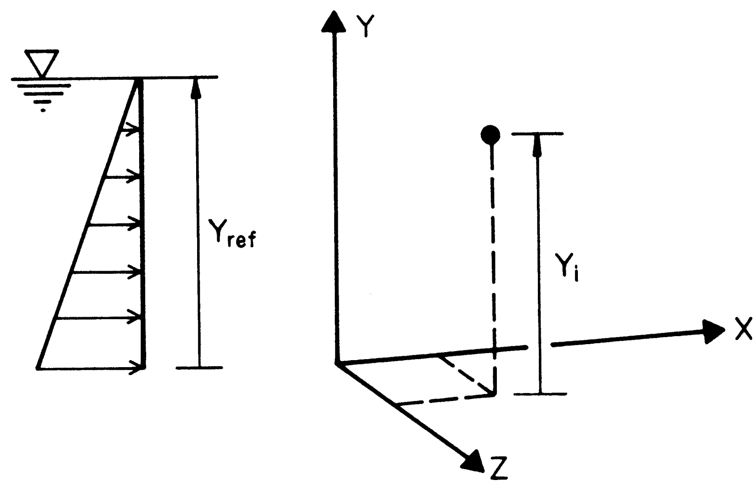
POSITIVE SURFACE LOADING P

If the uniformly distributed surface loading P is input as a positive quantity then it describes pressure loading on faces 2, 4 or 6 and tensile loading on faces 1, 3 or 5. If P is input as a negative quantity then it describes tensile loading on faces 2, 4 or 6 and pressure on faces 1, 3 or 5.

(2) LT = 2: A hydrostatically varying surface pressure on element faces 2, 4 or 6 can be specified by a reference fluid surface and a fluid weight density γ as input. Only one hydrostatic surface pressure card need be input in order to specify a hydrostatic loading on the complete structure. The consistent nodal loads are calculated by the program as follows. At each numerical integration point "i" on an element surface the pressure P_i is calculated from

$$P_i = \gamma \; (Y_i - Y_{ref})$$

where Y_i is the global Y coordinate of the point in question and Y_{ref} specifies the fluid surface assuming gravity acts along the -Y axis



If $P_i > 0$, corresponding to surface tension the contribution is ignored. If an element face is such that $Y_i > Y_{ref}$ for all i (16 integration points are used by program) then no nodal loads will be applied to the element. If some $P_i > 0$ and some $P_i < 0$ for a particular face then approximate nodal loads are obtained for the partially loaded surface.

VI. Thermal Loads

Thermal loads are computed assuming a constant temperature increase ΔT throughout the element.

$$\Delta T = T_{avg} - T_o$$

T_{avg} = the average of the 8 nodal point temperatures specified on joint data cards

T_o = stress free element temperature specified on the element card.

VII. Element Load Cases

Element load case A consists of all the contributions from distributed loadings, thermal loadings and gravity loading for all the elements taken collectively.

$$\text{Load case A} = \Sigma \begin{aligned} & \text{PA} \times \text{pressure loading} \\ & + \text{TA} \times \text{thermal loading} \\ & + \text{GXA} \times \text{gravity X loading} \\ & + \text{GYA} \times \text{gravity Y loading} \\ & + \text{GZA} \times \text{gravity Z loading} \end{aligned}$$

Element load case A for the set of three dimensional solid elements is added to element load case A for the other element types in the analysis. Element load cases B, C and D are analogous to element load case A. The loading cases for the structure are obtained by adding linear combinations of element load cases A, B, C and D to the nodal loads specified on the joint data cards.

VIII. Element Stresses are Output as Follows

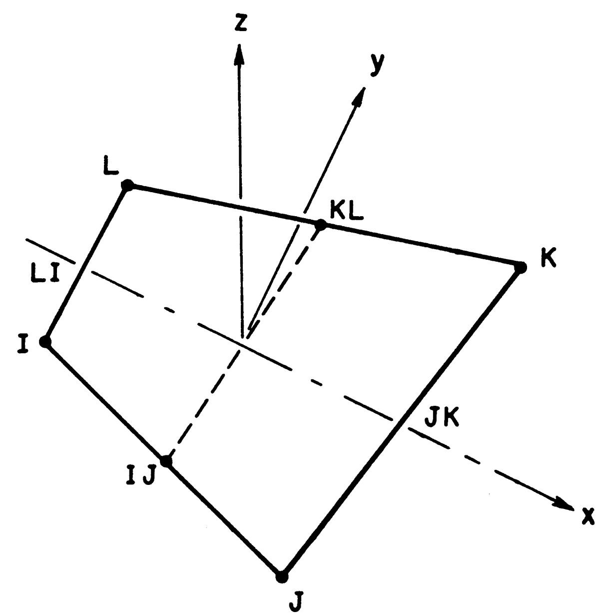
1. At the centroid of the element stresses are referred to the global axes. Three principal stresses are also presented.

2. At the center of an element face stresses are referred to a set of local axes (x, y, z). These local axes are individually defined for each face as follows: Let nodal points I, J, K and L be the four corners of the element face. Then

   x Specified by LI - JK, where LI and JK are midpoints of sides L-I and J-K

   z Normal to x and to the line joining midpoints IJ and KL.

   y Normal to x and z to complete the right handed system.



The corresponding nodal points I, J, K and L in each face are given in the table.

FACE	NODAL POINTS			
	I	J	K	L
1	1	2	6	5
2	4	3	7	8
3	3	7	6	2
4	4	8	5	1
5	8	5	6	7
6	4	1	2	3

Two surface principal stresses and the angle between the algebraically largest principal stress and the local x axis are printed with the output. It is optional to choose one or two locations of an element where stresses are to be computed. In the output, face zero designates the centroid of the element.

Type 6 - Plate and Shell Elements (Quadrilateral)

A. Control Card (3I5)

Columns	1 - 5	The number 6
	6 - 10	Number of shell elements
	11 - 15	Number of different materials

B. Material Property Information

Anisotropic material properties are possible. For each different material, two cards must be supplied.

Card 1: (I10,20X,4F10.0)

Columns	1 - 10	Material identification number
	31 - 40	Mass density
	41 - 50	Thermal expansion coefficient α_x
	51 - 60	Thermal expansion coefficient α_y
	61 - 70	Thermal expansion coefficient α_{xy}

Card 2: (6F10.0)

Columns	1 - 10	Elasticity element C_{xx}
	11 - 20	Elasticity element C_{xy}
	21 - 30	Elasticity element C_{xs}
	31 - 40	Elasticity element C_{yy}
	41 - 50	Elasticity element C_{ys}
	51 - 60	Elasticity element G_{xy}

Elements in plane stress material matrix [C]

$$\begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{xy} \end{Bmatrix} = \begin{bmatrix} C_{xx} & C_{xy} & C_{xs} \\ C_{xy} & C_{yy} & C_{ys} \\ C_{xs} & C_{ys} & G_{xy} \end{bmatrix} \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{Bmatrix}$$

C. Element Load Multiplers (5 cards)

Card 1: (4F10.0)

Columns	1 - 10	Distributed lateral load multiplier for load case A
	11 - 20	Distributed lateral load multiplier for load case B
	21 - 30	Distributed lateral load multiplier for load case C
	31 - 40	Distributed lateral load multiplier for load case D

Card 2: (4F10.0)

Columns	1 - 10	Temperature multiplier for load case A
	11 - 20	Temperature multiplier for load case B
	21 - 30	Temperature multiplier for load case C
	31 - 40	Temperature multiplier for load case D

Card 3: (4F10.0)

Columns	1 - 10	X-direction acceleration for load case A
	11 - 20	X-direction acceleration for load case B
	21 - 30	X-direction acceleration for load case C
	31 - 40	X-direction acceleration for load case D

Card 4: (4F10.0) Same as Card 3 for Y-direction

Card 5: (4F10.0) Same as Card 3 for Z-direction

D. <u>Element Cards (8I5,F10.0)</u>

One card for each element

Columns	1 - 5	Element number
	6 - 10	Node I
	11 - 15	Node J
	16 - 20	Node K
	21 - 25	Node L
	26 - 30	Node O*
	31 - 35	Material identification (If left blank, taken as one)
	36 - 40	Element data generator K_n**
	41 - 50	Element thickness
	51 - 60	Distributed lateral load (pressure)
	61 - 70	Mean temperature variation T from the reference level in undeformed position
	71 - 80	Mean temperature gradient $\partial T/\partial z$ across the shell thickness (a positive temperature gradient produces a negative curvature).

* When columns 26 - 30 are left blank, mid-node properties are computed by averaging the four nodes.

** Element cards must be in element number sequence. If element cards are omitted, the program automatically generates the omitted information as follows:

The increment for element number is one

$$\text{i.e.} \quad NE_{i+1} = NE_i + 1$$

The corresponding increment for nodal number is K_n

$$\text{i.e.} \quad NI_{i+1} = NI_i + K_n$$

$$NJ_{i+1} = NJ_i + K_n$$

$$NK_{i+1} = NK_i + K_n$$

$$NL_{i+1} = NL_i + K_n$$

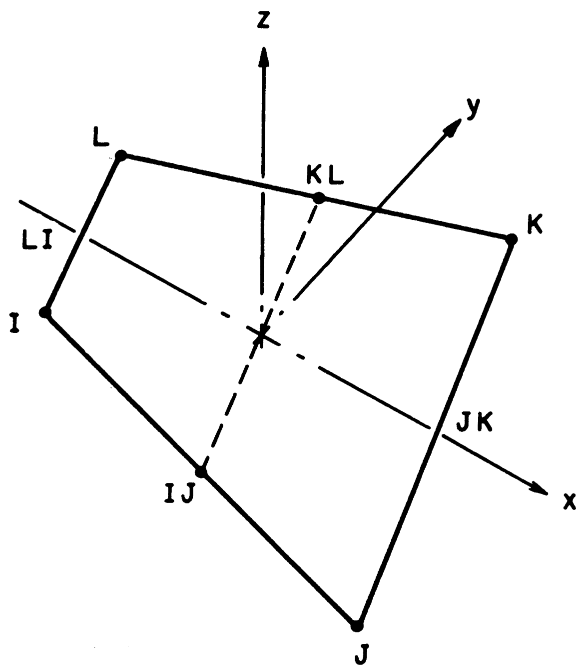
Material identification, element thickness, distributed lateral load, temperature and temperature gradient for generated elements are the same as the first element in the series. The last element card is always needed.

NOTE

The nodal point numbers I, J, K and L are in sequence in a counter-clockwise direction around the element. The local element coordinate system (x, y, z) is defined as follows:

- x Specified by LI - JK, where LI and JK are midpoints of sides L-I and J-K.
- z Normal to x and to the line joining midpoints IJ and KL.
- y Normal to x and z to complete the right-handed system.

This system is used to express all physical and kinematic shell properties (stresses, strains, material law, etc.), except that the body force density is referred to the global coordinate system (X, Y, Z).



For the analyses of smooth shells, rotational constraints normal to the surface may be imposed by the addition of Boundary elements at the nodes (element type #7).

Type 7 - Boundary Element

This element is used to constrain nodal displacements to specified values, to compute support reactions and to provide linear elastic supports to nodes. If the boundary condition code for a particular degree of freedom is specified as 1 on the joint data cards (section B-4) the displacement corresponding to that degree of freedom is zero and no support reactions are obtained with the print out. Alternatively, a boundary element can be used to accomplish the same effect except that support reactions are obtained since they are equal to the member end forces of the boundary elements which are printed. In addition the boundary element can be used to specify non-zero nodal displacements in any direction which is not possible using the joint data cards.

The boundary element is defined by a single directed axis through a specified nodal point, by a linear extensional stiffness along the axis and by a linear rotational stiffness about the axis. The boundary element is essentially a spring which can have axial displacement stiffness and axial rotational stiffness. There is no limit to the number of boundary elements which can be applied to any joint to produce the desired effects. Boundary elements have no effect on the size of the stiffness matrix.

INPUT DATA

A. Control Card (2I5)

Columns	1 - 5	The number 7.
	6 - 10	Total number of boundary elements.

B. Element Load Multipliers (4F10.0)

Columns	1 - 10	Multiplier for load case A
	11 - 20	Multiplier for load case B
	21 - 30	Multiplier for load case C
	31 - 40	Multiplier for load case D

C. Element Cards (8I5,3F10.0)

One card per element (in ascending nodal point order) except where automatic element generation is used.

Columns	1 - 5	Node N, at which the element is placed	
	6 - 10	Node I	Leave columns 11 - 25 blank if only node I is needed.
	11 - 15	Node J	
	16 - 20	Node K	
	21 - 25	Node L	
	26 - 30	Code for displacement	
	31 - 35	Code for rotation	
	36 - 40	Data generator Kn.	
	41 - 50	Specified displacement along element axis	
	51 - 60	Specified rotation about element axis	
	61 - 70	Spring stiffness (set to 10^{10} if left blank) for both extension and rotation.	

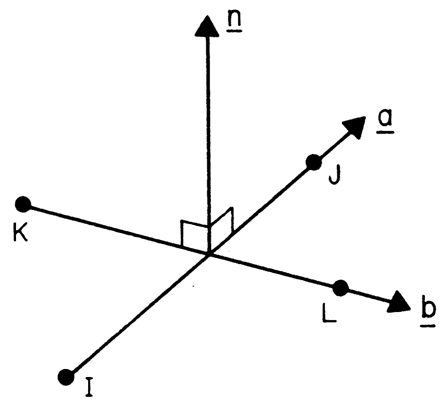
EXPLANATION OF INPUT DATA

(1) Direction of boundary element

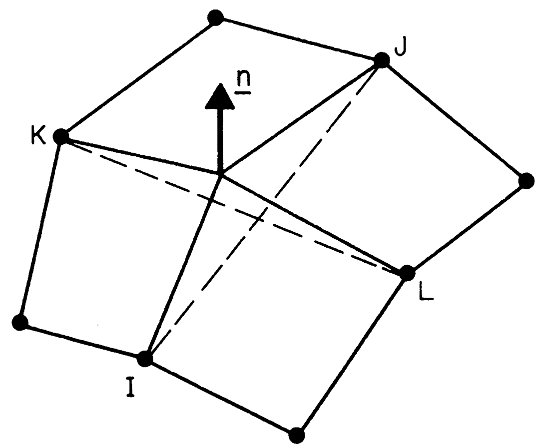
The direction of the boundary element at node N is specified in one of two ways.

(i) A second nodal point I defines the positive direction of the element from node I to node N.

(ii) Four nodal points I, J, K and L specify the positive direction of the element as the normal to the plane defined by two intersecting straight lines (vectors $\underline{a}$ and $\underline{b}$)



$\underline{n} = \underline{a} \times \underline{b}$



ROTATIONAL CONSTRAINT
IN THIN SHELL ANALYSIS

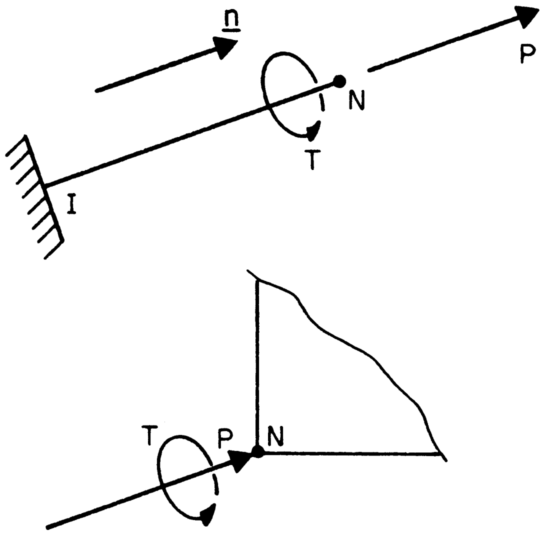
The four points I, J, K and L need not be unique. A useful application for the analysis of shallow thin shells employs the boundary element to approximate rotational constraint about the surface normal as shown above.

$\underline{n}$ is given by the vector cross product $\underline{n} = \underline{a} \times \underline{b}$

The positive direction of the boundary element corresponds to the direction of $\underline{n}$. The point of application is independent of $\underline{n}$

Note that node I in case (i) and nodes I, J, K and L in case (ii) are used only to define the direction of the element and if convenient may be any nodes used to define other elements. However 'artificial nodes' may be created to define directions of boundary elements. These 'artificial nodes' are input on the joint data cards (section B-4) with their coordinates and with all the boundary condition codes specified as 1 (one).

The positive direction of boundary elements is needed to interpret the direction of forces in the element and hence of reactions.



Positive force P and torque T acting on boundary element

P is +ve in direction of $\underline{n}$
T is +ve by right hand rule about $\underline{n}$

Positive reactions at node N

(2) Displacement and rotation codes

These codes are either a 0 (zero) or a 1 (one) punched in columns 30 and 35.

If displacement code = 0: The boundary element is not applied to the displacement degrees of freedom at node N.

If displacement code = 1: When this code is used the displacement δ specified in columns 41 - 50 and the spring stiffness k specified in columns 61 - 70 are used by the program in the following way. The load P evaluated from $P = k\,\delta$ is applied to node N in the positive direction of the element if δ is positive. If k is much greater then the stiffness of the structure at node N without the boundary element then the net affect is to produce a displacement very nearly equal to δ at node N. If $\delta = 0$ then $P = 0$ and the stiff spring approximates a rigid support. Note that the load P will contribute to the support reaction for nonzero δ. The boundary condition codes specified on the joint data cards (section B-4) must be consistent with the fact that a load P is being applied to node N to effect the desired displacement (even when this displacement is zero)

If rotation code = 0: The boundary element is not applied to the rotational degrees of freedom at node N.

NOTE: if both displacement and rotational codes are set equal to 0 the boundary element is ignored.

If rotation code = 1: This case is completely analogous to the situation described above where the displacement code is 1. A torque T evaluated from $T = k\,\theta$ is applied to node N about the axis of the element. The rotation θ is specified in columns 51 - 60.

(3) Data generator Kn

When a series of nodes are such that:

(i) All have identical boundary elements attached

(ii) All boundary elements have same direction

(iii) All specified displacements and rotations are identical

(iv) The nodal sequence forms an arithmetic sequence, i.e. N, $N + K_n$, $N + 2\,K_n$ etc. then only the first and last node in the sequence need be input. The increment K_n is input in columns 36 - 40 of the first card.

(4) Element load multipliers

Each of the four possible element load cases A, B, C and D associated with the boundary elements consists of the complete set of displacements as specified on the boundary element cards factored by the element load multiplier for the corresponding load case. As an example suppose that displacement of node N is specified as 1.0, spring stiffness as 10^{10} and no other boundary element displacements are specified. Let case A multiplier be 0.0 and case B multiplier be 2.0. For element load case A the specified displacement is 0.0 x 1.0 = 0.0 while that for B is 2.0 x 1.0 = 2.0. Linear combinations of element load cases A, B, C and D for all types of elements collectively for a particular problem are specified on the joint data input cards (section B-4). As far as the boundary element is concerned this device is useful when a particular node has a support displacement in one load case but is fixed in others.

Type 8 - Three-Dimensional Thick Shell Element (16 Nodes)

Three-dimensional 16 node, curved isoparametric elements with three translational degrees of freedom per node are identified by the number 8. Isotropic material properties are assumed. The element load cases (A, B, C and D) are defined as a combination of surface pressure, hydrostatic loads, inertia loads in three directions and thermal loads. The six components of stress and three principal stresses are computed at the center of each element. Also, surface stresses are evaluated. Five incompatible displacement modes are assumed in the formation of element stiffness matrices in addition to the sixteen isoparametric interpolation functions.

A. Control Card (4I5) --

Columns	1 - 5	The number 8
	6 - 10	Number of thick shell elements
	11 - 15	Number of different materials
	16 - 20	Number of element distributed load sets

B. Material Property Cards (I5,4F10) -- One card for each different material

Columns	1 - 5	Material identification number
	6 - 15	Modulus of elasticity (only elastic, isotropic materials are considered)
	16 - 25	Poisson's ratio
	26 - 35	Weight density of material
	36 - 45	Coefficient of thermal expansion

C. Distributed Surface Loads (2I5,2F10,2I5) -- One card is required for each unique set of uniformly distributed surface loads and for each reference fluid level for hydrostatically varying pressure loads. See notes III and V for sign convention.

Columns 1 - 5 Load set identification number

6 - 10 LT (load type)

LT = 1 if this card specifies a uniformly distributed load.

LT = 2 if this card specifies a hydrostatically varying pressure.

11 - 20 P

If LT = 1, P is the magnitude of the uniformly distributed load

If LT = 2, P is the weight density of the fluid causing the hydrostatic pressure

21 - 30 Y

If LT = 1, leave blank

If LT = 2, Y is the global Y coordinate of the surface of fluid causing hydrostatic pressure loading

31 - 35 Element face number of which surface load acts. Face numbers are from 1 to 6 as described in note IV for uniformly distributed loads and can be only faces 2, 4 or 6 for hydrostatically varying pressures.

D. Reference Temperature (1F10) --

Columns 1 - 10 Stress free temperature

E. Element Load Case Multipliers (5 cards of 4F10) -- Multipliers on the element load case are scaling factors in order to provide flexibility in modifying applied loads.

Card 1:	Columns	1 - 10	PA	Pressure load multipliers
		11 - 20	PB	
		21 - 30	PC	
		31 - 40	PD	

PA is a factor used to scale the complete set of distributed surface loads. This scaled set of loads is assigned to element load case A. Note that zero is a valid multiplier. PB, PC and PD are similar to PA except that scaled loads are assigned to element load cases B, C and D respectively. For the majority of applications these factors should be 1.0

Card 2: Columns

1 - 10	TA	Thermal load multipliers
11 - 20	TB	
21 - 30	TC	
31 - 40	TD	

TA is a factor used to scale the complete set of thermal loads. The scaled set of loads are then assigned to element load case A. TB, TC and TD are similar and refer to element load cases B, C and D respectively.

Card 3: Columns

1 - 10	GXA	Gravity load multipliers for + X global direction
11 - 20	GXB	
21 - 30	GXC	
31 - 40	GXD	

Card 4: Columns

1 - 10	GYA	Gravity load multipliers for + Y global direction
11 - 20	GYB	
21 - 30	GYC	
31 - 40	GYD	

Card 5: Columns

1 - 10	GZA	Gravity load multipliers for + Z global direction
11 - 20	GZB	
21 - 30	GZC	
31 - 40	GZD	

Gravity loads are computed from the weight density of the material and from the geometry of the element. GXA is a multiplier which reflects the location of the gravity axis and any load factors used. The program computes the weight of the element, multiplies it by GXA and assigns the resulting loads to the + X direction of element load case A. Consequently GXA is the product of the component of gravity along the +X global axis (from - 1.0 to 1.0) and any desired load factor. GXB, GXC and GXD are similar to GXA and refer to element load cases B, C and D respectively. GYA and GZA refer to the global Y and Z directions respectively.

F. Element Cards (4I5,4I2,2X,F10/16I5) Two Cards for each element are needed.

Card 1: Columns

Columns		
1 - 5	Element number	
6 - 10	Integration order	
11 - 15	Material number	
16 - 20	Generation parameter (INC)	
21 - 22	LSA	LSA is the distributed surface load set identification number of the distributed load acting on this element to be assigned to element load case A. LSB, LSC and LSD refer to element lcad cases B, C and D respectively
23 - 24	LSB	
25 - 26	LSC	
27 - 28	LSD	
31 - 40	Element temperature	

Card 2: Columns

Columns	Description	Node
1 - 5	Global nodal point numbers corresponding to element nodes (See note II)	1
6 - 10		2
11 - 15		3
16 - 20		4
21 - 25		5
26 - 30		6
31 - 35		7
36 - 40		8
41 - 45		9
46 - 50		10
51 - 55		11
56 - 60		12
61 - 65		13
66 - 70		14
71 - 75		15
76 - 80		16

G. Notes

I. Element Generation

1. Element cards must be in ascending order.

2. Generation is possible as follows:

   If a series of element cards are omitted

   a. Nodal point numbers are generated by adding INC to those of preceding element. (If omitted, INC is set equal to 1).

   b. Same material properties are used as for the preceding element.

   c. Same temperature is used for succeeding elements.

   d. If on first card for the series the integration order is:

      > 0 Same value is used for succeeding elements

      = 0 A new element stiffness is not formed. Element stiffness is assumed to be identical to that of the preceding element.

      < 0 Absolute value is used for the first element of the series, and the same element stiffness is used for succeeding elements.

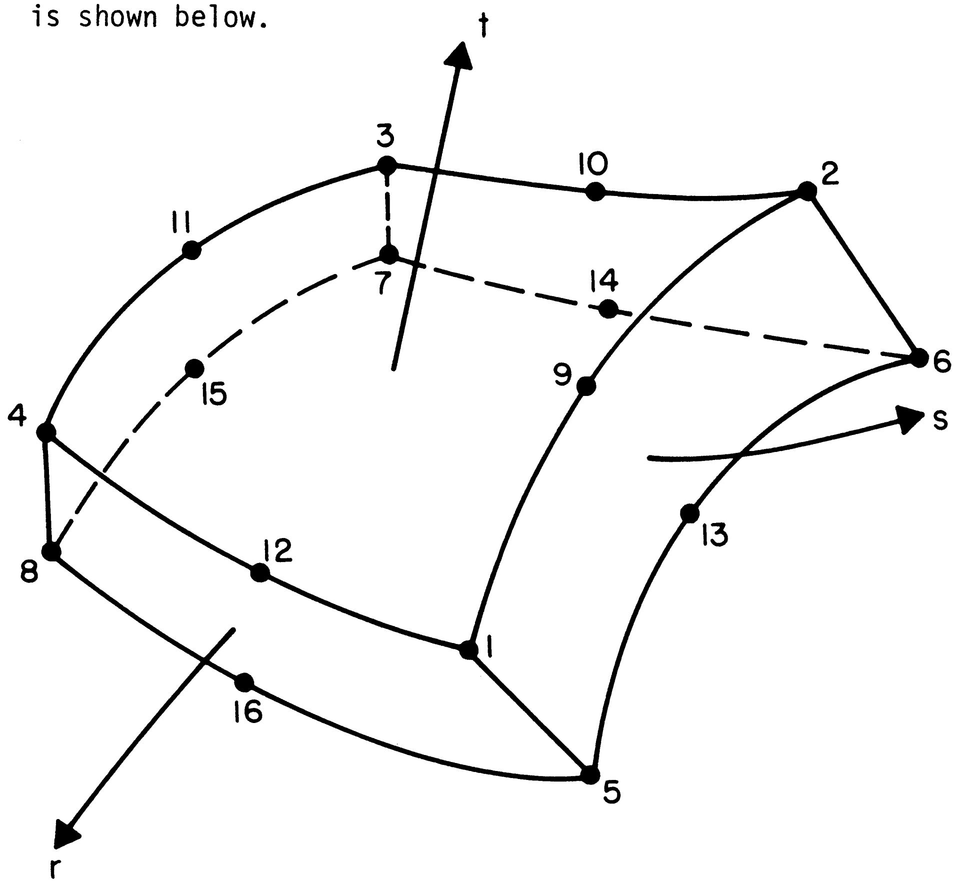
   e. If on first card for the series, the distributed load number (for any load case) is:

      > 0 Same load is applied to succeeding elements

      < 0 The load case is applied to this element but not to succeeding elements in the series.

3. Element Card for the last element must be supplied.

II. Local element coordinate system is a natural system for this element in which the element maps onto a cube. Local element numbering is shown below.



III. <u>Element Face are Numbered as Follows:</u>

Face 1 corresponds to + r direction
Face 2 corresponds to - r direction
Face 3 corresponds to + s direction
Face 4 corresponds to - s direction
Face 5 corresponds to + t direction
Face 6 corresponds to - t direction

1, 3, 5 are positive faces
2, 4, 6 are negative faces

IV. <u>Integration Order</u>

Computation time (for element stiffness) increases with the cube of the integration order. Therefore, the smallest satisfactory order should be used. This is found to be:

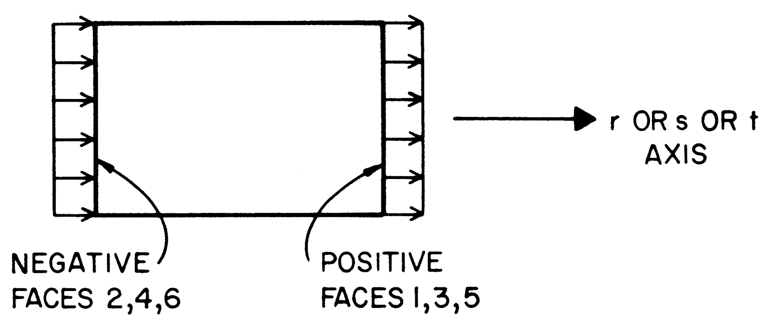
3 for regular shape

4 for irregular shape

V. Distributed Surface Loads

Two types of surface loadings may be specified; load type 1 (LT = 1), uniformly distributed surface load and load type 2 (LT = 2), hydrostatically varying surface pressure (but not surface tension). Both loading types are for loads normal to the surface and do not include surface shears. Surface loadings that do not fall into these categories must be input as consistent nodal loads on the concentrated load data cards (see Section B4).

(1) LT = 1: A positive surface load acts in the direction of the outward normal of a positive element face and along the inward normal of a negative element face as shown in the following diagram.



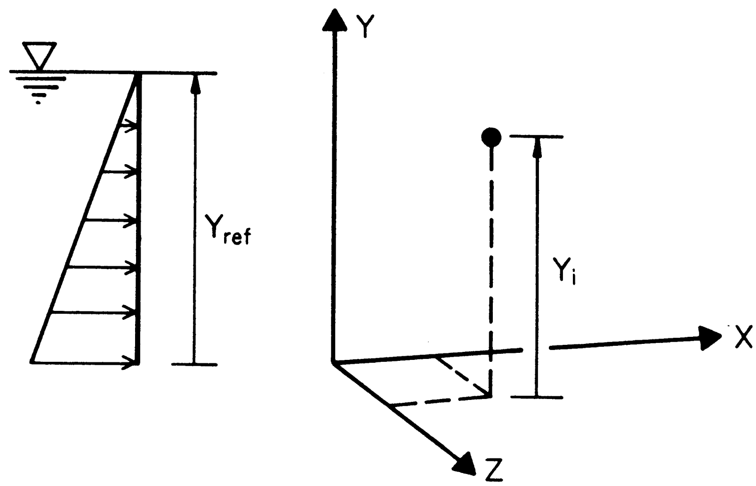
POSITIVE SURFACE LOADING P

If the uniformly distributed surface loading P is input as a positive quantity then it describes pressure loading on faces 2, 4 or 6 and tensile loading on faces 1, 3 or 5. If P is input as a negative quantity then it describes tensile loading on faces 2, 4 or 6 and pressure on faces 1, 3 or 5.

(2) LT = 2: A hydrostatically varying surface pressure on element faces 2, 4 or 6 can be specified by a reference fluid surface and a fluid weight density γ as input. Only one hydrostatic surface pressure card need be input in order to specify a hydrostatic loading on the complete structure. The consistent nodal loads are calculated by the program as follows. At each numerical integration point "i" on an element surface the pressure P_i is calculated from

$$P_i = \gamma \; (Y_i - Y_{ref})$$

where Y_i is the global Y coordinate of the point in question and Y_{ref} specifies the fluid surface assuming gravity acts along the -Y axis



If $P_i > 0$, corresponding to surface tension the contribution is ignored. If an element face is such that $Y_i > Y_{ref}$ for all i (16 integration points are used by program) then no nodal loads will be applied to the element. If some $P_i > 0$ and some $P_i < 0$ for a particular face then approximate nodal loads are obtained for the partially loaded surface.

VI. Thermal Loads

Thermal loads are computed assuming a constant temperature increase ΔT throughout the element.

$$\Delta T = Te - Tr$$

T_e = element temperature specified on element data card (see F above)

T_r = reference stress free temperature (see D above)

VII. Element Load Cases

Element load case A consists of all the contributions from distributed loadings, thermal loadings and gravity loading for all the elements taken collectively.

$$\text{Load case A} = \Sigma \quad PA \times \text{pressure loading} + TA \times \text{thermal loading} + GXA \times \text{gravity X loading} + GYA \times \text{gravity Y loading} + GZA \times \text{gravity Z loading}$$

Element load case A for the set of three dimensional solid elements is added to element load case A for the other element types in the analysis. Element load cases B, C and D are analogous to element load case A. The loading cases for the structure are obtained by adding linear combinations of element load cases A, B, C and D to the nodal loads specified on the joint data cards.

VIII. Stresses

The stresses are computed at 7 points, the center of each face and the center of the element. The stresses computed and output are with respect to the global coordinate system.

APPENDIX C - PROGRAM CAPACITY

C-1 High Speed Storage Requirements

The high speed storage requirements of the program can be changed depending on the size of the problem to be solved. This is done by changing the two Fortran statements at the start of SAP2, i.e.

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

The minimum value of n needed is computed as follows:

n = 10* (number of joints) + M

where

M = the maximum value of each of the following:

(1) Truss elements

M = 5* NMAT   NMAT = number of material types

(2) Beam elements

M = 3*NMAT+12*NFIX+6*NPROP

NFIX = number of fixed end force sets

NPROP = number of different beam properties

(3) Plane stress and plane strain elements.

M = 4*NMAT+11*NMAT*NTC

NTC = number of material temperatures

(4) Axisymmetric quadrilateral

M = 4*NMAT

(5) Three-dimensional solid elements

M = 4*NMAT+4*NLD+2475   NLD = number of element load sets

(6) Plate and shell elements

M = 12*NMAT

(7) Boundary elements

M = 0

(8) Solid thick shell element

M = 4*NMAT+4NLD+6615

Note: (1) A convenient general rule for computing a minimum value of n (except for solid elements) is:

n = 11* (number of joints)

(2) For optimum efficiency, however, a value of n, considerably greater than the minimum, should be used.

(3) If the value of n is set too small an error message is printed and program execution is terminated.

C-2 Low Speed Storage Requirements

For very large problems, the amount of low speed backup storage on the computer will govern the maximum size of structure which can be solved. Six temporary storage files are used with the following maximum storage requirements:

Tape 1

For element stress-displacement transformation matrices

Number of locations on Tape 1 =

$$\sum_{i=1}^{Ne} [(N_s + 2) * N_d + N_s * 4 + 2]_i$$

Where N_e = total number of elements

N_s = number of output stresses associated with an element

N_d = number of displacement degrees of freedom associated with an element

Tape 2

For element stiffness matrices

Number of locations on Tape 2 =

$$\sum_{i=1}^{Ne} [N_d * (N_d + 6) + 1]_i$$

or for temporary storage during solution of equations

Number of locations on Tape 2 =

$$\left| \frac{N_{band}}{N_{eb}} \right| (N_{band} * N_{eb})$$

Where N_{band} = half band width of equations

N_{eb} = number of equations in a block

$$= \frac{(MTOT - 4 * N_\ell)}{(N_{band} + N_\ell + 1) * 2}$$

N_ℓ = number of load conditions

or, for storage of displacements

Number of locations on Tape 2 = $N_{eq} * N_\ell$

Where N_{eq} = total number of equations

Tape 3

For temporary storage during solution of equations

Number of locations on Tape 3 =

$$N_{eq} * (N_{band} + N_{\ell})$$

Tape 4

For storage of total stiffness and load matrices.

Number of locations on Tape 4 =

$$N_{eq} * (N_{band} + N_{\ell})$$

Tape 7

For temporary storage in the formation of the total stiffness matrix

Number of locations on Tape 2 =

$$\sqrt{N_e} * (N_d * (N_d + 6) + 1)$$

or, for temporary storage during solution of equations; same as second option for Tape 2.

Tape 8

For storage of boundary condition array and load multipliers

Number of locations on Tape 8 =

$$6 * N_j + 4 * N_{\ell}$$

where N_j = total number of joints in the system

C-3 Solid-SAP Common Storage Allocation

COMMON A (MTOT)

COMMON/ELPAR/NPAR (14), NUMNP, MBAND, N1, N2, N3, N4, N5, NEQ

COMMON/EM/2594 locations max.

COMMON/JUNK/222 locations max.

APPENDIX D

FORTRAN IV COMPUTER PROGRAM LISTING

```
      PROGRAM    SAP2 (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,
     1 TAPE2,TAPE3,TAPE4,TAPE7,TAPE8)
C  ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** *
C     SAP2 A STATIC ANALYSIS PROGRAM FOR THREE-DIMENSIONAL STRUCTURES
C                     REVISED DECEMBER 1972
C  ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** *
      COMMON /JUNK / HED(12),JUK(210)
      COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
      COMMON / EM / QQQ(2594)
      DIMENSION T(7)
C
C     PROGRAM CAPACITY CONTROLLED BY THE FOLLOWING TWO STATEMENTS ...
      COMMON A(7000)
      MTOT=7000
C
C     PROGRAM CONTROL DATA
C
    5 CALL SECOND(T(1))
      READ (5,100)     HED,NUMNP,NELTYP,LL,NF,NDYN
      IF (NUMNP.EQ.0) STOP
      WRITE(6,200)     HED,NUMNP,NELTYP,LL
C
C     INPUT JOINT DATA--ID ARRAY ON TAPE 8
C
      N1=1
      N2=N1+6*NUMNP
      N3=N2+NUMNP
      N4=N3+NUMNP
      N5=N4+NUMNP
      N6=N5+NUMNP
      IF(N6.GT.MTOT) CALL ERROR(N6-MTOT)
      CALL INPUTJ(A(N1),A(N2),A(N3),A(N4),A(N5),NUMNP,NEQ)
C
C     FORM ELEMENT STIFFNESSES--STIFF. ON TAPE 2 -STRESS MATRIX ON TAPE1
C
      CALL SECOND(T(2))
      MBAND=0
      NUMEL=0
      REWIND 1
      REWIND 2
      DO 900 M=1,NELTYP
      READ  (5,1001) NPAR
      WRITE (1) NPAR
      NUMEL=NUMEL+NPAR(2)
      MTYPE=NPAR(1)
      CALL ELTYPE(MTYPE)
  900 CONTINUE
C
C     INPUT NODAL LOADS                       --- WRITE ON TAPE 3
C
      NEQB=(MTOT-4*LL)/(MBAND+LL+1)/2
      NBLOCK=(NEQ-1)/NEQB +1
      IF (NEQB.GT.NEQ) NEQB=NEQ
      N3=N2+NEQB*LL
      N4=N3+6*LL
      WRITE (6,201) NEQ,MBAND,NEQB,NBLOCK
      CALL SECOND(T(3))
      CALL INL(A(N1),A(N2),A(N3),A(N4),NUMNP,NEQB,LL)
      CALL SECOND(T(4))
C
C     FORM TOTAL STIFFNESS MATRIX --ON TAPE 4
C
      NE2B=2*NEQB
      N2=N1+NEQB*MBAND
      N3=N2+NEQB*LL
      N4=N3+4*LL
      NN2=N1+NE2B*MBAND
      NN3=NN2+NE2B*LL
      NN4=NN3+4*LL
      CALL ADDSTF(A(N1),A(NN2),A(NN3),A(NN4),NUMEL,NBLOCK,NE2B,LL,MBAND)
      CALL SECOND(T(5))
C
C     SOLVE FOR DISPLACEMENT UNKNOWNS
C
      NSB=(MBAND+LL)*NEQB
      NSBB=NEQB*LL*(2+(MBAND-1)/NEQB)
      IF(NSBB.LT.NSB) NSBB=NSB
      N4=N3+NSBB
      CALL USOL(A(N1),A(N3),A(N4),NEQB,MBAND,LL,NBLOCK,NSB,4,3,7,2,2)
      CALL SECOND(T(6))
C
C     POINT DISPLACEMENT
C
      N2=N1+NUMNP*6
      N3=N2+6*LL
      CALL PRINTD(A(N1),A(N2),A(N3),NEQB,NUMNP,LL,NBLOCK,NEQ,2)
C
C     COMPUTE STRESSES
C
   30 N2=N1+4*LL
      N3=N2+NEQB*LL
      LB=(MTOT-N3)/(NEQ +12)
      NDYN=0
      CALL STRESS(A(N1),A(N2),A(N3),NEQB,LB,LL,NEQ,NBLOCK)
C
      CALL SECOND (T(7))
      DO 40 I=1,6
   40 T(I)=T(I+1)-T(I)
      T(7)=T(1)+T(2)+T(3)+T(4)+T(5)+T(6)
      WRITE (6,203) T
C
      GO TO 5
C
  100 FORMAT(12A6/6I5)
  200 FORMAT(1H1,12A6///
     .  29H   NUMBER OF NODAL POINTS    =  ,I5//
     .  29H   NUMBER OF ELEMENT TYPES   =  ,I5//
     .  29H   NUMBER OF LOAD CASES      =  ,I5)
  201 FORMAT(34H2 TOTAL NUMBER OF EQUATIONS          =,I5,
     1         /34H   BANDWIDTH                        =,I5,
     2         /34H   NUMBER OF EQUATIONS IN A BLOCK  =,I5,
     3         /34H   NUMBER OF BLOCKS                 =,I5)
  203 FORMAT ( 12H2OVERALL LOG //
     .             33H   NODAL POINT INPUT.............. ,F8.2//
     .             33H   FORM ELEMENT STIFFNESSES....... ,F8.2//
     .             33H   INPUT NODAL LOADS.............. ,F8.2//
     .             33H   FORM TOTAL STIFFNESS........... ,F8.2//
     .             33H   EQUATION SOLVING............... ,F8.2//
     .             33H   ELEMENT STRESSES............... ,F8.2///
     .             33H   TOTAL SOLUTION TIME............ ,F8.2)
C
 1001 FORMAT (14I5)
      END

      SUBROUTINE INPUTJ(ID,X,Y,Z,T,NUMNP,NEQ)
C-----E. WILSON JULY 1971
      DIMENSION X(1),Y(1),Z(1),ID(NUMNP,6),T(1)
C
C---- READ OR  GENERATE NODAL POINT DATA---------------------------------
      WRITE (6,2000)
      WRITE (6,2001)
      NOLD=0
   10 READ  (5,1000) N,(ID(N,I),I=1,6),X(N),Y(N),Z(N),KN,T(N)
      WRITE (6,2002) N,(ID(N,I),I=1,6),X(N),Y(N),Z(N),KN,T(N)
      IF(NOLD.EQ.0) GO TO 50
C-----CHECK IF GENERATION IS REQUIRED-------------------------------------
      DO 20 I=1,6
      IF(ID(N,I).EQ.0.AND.ID(NOLD,I).LT.0) ID(N,I)=ID(NOLD,I)
   20 CONTINUE
      IF(KN.EQ.0) GO TO 50
      NUM=(N-NOLD)/KN
      NUMN=NUM-1
      IF(NUMN.LT.1) GO TO 50
      XNUM=NUM
```

```
      DX=(X(N)-X(NOLD))/XNUM
      DY=(Y(N)-Y(NOLD))/XNUM
      DZ=(Z(N)-Z(NOLD))/XNUM
      DT=(T(N)-T(NOLD))/XNUM
      K=NOLD
      DO 30 J=1,NUMN
      KK=K
      K=K+KN
      X(K)=X(KK)+DX
      Y(K)=Y(KK)+DY
      Z(K)=Z(KK)+DZ
      T(K)=T(KK)+DT
      DO 30 I=1,6
      ID(K,I)=ID(KK,I)
      IF (ID(K,I).GT.1) ID(K,I)=ID(KK,I)+KN
   30 CONTINUE
C
   50 NOLD=N
      IF(N.NE.NUMNP) GO TO 10
C
C---- PRINT ALL NODAL POINT DATA----------------------------------------
C
      WRITE (6,2003)
      WRITE (6,2001)
      WRITE (6,2005) (N,(ID(N,I),I=1,6),X(N),Y(N),Z(N),T(N),N=1,NUMNP)
C
C-----NUMBER UNKNOWNS AND SET MASTER NODES NEGATIVE---------------------
C
      NEQ=0
      DO 60 N=1,NUMNP
      DO 60 I=1,6
      ID(N,I)=IABS(ID(N,I))
      IF(ID(N,I)-1) 57,58,59
   57 NEQ=NEQ+1
      ID(N,I)=NEQ
      GO TO 60
   58 ID(N,I)=0
      GO TO 60
   59 ID(N,I)=-ID(N,I)
   60 CONTINUE
      WRITE (5,2004) (N,(ID(N,I),I=1,6),N=1,NUMNP)
      REWIND 8
      WRITE (8) ID
C
      RETURN
 1000 FORMAT (7I5,3F10.0,I5,F10.0)
 2000 FORMAT (//23H NODAL POINT INPUT DATA )
 2001 FORMAT (5HONODE 3X 24HBOUNDARY CONDITION CODES 11X
     . 23HNODAL POINT COORDINATES / 7H NUMBER 2X 1HX 4X 1HY 4X 1HZ 3X
     . 2HXX 3X 2HYY 3X 2HZZ12X 1HX 12X 1HY 12X 1HZ 12X 1HT )
 2002 FORMAT (I5,6I5,3F13.3,I5,F13.3)
 2003 FORMAT (//21H GENERATED NODAL DATA)
 2004 FORMAT (//17H EQUATION NUMBERS/
     1 35H    N    X    Y    Z   XX   YY   ZZ /(7I5))
 2005 FORMAT (I5,6I5,4F13.3)
      END
```

```
      SUBROUTINE ELTYPE(MTYPE)
      GO TO (1,2,3,4,5,6,7,8) MTYPE
C
C     THREE DIMENSIONAL TRUSS ELEMENTS
C
    1 CALL TRUSS
      GO TO 900
C
C     THREE DIMENSIONAL BEAM ELEMENTS
C
    2 CALL BEAM
      GO TO 900
C
C     PLANE STRESS ELEMENTS
C
    3 CALL PLANE
      GO TO 900
C
C     AXISYMMETRIC SOLID ELEMENTS
C
    4 CALL PLANE
      GO TO 900
C
C     THREE DIMENSIONAL SOLID ELEMENTS
C
    5 CALL THREED
      GO TO 900
C
C     PLATE BENDING ELEMENTS
C
    6 CALL SHELL
      GO TO 900
C
C
    7 CALL BOUND
      GO TO 900
C
C     THICK SHELL ELEMENTS
C
    8 CALL THKSHL
C
  900 RETURN
      END
```

```
      SUBROUTINE INL(ID,B,TR,TMASS,NUMNP,NEQB,LL)
C
C     INPUT NODAL LOADS AND MASSES
C
      DIMENSION ID(NUMNP,6),B(NEQB,LL),TR(6,LL),TMASS(NEQB)
      COMMON / JUNK / R(6),TXM(6)
C
      NT=3
      REWIND NT
      KSHF=0
      WRITE (6,2002)
      DO 750 I=1,NEQB
      TMASS(I)=0.
      DO 750 K=1,LL
  750 B(I,K)=0.0
C
      DO 900 NN=1,NUMNP
C
      DO 100 I=1,6
      TXM(I)=0.
      DO 100 J=1,LL
  100 TR(I,J)=0.0
C
      IF(NN.EQ.1) GO TO 300
  150 IF(N.NE.NN) GO TO 400
      DO 200 I=1,6
      IF (L) 180,180,190
  180 TXM(I)=R(I)
      GO TO 200
  190 TR(I,L)=R(I)
  200 CONTINUE
  300 READ (5,1001) N,L,R
      IF (N.EQ.0) GO TO 150
      WRITE(6,2001) N,L,R
      GO TO 150
C
  400 DO 800 J=1,6
```

```
      II=ID(NN,J)-KSHF
      IF (II) 800,800,500
  500 DO 600  K=1,LL
  600 B(II,K)=TR(J,K)
      TMASS(II)=TXM(J)
  610 IF(II.NE.NEQB) GO TO 800
      WRITE (NT) B,TMASS
      KSHF=KSHF+NEQB
      DO 700 I=1,NEQB
      TMASS(I)=0.
      DO 700 K=1,LL
  700 B(I,K)=0.0
  800 CONTINUE
  900 CONTINUE
      WRITE (NT) B,TMASS
C
      RETURN
 1001 FORMAT (2I5,7F10.0)
 2001 FORMAT (2I5,7F10.3)
 2002 FORMAT (23H2.....NODAL POINT LOADS // 10H NODE LOAD 23X
     . 14HAPPLIED LOADS              / 10H  NO. CASE  6X  2HRX  8X
     .  2HRY 8X 2HRZ 8X 2HMX 8X 2HMY 8X 2HMZ )
      END


      SUBROUTINE ERROR(N)
      WRITE (6,2000) N
 2000 FORMAT (// 20H STORAGE EXCEEDED BY  I6)
      STOP
      END


      SUBROUTINE ADDSTF(A,B,STR,TMASS,NUMEL,NBLOCK,NE2B,LL,MBAND)
C     FORMS GLOBAL EQUILIBRIUM EQUATIONS IN BLOCKS
      DIMENSION A(NE2B,MBAND),B(NE2B,LL),STR(4,LL),TMASS(NE2B),SS(1)
      COMMON /EM/ LRD,ND,LM(2592)
      EQUIVALENCE (SS,ND)
      NEQB=NE2B/2
      K=NEQB+1
      X=NBLOCK
      MB=SQRT(X)
      MB=MB/2+1
      NEBB=MB*NE2B
      MM=1
C
      NSHIFT=0
      REWIND 3
      REWIND 4
C
C     READ ELEMENT LOAD MULTIPLIERS
C
      WRITE (6,2000)
      DO 50 L=1,LL
      READ  (5,1002)    (STR(I,L),I=1,4)
   50 WRITE (6,2002) L,(STR(I,L),I=1,4)
      WRITE (8) STR
C
C     FORM EQUATIONS IN BLOCKS      ( 2 BLOCKS AT A TIME)
C
      DO 1000 M=1,NBLOCK ,2
      DO 100 I=1,NE2B
      DO 100 J=1,MBAND
  100 A(I,J)=0.
      READ (3) ((B(I,L),I=1,NEQB),L=1,LL),(TMASS(I),I=1,NEQB)
      IF (M.EQ.NBLOCK) GO TO 200
```

```
      READ (3) ((B(I,L),I=K,NE2B),L=1,LL),(TMASS(I),I=K,NE2B)
  200 CONTINUE
C
      REWIND 7
      REWIND 2
      NA=7
      NUME=NUM7
      IF (MM.NE.1) GO TO 75
      NA=2
      NUME=NUMEL
      NUM7 =0
C
   75 DO 700 N=1,NUME
      READ (NA) LRD,(SS(I),I=1,LRD)
      DO 600 I=1,ND
      LMN=1-LM(I)
      II=LM(I)-NSHIFT
      IF (II.LE.0.OR.II.GT.NE2B) GO TO 600
      DO 300 L=1,LL
      DO 300 J=1,4
      KK=ND*(ND+J)+1
  300 B(II,L)=B(II,L)+SS(I+KK)*STR(J,L)
      DO 500 J=1,ND
      JJ=LM(J)+LMN
      IF(JJ) 500,500,390
  390 KK=ND*J+1
  400 A(II,JJ)=A(II,JJ)+SS(I+KK)
  500 CONTINUE
  600 CONTINUE
C
C     DETERMINE IF STIFFNESS IS TO BE PLACED ON TAPE 7
C
      IF (MM.GT.1) GO TO 700
      DO 650 I=1,ND
      II=LM(I) -NSHIFT
      IF(II.GT.NE2B.AND.II.LE.NEBB) GO TO 660
  650 CONTINUE
      GO TO 700
  660 WRITE (7) LRD,(SS(I),I=1,LRD)
      NUM7=NUM7+1
C
  700 CONTINUE
      WRITE(4) ((A(I,J),I=1,NEQB),J=1,MBAND),((B(I,L),I=1,NEQB),L=1,LL)
      IF(M.EQ.NBLOCK) GO TO 1000
      WRITE(4) ((A(I,J),I=K,NE2B),J=1,MBAND),((B(I,L),I=K,NE2B),L=1,LL)
      IF (MM.EQ.MB) MM=0
      MM=MM+1
 1000 NSHIFT=NSHIFT+NE2B
C
      RETURN
 1002 FORMAT (4F10.0)
 2000 FORMAT (10H2STRUCTURE 12X 25HELEMENT LOAD MULTIPLIERS /
     . 10H LOAD CASE 9X 1HA 9X 1HB 9X 1HC 9X 1HD/)
 2002 FORMAT (I6,7X,4F10.3)
      END


      SUBROUTINE PRINTD(ID,D,B,NEQB,NUMNP,LL,NBLOCK,NEQ,NT)
C
      DIMENSION ID(NUMNP,6),B(NEQB,LL),D(6,LL)
C
      REWIND NT
      REWIND 8
      READ (8) ID
      M=NEQ
      NN=NEQB*NBLOCK
      WRITE (6,2003)
      N=NUMNP
C
      DO 500 KK=1,NUMNP
```

```
C
      I=6
      DO 250 II=1,6
      DO 100 L=1,LL
  100 D(I,L)=0.
      IF(M.GT.NN) GO TO 150
      IF (M.EQ.0) GO TO 150
      READ (NT) B
      NN=NN-NEQB
  150 IF(ID(N,I).LT.1) GO TO 250
      K=M-NN
      M=M-1
C
      DO 200 L=1,LL
  200 D(I,L)=B(K,L)
  250 I=I-1
C
      WRITE (6,2004) N,(L,(D(I,L),I=1,6),L=1,LL)
C
  500 N=N-1
C
      RETURN
C
 2003 FORMAT (40H1.......NODE DISPLACEMENTS AND ROTATIONS//
     .  5H NODE  5H LOAD  11X  1HX  11X  1HY  11X  1HZ  9X  2HXX
     .  9X  2HYY  9X  2HZZ)
 2004 FORMAT (1H0,I4,I5,1P3E12.3,3E11.2/(I10,3E12.3,3E11.2))
      END
```

```
      SUBROUTINE USOL (A,B,MAXB,NEQB,MB,LL,NBLOCK,NSB,NORG,NBKS,NT1,
     .                                                      NT2,NRST)
      DIMENSION A(NSB),B(NSB),MAXB(NEQB)
C-----------------------------------------------------------------------
      NC=MB+LL
      NBR=(MB-1)/NEQB+1
      INC=NEQB-1
      NMB=NEQB*MB
      N2=NT2
      N1=NT1
      REWIND NORG
      REWIND NBKS
C
C---- REDUCE EQUATIONS  BLOCK-BY-BLOCK ---------------------------------
C
      DO 900 N=1,NBLOCK
      IF (N.GT.1.AND.NBR.EQ.1) GO TO 110
      IF (NBR.EQ.1) GO TO 105
      REWIND N1
      REWIND N2
  105 NI=N1
      IF(N.EQ.1) NI=NORG
      READ (NI) A
  110 DO 300 I=1,NEQB
      D=A(I)
      IF(D) 115,300,120
  115 M=NEQB*(N-1)+I
      WRITE (6,116) M,D
  116 FORMAT (33HOSET OF EQUATIONS MAY BE SINGULAR   /
     .  26H DIAGONAL TERM OF EQUATION   I8, 8H  EQUALS   1PE12.4)
C
  120 II=I
      DO 125 J=2,NC
      II=II+NEQB
  125 A(II)=A(II)/D
C
      DO 130 J=I,NMB,NEQB
      IF (A(J).NE.0.) MAXB(I)=J
  130 CONTINUE
C
      JL=I+1
      IF (JL.GT.NEQB) GO TO 300
      II=I
      DO 200 J=JL,NEQB
      II=II+NEQB
      IF(II.GT.NMB) GO TO 200
      C=A(II)
      IF (C.EQ.0.0) GO TO 200
      C=C*A(I)
C
      KK=J-II
      MAX=MAXB(I)
      DO 150 JJ=II,MAX,NEQB
  150 A(JJ+KK)=A(JJ+KK)-C*A(JJ)
C
      KK=J +NMB
      JJ=I+NMB
      DO 175 L=1,LL
      A(KK)=A(KK)-C*A(JJ)
      KK=KK+NEQB
  175 JJ=JJ+NEQB
  200 CONTINUE
  300 CONTINUE
      WRITE (NBKS) A,MAXB
C
C---- SUBSTITUTE INTO REMAINING EQUATIONS ------------------------------
C
      DO 800 NN=1,NBR
      IF(N+NN.GT.NBLOCK) GO TO 800
      NI=N1
      IF(N.EQ.1) NI=NORG
      IF(NN.EQ.NBR) NI=NORG
      READ (NI) B
      IL=1+NN*NEQB*NEQB
      DO 700 I=1,NEQB
      II=IL
      DO 690 K=1,NEQB
      IF (II.GT.NMB) GO TO 690
      C=A(II)
      IF (C.EQ.0.0) GO TO 690
      C=C*A(K)
      MAX=MAXB(K)
C
      KK=I-II
      DO 640 JJ=II,MAX,NEQB
  640 B(JJ+KK)=B(JJ+KK)-C*A(JJ)
C
      KK=I+NMB
      JJ=K+NMB
      DO 650 L=1,LL
      B(KK)=B(KK)-C*A(JJ)
      KK=KK+NEQB
  650 JJ=JJ+NEQB
C
  690 II=II-INC
  700 IL=IL+NEQB
C
      IF(NBR.NE.1) GO TO 750
      DO 740 I=1,NSB
  740 A(I)=B(I)
      GO TO 800
  750 WRITE (N2) B
  800 CONTINUE
C
      M=N1
      N1=N2
  900 N2=M
C
C---- BACKSUBSTITUTION - RESULTS ON TAPE NRST  -----------------------
C
      LS=LL*NEQB
      NEB=NEQB*(NBR+1)
      NUM=NBR*NEQB
      MAX=NEB*LL
      DO 905 I=1,MAX
```

```
  905 B(I)=0.
      REWIND NRST
C-----------------------------------------------------------------------
      DO 1000 N=1,NBLOCK
      BACKSPACE NBKS
      READ (NBKS) A,MAXB
      BACKSPACE NBKS
      DO 910 L=1,LL
      K=L*NEB
      DO 910 J=1,NUM
      I=K-NEQB
      B(K)=B(I)
  910 K=K-1
C
      I=NMB
      DO 920 L=1,LL
      K=(L-1)*NEB
      DO 920 J=1,NEQB
      I=I+1
      K=K+1
  920 B(K)=A(I)
C
      DO 955 I=1,NEQB
      J=NEQB+1-I
      MAX=MAXB(J)
      IF (A(J).EQ.0.) GO TO 955
      DO 950 L=1,LL
      KK=J+(L-1)*NEB
      JJ=KK+1
      IL=J+NEQB
      C=B(KK)
      DO 940 II=IL,MAX,NEQB
      C=C-A(II)*B(JJ)
  940 JJ=JJ+1
  950 B(KK)=C
  955 CONTINUE
C
      I=0
      DO 960 L=1,LL
      K=(L-1)*NEB
      DO 960 J=1,NEQB
      K=K+1
      I=I+1
  960 A(I)=B(K)
C
      WRITE (NRST) (A(I),I=1,LS)
 1000 CONTINUE
C-----------------------------------------------------------------------
      RETURN
      END
```

```
      SUBROUTINE STRESS(STR,B,D,NEQB,LB,LL,NEQ,NBLOCK)
C
      DIMENSION D(NEQ,LB),B(NEQB,LL),STR(4,LL)
      COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,MEQ
      COMMON /JUNK/ LT,LH
C
      READ (8) STR
      NT=(LL-1)/LB +1
      LH=0
      REWIND 3
C
      DO 1000 II=1,NT
C
      LT =LH+1
      LLT=1-LT
      LH=LT+LB-1
      IF(LH.GT.LL) LH=LL
C
C     MOVE DISPLACEMENTS INTO CORE FOR LB LOAD CONDITIONS
C
      REWIND 2
      IF(NDYN.EQ.3) READ (2)
      NQ=NEQB*NBLOCK
      DO 200 NN=1,NBLOCK
      READ (2) B
      N=NEQB
      IF (NN.EQ.1) N=NEQ-NQ+NEQB
      NQ=NQ-NEQB
      DO 200 J=1,N
      I=NQ+J
      DO 200 L=LT,LH
      K=L+LLT
  200 D(I,K)=B(J,L)
      LK=LH-LT+1
C
C     CALCULATE STRESSES FOR ALL ELEMENTS FOR LB LOAD CONDITIONS
C
      REWIND 1
      DO 1000 M=1,NELTYP
      READ (1) NPAR
      MTYPE=NPAR(1)
      NPAR(1)=0
      CALL ELTYPE(MTYPE)
 1000 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE CALBAN(MBAND,NDIF,LM,XM,S,P,ND,NDM)
C-----CALCULATES BAND WIDTH AND WRITES STIFFNESS MATRIX ON TAPE 2
      DIMENSION LM(1),XM(1),S(NDM,NDM),P(NDM,4)
      MIN=100000
      MAX=0
      DO 800 L=1,ND
      IF (LM(L).EQ.0) GO TO 800
      IF (LM(L).GT.MAX) MAX=LM(L)
      IF (LM(L).LT.MIN) MIN=LM(L)
  800 CONTINUE
      NDIF=MAX-MIN+1
      IF (NDIF.GT.MBAND) MBAND=NDIF
      LRD=1+ND*(ND+6)
      WRITE (2) LRD,ND,(LM(I),I=1,ND),((S(I,J),I=1,ND),J=1,ND),
     1 ((P(I,J),I=1,ND),J=1,4),(XM(I),I=1,ND)
      RETURN
      END
```

```
      SUBROUTINE STRSC(STR,D,NEQ,NTAG)
      DIMENSION STR(4,1),D(NEQ,1)
      COMMON /JUNK/ LT,LH,L,SG(20),SIG(7),EXTRA(150)
      COMMON /EM/ NS,ND,LM(48),B(48,48),TI(48,4)
C
      IF (NTAG.EQ.0) GO TO 800
      LL=L-LT+1
      DO 300 I=1,NS
      SG(I)=0.0
      DO 300 J=1,4
  300 SG(I)=SG(I)+TI(I,J)*STR(J,L)
      DO 500 J=1,ND
      JJ=LM(J)
      IF(JJ.EQ.0) GO TO 500
      DO 400 I=1,NS
```

```
  400 SG(I)=SG(I)+B(I,J)*D(JJ,LL)
C
  500 CONTINUE
      GO TO 900
  800 READ  (1) ND,NS,(LM(I),I=1,ND),(( B(I,J),I=1,NS),J=1,ND),
     1 ((TI(I,J),I=1,NS),J=1,4)
  900 RETURN
      END
```

```
      SUBROUTINE POSINV(A,NMAX,NDD)
C
      DIMENSION A(NDD,NDD)
C
      DO 200 N=1,NMAX
C
      D=A(N,N)
      DO 100 J=1,NMAX
  100 A(N,J)=-A(N,J)/D
C
      DO 150 I=1,NMAX
      IF(N-I) 110,150,110
  110 DO 140 J=1,NMAX
      IF(N-J) 120,140,120
  120 A(I,J)=A(I,J)+A(I,N)*A(N,J)
  140 CONTINUE
  150 A(I,N)=A(I,N)/D
C
      A(N,N)=1.0/D
C
  200 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE VECTOR(V,XI,YI,ZI,XJ,YJ,ZJ)
      DIMENSION V(4)
      X=XJ-XI
      Y=YJ-YI
      Z=ZJ-ZI
      V(4)=SQRT(X*X+Y*Y+Z*Z)
      V(3)=Z/V(4)
      V(2)=Y/V(4)
      V(1)=X/V(4)
      RETURN
      END
```

```
      SUBROUTINE CROSS(A,B,C)
      DIMENSION A(4),B(4),C(4)
      X=A(2)*B(3)-A(3)*B(2)
      Y=A(3)*B(1)-A(1)*B(3)
      Z=A(1)*B(2)-A(2)*B(1)
      C(4)=SQRT(X*X+Y*Y+Z*Z)
      C(3)=Z/C(4)
      C(2)=Y/C(4)
      C(1)=X/C(4)
      RETURN
      END
```

```
      FUNCTION DOT(A,B)
      DIMENSION A(4),B(4)
      DOT=A(1)*B(1)+A(2)*B(2)+A(3)*B(3)
      RETURN
      END
```

```
      SUBROUTINE BOUND
      COMMON A(1)
      COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
      COMMON / JUNK / LT,LH,L,SIG(20)
C
      IF (NPAR(1).EQ.0) GO TO 500
      CALL CLAMP (NPAR(2),A(N1),A(N2),A(N3),A(N4),NUMNP,MBAND)
      RETURN
C
  500 WRITE (6,2002)
      NUME=NPAR(2)
      DO 800 MM=1,NUME
      CALL STRSC (A(N1),A(N3),NEQ,0)
      WRITE (6,2001)
      DO 800 L=LT,LH
      CALL STRSC (A(N1),A(N3),NEQ,1)
      WRITE (6,3002) MM,L,(SIG(I),I=1,2)
  800 CONTINUE
      RETURN
C
 2001 FORMAT (/)
 2002 FORMAT (/19H0 CONSTRAINT FORCES//
     1 55H      NUMBER    LOAD CASE           FORCE           MOMENT//)
 3002 FORMAT (1X,2I10,4X,2E15.5)
      END

      SUBROUTINE CLAMP (NUMEL,ID,X,Y,Z,NUMNP,MBAND)
      COMMON/EM/LM(24),ND,NS,S(24,24),P(24,4),XM(24),SA(12,24),TT(12,4)
      DIMENSION X(1),Y(1),Z(1),ID(NUMNP,1)
      COMMON / JUNK / R(6),RM(4)
C
      DO 30 I=1,1058
   30 LM(I)=0
      NS=2
      ND=6
      READ (5,1005) RM
C
      NE=0
      WRITE (6,2000) NUMEL
  210 KG=0
      MARK=0
C
  200 READ (5,1000) NP,NI,NJ,NK,NL,KD,KR,KN,SD,SR,TRACE
      IF (TRACE.EQ.0.) TRACE=1.0E+10
      IF (KG.GT.0) GO TO 550
C
      KG=KN
      IF(NJ.EQ.0) GO TO 250
      X1=X(NJ)-X(NI)
      Y1=Y(NJ)-Y(NI)
      Z1=Z(NJ)-Z(NI)
      X2=X(NL)-X(NK)
      Y2=Y(NL)-Y(NK)
      Z2=Z(NL)-Z(NK)
      T1=Y1*Z2-Y2*Z1
      T2=Z1*X2-Z2*X1
      T3=X1*Y2-X2*Y1
      GO TO 260
  250 T1=X(NI)-X(NP)
      T2=Y(NI)-Y(NP)
      T3=Z(NI)-Z(NP)
  260 XL=T1*T1+T2*T2+T3*T3
      XL=SQRT(XL)
      T1=T1/XL
      T2=T2/XL
      T3=T3/XL
C
      IF (KD.EQ.0) GO TO 300
      SA(1,1)=T1*TRACE
      SA(1,2)=T2*TRACE
      SA(1,3)=T3*TRACE
      S(1,1)=T1*T1*TRACE
      S(1,2)=T1*T2*TRACE
      S(1,3)=T1*T3*TRACE
      S(2,2)=T2*T2*TRACE
      S(2,3)=T2*T3*TRACE
      S(3,3)=T3*T3*TRACE
      PP=TRACE*SD
      R(1)=T1*PP
      R(2)=T2*PP
      R(3)=T3*PP
      GO TO 350
  300 S(1,1)=0.
      S(1,2)=0.
      S(1,3)=0.
      S(2,2)=0.
      S(2,3)=0.
      S(3,3)=0.
      SA(1,1)=0.
      SA(1,2)=0.
      SA(1,3)=0.
      R(1)=0.
      R(2)=0.
      R(3)=0.
C
  350 IF (KR.EQ.0) GO TO 400
      SA(2,4)=T1*TRACE
      SA(2,5)=T2*TRACE
      SA(2,6)=T3*TRACE
      S(4,4)=T1*T1*TRACE
      S(4,5)=T1*T2*TRACE
      S(4,6)=T1*T3*TRACE
      S(5,5)=T2*T2*TRACE
      S(5,6)=T2*T3*TRACE
      S(6,6)=T3*T3*TRACE
      PP=TRACE*SR
      R(4)=T1*PP
      R(5)=T2*PP
      R(6)=T3*PP
      GO TO 450
  400 S(4,4)=0.
      S(4,5)=0.
      S(4,6)=0.
      S(5,5)=0.
      S(5,6)=0.
      S(6,6)=0.
      SA(2,4)=0.
      SA(2,5)=0.
      SA(2,6)=0.
      R(4)=0.
      R(5)=0.
      R(6)=0.
  450 DO 500 I=2,6
      I1=I-1
      DO 500 J=1,I1
  500 S(I,J)=S(J,I)
      DO 520 I=1,ND
      DO 520 J=1,4
  520 P(I,J)=R(I)*RM(J)
      NN=NP
      NNI=NI
      NNJ=NJ
      NNK=NK
      NNL=NL
      NKD=KD
      NKR=KR
      SSD=SD
      SSR=SR
      TTR=TRACE
      GO TO 560
  550 MARK=1
  555 NN=NN+KG
      NNI=NNI+KG
  560 NE=NE+1
```

```
      WRITE (6,2100) NN,NNI,NNJ,NNK,NNL,NKD,NKR,KN,SSD,SSR,TTR,NE
C
      DO 600 I=1,ND
  600 LM(I)=ID(NN,I)
C
      NDM=24
      CALL CALBAN (MBAND,NDIF,LM,XM,S,P,ND,NDM)
      WRITE (1) ND,NS,(LM(L),L=1,ND),((SA(L,K),L=1,NS),K=1,ND),
     1 ((TT(L,K),L=1,NS),K=1,4)
C
      IF(NE.EQ.NUMEL) GO TO 700
      IF (NN.LT.NP) GO TO 555
      IF (MARK.EQ.1) GO TO 210
      GO TO 200
  700 WRITE (6,2005) RM
      RETURN
C
 1000 FORMAT (8I5,3F10.0)
 1005 FORMAT (4F10.0)
 2000 FORMAT (25H1 ...BOUNDARY ELEMENTS...//28H ELEMENT TYPE.........=
     .  7//23H NUMBER OF ELEMENTS...= I5 ///
     . 6X 4HNODE 40H ..NODES DEFINING CONSTRAINT DIRECTION.. 5X 5HCODES
     .  5X 7X 5HDISPL 5X 8HROTATION 6X 5HSTIFF 5X 10HCONSTRAINT /9X 1HN
     . 5X 2HNI 8X 2HNJ 8X 2HNK 8X 2HNL 3X 2HKD 3X 2HKR 3X 2HKN 11X 1HD
     . 11X 1HR 11X 1HS 7X 5HNUMBER)
 2005 FORMAT (/////25H ELEMENT LOAD MULTIPLIERS//
     .    9X 1HA 9X 1HB 9X 1HC 9X 1HD /4F10.4)
 2100 FORMAT (5I10,3I5,1P3E12.2,I11)
      END
```

```
      SUBROUTINE TRUSS
C
      COMMON A(1)
      COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
      COMMON / JUNK / LT,LH,L,SIG(20)
C
      IF(NPAR(1).EQ.0) GO TO 500
      N6=N5+NUMNP
      N7 =N6+NPAR(3)
      N8 =N7+NPAR(3)
      N9 =N8+NPAR(3)
      N10=N9+NPAR(3)
      MM=N10+NPAR(3)-MTOT
      IF(MM.GT.0) CALL ERROR(MM)
      CALL RUSS(A(N1),A(N2),A(N3),A(N4),A(N5),A(N6),A(N7),A(N8),A(N9),
     .          A(N10),NUMNP)
      RETURN
C
  500 WRITE (6,2002)
      NUME=NPAR(2)
      DO 800 MM=1,NUME
      CALL STRSC (A(N1),A(N3),NEQ,0)
      WRITE (6,2001)
      DO 800 L=LT,LH
      CALL STRSC (A(N1),A(N3),NEQ,1)
      WRITE (6,3002) MM,L,SIG(1),SIG(2)
  800 CONTINUE
      RETURN
C
 2001 FORMAT (/)
 2002 FORMAT (//23H   TRUSS MEMBER ACTIONS //
     .           46H0 MEMBER    LOAD          STRESS            FORCE  )
 3002 FORMAT (2I8,F15.5,F15.3)
      END
```

```
      SUBROUTINE RUSS (ID,X,Y,Z,T,E,THERM,DEN,AREA,WT,NUMNP)
C
      DIMENSION X(1),Y(1),Z(1),ID(NUMNP,1),E(1),THERM(1),DEN(1),AREA(1)
     . ,T(1),WT(1)
      COMMON /ELPAR/ NPAR(14),NNNNN,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
      COMMON /EM/LM(24),ND,NS,S(24,24),P(24,4),XM(24),ST(12,24),TT(12,4)
      COMMON /JUNK/ EMUL(4,4),I,J,K,L,M,N,II,JJ,KK,MTYPE,TEMP,DX,DY,DZ,
     1 XL2,XL,XX,YY,F,FT,FX,FY,FZ,MIN,MAX,NDIF,KKK,TEM,MTYP
C
C     CONTROL INFORMATION AND MEMBER PROPERTIES
C
      NUME=NPAR(2)
      NUMMAT=NPAR(3)
      WRITE (6,2000) NUME,NUMMAT
      WRITE (6,2001)
      DO 10 I=1,NUMMAT
      READ  (5,1001) N,E(N),THERM(N),DEN(N),AREA(N),WT(N)
   10 WRITE (6,2002) N,E(N),THERM(N),DEN(N),AREA(N),WT(N)
C
C     ELEMENT LOAD MULTIPLIERS
C
      READ (5,1003)  EMUL
      WRITE (6,2003) EMUL
C
C     ELEMENT INFORMATION
      WRITE (6,2005)
C
      N=1
  100 READ (5,1004)  M,II,JJ,MTYP,TEM,KK
      IF(KK.EQ.0) KK=1
  120 IF(M.NE.N) GO TO 200
      I=II
      J=JJ
      MTYPE=MTYP
```

```
      REFT=TEM
      KKK=KK
C
C     1. FORM ELEMENT STIFFNESS AND STRESS MATRICES
C
  200 DX=X(I)-X(J)
      DY=Y(I)-Y(J)
      DZ=Z(I)-Z(J)
      XL2=DX*DX+DY*DY+DZ*DZ
      XL=SQRT(XL2)
      XX=E(MTYPE)*AREA(MTYPE)*XL
      ST(1,1)=DX/XL2
      ST(1,2)=DY/XL2
      ST(1,3)=DZ/XL2
      ST(1,4)=-ST(1,1)
      ST(1,5)=-ST(1,2)
      ST(1,6)=-ST(1,3)
C
      DO 300 L=1,6
      YY=ST(1,L)*XX
      DO 250 K=L,6
      S(K,L)=ST(1,K)*YY
  250 S(L,K)=S(K,L)
      ST(1,L)=E(MTYPE)*ST(1,L)
  300 ST(2,L)=AREA(MTYPE)*ST(1,L)
C
C     2. INERTIA AND THERMAL LOADS
C
      F=WT(MTYPE)*XL/2.
      TEMP=(T(I)+T(J))*0.5 - REFT
      FT=-TEMP*THERM(MTYPE)*E(MTYPE)*AREA(MTYPE)
      FX=DX*FT/XL
      FY=DY*FT/XL
      FZ=DZ*FT/XL
C
      DO 350 L=1,4
      TT(2,L)=FMUL(L,4)*FT
      TT(1,L)=TT(2,L)/AREA(MTYPE)
      P(1,L)=FMUL(L,1)*F-FMUL(L,4)*FX
      P(2,L)=FMUL(L,2)*F-FMUL(L,4)*FY
      P(3,L)=FMUL(L,3)*F-FMUL(L,4)*FZ
      P(4,L)=FMUL(L,1)*F+FMUL(L,4)*FX
      P(5,L)=FMUL(L,2)*F+FMUL(L,4)*FY
  350 P(6,L)=FMUL(L,3)*F+FMUL(L,4)*FZ
      F=DEN(MTYPE)*AREA(MTYPE)*XL/2.
      DO 375 L=1,6
  375 XM(L)=F
C
C     3. FORM LOCATION MATRIX AND COMPUTE BAND WIDTH
C
      DO 400 L=1,3
      LM(L)=ID(I,L)
  400 LM(L+3)=ID(J,L)
C
      ND=6
      NS=2
      NDM=24
      CALL CALBAN (MBAND,NDIF,LM,XM,S,P,ND,NDM)
      WRITE (1) ND,NS,(LM(L),L=1,ND),((ST(L,K),L=1,NS),K=1,ND),
     1 ((TT(L,K),L=1,NS),K=1,4)
C
C     4. CHECK FOR MORE ELEMENTS
C
      WRITE (6,2004) N,I,J,MTYPE,REFT,NDIF
      IF (N.EQ.NUME) RETURN
      N=N+1
      I=I+KKK
      J=J+KKK
      IF(N.GT.M) GO TO 100
      GO TO 120
C
 1001 FORMAT (I5,5F10.0)
 1003 FORMAT (4F10.0)
 1004 FORMAT (4I5,1F10.0,I5)
 2000 FORMAT (///25H NUMBER OF TRUSS MEMBERS= I5/
     1 25H NUMBER OF DIFF. MEMBERS= I5)
 2001 FORMAT (///1X,4HTYPE,14X,1HE,10X,5HALPHA,12X,3HDEN,11X,4HAREA
     1 11X,4HWT/L )
 2002 FORMAT (I5,5E15.7)
 2003 FORMAT(///25H ELEMENT LOAD MULTIPLIERS / 20X,1HA,14X,1HB,14X,1HC,
     1 14X,1HD,/6H X-DIR4E15.6/ 6H Y-DIR4E15.6/ 6H Z-DIR4E15.6/
     2 6H  TEMP4E15.5)
 2004 FORMAT (4I6,F10.2,I7)
 2005 FORMAT (///42H     N      I      J  TYPE       TEMP    BAND  )
      END
```

```
      SUBROUTINE BEAM
C
      COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
      COMMON / JUNK / LT,LH,L,SIG(20)
      COMMON A(1)
C
      IF(NPAR(1).EQ.0) GO TO 500
      N6=N5+NPAR(5) + NUMNP
      N7=N6+NPAR(5)
      N8=N7+NPAR(5)
      N9=N8+12*NPAR(4)
      N10=N9+6*NPAR(3)
      IF(N10.GT.MTOT) CALL ERROR(N10-MTOT)
      CALL TEAM(NPAR(2),NPAR(3),NPAR(4),NPAR(5),A(N1),A(N2),A(N3),
     1 A(N4),A(N5),A(N6),A(N7),A(N8),A(N9),NUMNP,MBAND)
      RETURN
C
  500 WRITE (6,2002)
      NUME=NPAR(2)
      DO 800 MM=1,NUME
      CALL STRSC (A(N1),A(N3),NEQ,0)
      WRITE (6,2001)
      DO 800 L=LT,LH
      CALL STRSC (A(N1),A(N3),NEQ,1)
      WRITE(6,3002) MM,L,(SIG(I),I=1,12)
  800 CONTINUE
      RETURN
 2001 FORMAT (/)
 2002 FORMAT(/29HO.....BEAM FORCES AND MOMENTS//
     . 104HOBEAM LOAD 5X 5HAXIAL  2(7X,5HSHEAR),5X 7HTORSION
     .  2(5X,7HBENDING)/ 10H  NO.  NO.  8X  2HR1  10X  2HR2  10X
     .  2HR3 10X 2HM1 10X 2HM2 10X 2HM3)
 3002 FORMAT (I5,I4,1PE11.3,5E12.3/8X,6E12.3/)
      END


      SUBROUTINE TEAM(NBEAM,NUMETP,NUMFIX,NUMMAT,ID,X,Y,Z,E,G,RO,
     .SFT, COPROP,NUMNP,MBAND)
C
C     FORMS 3-D BEAM STIFFNESS AND STRESS ARRAYS
C
      COMMON/EM/LM(24),ND,NS,ASA(24,24),RF(24,4),XM(24),SA(12,24),
     .     SF(12,4)
      DIMENSION X(1),Y(1),Z(1),ID(NUMNP,1),E(1),G(1),SFT(NUMFIX,1)
     .  ,COPROP(NUMETP,1),RO(1),EMUL(3,4)
      COMMON /NEWB/ LC(4),T(3,3),JK(5),MELTYP,DL,MATTYP
      DIMENSION ILC(4),TI(3,3),TJ(3,3),STIF(722),TS(2,2),LS(4)
      EQUIVALENCE (STIF,LM)
C
C     INITIALIZATION
C
      WRITE (6,2005) NBEAM,NUMETP,NUMFIX,NUMMAT
      N=0
      DO 5 I=1,1058
    5 STIF(I)=0.
C
C     READ AND PRINT MATERIAL PROPERTY DATA
C
      WRITE (6,2001)
      DO 10 I=1,NUMMAT
      READ  (5,1001) N,E(N),G(N),RO(N)
      WRITE (6,2002) N,E(N),G(N),RO(N)
   10 G(N)=0.5*E(N)/(1.+G(N))
C
C     READ AND PRINT GEOMETRIC PROPERTIES OF COMMON ELEMENTS.
C
      WRITE (6,2003)
      DO 30 I=1,NUMETP
      READ  (5,1002) N,(COPROP(N,J),J=1,6)
      IF((COPROP(N,1).NE.0.0).AND.(COPROP(N,4).NE.0.0).AND.
     1  (COPROP(N,5).NE.0.0).AND.(COPROP(N,6).NE.0.0))  GO TO 20
      WRITE (6,2013)
      CALL EXIT
   20 WRITE (6,2004) N,(COPROP(N,J),J=1,6)
   30 CONTINUE
C
C     ELEMENT LOAD MULTIPLIERS
C
      READ  (5,1006) ((EMUL(I,J),J=1,4),I=1,3)
      WRITE (6,2006) ((EMUL(I,J),J=1,4),I=1,3)
C
C     READ AND PRINT FIXED END FORCES IN LOCAL COORDINATES
C
      IF(NUMFIX .EQ. 0) GO TO 56
      WRITE (5,2010)
      DO 55 I=1,NUMFIX
      READ  (5,1005) N,(SFT(N,J),J=1,12)
   55 WRITE (6,2011) N,(SFT(N,J),J=1,12)
   56 CONTINUE
C
C     READ AND PRINT ELEMENT DATA. GENERATE MISSING INPUT.
C
      WRITE (6,4000)
      L=0
   50 KKK=0
      READ (5,3000) INEL,INI,INJ,INK,IMAT,IMEL,ILC,INELKI,INELKJ,INC
      IF (INEL.NE.1) GO TO 15
      NI=INI
      NJ=INJ
      NK=INK
   15 IF (INC.EQ.0) INC=1
   65 L=L+1
      KKK=KKK+1
      ML=INEL-L
      IF (ML)  66,67,68
   66 WRITE (6,4003) INEL
      CALL EXIT
   67 NEL=INEL
       NI    =INI
       NJ    =INJ
      NK=INK
       MATTYP=IMAT
       MELTYP=IMEL
      DO 90 I=1,4
   90 LC(I)=ILC(I)
       NLOAD=LC(1)+LC(2)+LC(3)+LC(4)
       NEKODI=INELKI
       NEKODJ=INELKJ
      DO 91 I=1,3
   91 T(2,I)=TI(2,I)
      GO TO 69
   68 NEL=INEL-ML
       NI    =IN+KKK*INCR
       NJ    =JN+KKK*INCR
   69 CONTINUE
      WRITE (6,4001) NEL,NI,NJ,NK,MATTYP,MELTYP,LC,NEKODI,NEKODJ
C
   74 DX=X(NJ)-X(NI)
      DY=Y(NJ)-Y(NI)
      DZ=Z(NJ)-Z(NI)
      DL=SQRT(DX*DX+DY*DY+DZ*DZ)
      IF(DL) 75,75,76
   75 WRITE (6,4005) NEL
      CALL EXIT
C
C     FORM GLOBAL TO LOCAL COORDINATE TRANSFORMATION.
C
   76 T(1,1)=DX/DL
      T(1,2)=DY/DL
      T(1,3)=DZ/DL
C
C     COMPUTE DIRECTION COSINES OF LOCAL Y-AXIS
C
      A1=X(NJ)-X(NI)
      A2=Y(NJ)-Y(NI)
```

```
      A3=Z(NJ)-Z(NI)
      B1=X(NK)-X(NI)
      B2=Y(NK)-Y(NI)
      B3=Z(NK)-Z(NI)
      AA=A1*A1+A2*A2+A3*A3
      AB=A1*B1+A2*B2+A3*B3
      U1=AA*B1-AB*A1
      U2=AA*B2-AB*A2
      U3=AA*B3-AB*A3
      UU=U1*U1+U2*U2+U3*U3
      UU=SQRT(UU)
      IF (UU.GT.0.) GO TO 40
      WRITE (6,4002) INEL
      STOP
   40 T(2,1)=U1/UU
      T(2,2)=U2/UU
      T(2,3)=U3/UU
      T(3,1)=T(1,2)*T(2,3)-T(1,3)*T(2,2)
      T(3,2)=T(1,3)*T(2,1)-T(1,1)*T(2,3)
      T(3,3)=T(1,1)*T(2,2)-T(1,2)*T(2,1)
C
C     CHECK IF NEW STIFFNESS NEEDED
C
      IF (NEL.GE.1) GO TO 80
      IF  (ABS(DS-DL).GT.DL/100.)  GO TO 80
      IF ((MT.NE.MATTYP).OR.(MF.NE.MELTYP)) GO TO 80
      IF ((JK(1).NE.NEKODI).OR.(JK(2).NE.NEKODJ)) GO TO 80
      DO 81 I=1,4
      IF (LS(I).NE.LC(I)) GO TO 80
   81 CONTINUE
      DO 82 I=1,2
      DO 82 J=1,2
      IF (ABS(TS(I,J)-T(I,J)).GT.ABS(T(I,J)/100.)) GO TO 80
   82 CONTINUE
      GO TO 185
C
   80 DS=DL
      MT=MATTYP
      ME=MELTYP
      DO 77 I=1,2
      DO 77 J=1,2
   77 TS(I,J)=T(I,J)
      DO 78 I=1,4
   78 LS(I)=LC(I)
      JK(1)=NEKODI
      JK(2)=NEKODJ
C
C     FORM NEW STIFFNESS
C
      CALL NEWBM (E,G,RO,COPROP,SFT,NUMFIX,NUMETP)
C
C     ADD GRAVITY LOADING ... POINT LOADS ONLY COMPUTED
C
      DO 180 I=1,3
      DO 180 J=1,4
      RF(I,J)=RF(I,J)+EMUL(I,J)*XM(I)
  180 RF(I+6,J)=RF(I+6,J)+EMUL(I,J)*XM(I+6)
C
C     FORM ELEMENT LOCATION MATRIX
C
  185 CONTINUE
      DO 170 M=1,6
      LM(M)=ID(NI,M)
      LM(M+12)=0
      LM(M+18)=0
  170 LM(M+6)=ID(NJ,M)
C
      NS=12
      ND=12
C
C     TRANSFORM TO MASTER DEGREES OF FREEDOM
C
      CALL SLAVE (X,Y,Z,ID,NUMNP,NI,NJ)
C
C     WRITE ELEMENT INFORMATION ON TAPE
C
      NDM=24
      CALL CALBAN (MBAND,NDIF,LM,XM,ASA,RF,ND,NDM)
      WRITE (1) ND,NS,(LM(I),I=1,ND),((SA(I,J),I=1,NS),J=1,ND),
     1 ((SF(I,J),I=1,NS),J=1,4)
C
C        CHECK FOR LAST ELEMENT
C
        IF (NBEAM-NEL)  66,500,260
  260 CONTINUE
      IF (ML.GT.0)  GO TO 65
       IN      =INI
       JN      =INJ
      INCR=INC
      GO TO 60
  500 RETURN
C
 1001 FORMAT(I5,3F10.0)
 1002 FORMAT(I5,6F10.0)
 1005 FORMAT(I5,6F10.0/F15.0,5F10.0)
 1006 FORMAT (4F10.0)
 2001 FORMAT(55H1MATERIAL     YOUNG S      POISSON S            MASS
     1      /55H              MODULUS        RATIO            DENSITY       )
 2002 FORMAT(1H ,I5,3X,F12.0,F14.5,F14.5)
 2003 FORMAT(1H1/
     .  26H BEAM GEOMETRIC PROPERTIES//
     1   49H ELEMENT        AREA         AREA           AREA
     2  60H INERTIA      INERTIA      INERTIA
     3 / 49H  TYPE                X            Y              Z
     4                   30H      X            Y              Z              )
 2004 FORMAT(1H ,I5,2X,6F12.3)
 2005 FORMAT(39H1.......THREE DIMENSIONAL BEAM ELEMENTS//
     .  36H   NUMBER OF BEAMS                   =,I5/
     .  36H   NUMBER OF GEOMETRIC PROPERTY SETS=,I5/
     .  36H   NUMBER OF FIXED END FORCE SETS    =,I5/
     .  36H   NUMBER OF MATERIALS               =,I5)
 2006 FORMAT(///25H ELEMENT LOAD MULTIPLIERS / 20X,1HA,14X,1HB,14X,1HC,
     1 14X,1HD,/6H X-DIR4E15.6/ 6H Y-DIR4E15.6/ 6H Z-DIR4E15.6/ )
 2010 FORMAT(1H1,1H ,
     1  30X40H FIXED END FORCES IN LOCAL COORDINATES
     2//53H TYPE    NODE         FORCE X      FORCE Y      FORCE Z
     3                    35H MOMENT X      MOMENT Y     MOMENT Z             )
 2011 FORMAT(1H ,I3,6X,1HI,3X,6F12.3/1H ,9X,1HJ,3X,6F12.3/)
 2013 FORMAT(1H0/
     1  60H SECTION PROPERTIES OTHER THAN SHEAR AREAS MAY NOT BE SPECIF
     2  34HIED AS ZERO. EXECUTION TERMINATED.)
 3000 FORMAT (10I5,2I6,I8)
 4000 FORMAT(1H1/
     . 5HOBEAM 5X 5HNODES 5X 5H MATL 5H GEOM 5X 10HELEM LOADS 4X    10X
     . 12H  END CODES / 5H  ND      4X 1HI 4X 1HJ 4X 1HK 5H  NO  5H  NO
     .  4X 1HA 4X 1HB 4X 1HC 4X 1HD 9X 1HI 9X 1HJ)
 4001 FORMAT (10I5,2I10)
 4002 FORMAT (9HOBEAM NO ,I5, 26H      K NODE ON BEAM X-AXIS      ,
     .  26H......EXECUTION TERMINATED )
 4003 FORMAT(36HOELEMENT CARD ERROR, ELEMENT NUMBER= I6)
 4004 FORMAT(1H ,31HNODAL POINT NUMBERS FOR ELEMENT,I5,36HARE IDENTICAL.
     1 EXECUTION TERMINATED.)
 4005 FORMAT(8HOELEMENT,I5,39H HAS ZERO LENGTH. EXECUTION TERMINATED.)
      END
```

```
      SUBROUTINE NEWBM(E,G,RO,COPROP,SFT,NUMFIX,NUMETP)
C
C     FORM NEW BEAM STIFFNESS
C
      DIMENSION E(1),G(1),RO(1),COPROP(NUMETP,1),SFT(NUMFIX,1)
      COMMON/EM/LM(24),ND,NS,ASA(24,24),RF(24,4),XM(24),SA(12,24),
     .     SF(12,4)
      COMMON /NEWB/ LC(4),T(3,3),JK(6),MELTYP,DL,MATTYP
      DIMENSION R(12),S(12,12),C(12)
```

```
C
      DO 5 I=1,144
    5 S(I)=0.
      AX=COPROP(MELTYP,1)
       AY=COPROP(MELTYP,2)
       AZ=COPROP(MELTYP,3)
      AAX=COPROP(MELTYP,4)
      AAY=COPROP(MELTYP,5)
      AAZ=COPROP(MELTYP,6)
      SHFY=0.0
      SHFZ=0.0
      ZY=E(MATTYP)/(DL*DL)
      EIY=ZY*AAY
      EIZ=ZY*AAZ
      IF(AY.NE.0.0) SHFY=6.*EIZ/(G(MATTYP)*AY)
      IF(AZ.NE.0.0) SHFZ=6.*EIY/(G(MATTYP)*AZ)
      COMMY=EIY/(1.+2.*SHFZ)
      COMMZ=EIZ/(1.+2.*SHFY)
C
C         FIXED END FORCES IN LOCAL COORDS
C
      DO 73 N=1,4
      M=LC(N)
      IF (M.GT.0)  GO TO 71
      DO 70 I=1,12
   70 SF(I,N)=0.
      GO TO 73
   71 DO 72 I=1,12
   72 SF(I,N)=SFT(M,I)
   73 CONTINUE
C
C     FORM ELEMENT STIFFNESS IN LOCAL COORDINATES
C
      S(1,1)= E(MATTYP)* AX/DL
      S(4,4)= G(MATTYP)*AAX/DL
      S(2,2)= COMMZ*12./DL
      S(3,3)= COMMY*12./DL
      S(5,5)= COMMY* 4.*DL*(1.+0.5*SHFZ)
      S(6,6)= COMMZ* 4.*DL*(1.+0.5*SHFY)
      S(2,6)= COMMZ* 6.
      S(3,5)=-COMMY* 6.
      DO 102 I=1,6
      J=I+6
  102 S(J,J)=S(I,I)
      DO 104 I=1,4
      J=I+6
  104 S(I,J)=-S(I,I)
      S(6,12)= S(6,6)*(1.-SHFY)/(2.+SHFY)
      S(5,11)= S(5,5)*(1.-SHFZ)/(2.+SHFZ)
      S(2,12)= S(2,6)
      S(6, 8)=-S(2,6)
      S(8,12)=-S(2,6)
      S(3,11)= S(3,5)
      S(5, 9)=-S(3,5)
      S(9,11)=-S(3,5)
      DO 106 I=2,12
      K=I-1
      DO 106 J=1,K
  106 S(I,J)=S(J,I)
C
C     MODIFY ELEMENT STIFFNESS AND ELEMENT FIXED END FORCES FOR KNOWN
C     ZERO MEMBER END FORCES.
C
      IF ((JK(1)+JK(2)).EQ.0) GO TO 145
      DO 140 K=1,2
      KK=JK(K)
      KD=100000
      I1=6*(K-1)+1
      I2=I1+5
      DO 140 I=I1,I2
      IF (KK.LT.KD)  GO TO 140
      SII=S(I,I)
      DO 125 N=1,12
  125 R(N)=S(I,N)
      DO 130 M=1,12
      C(M)=S(M,I)/SII
      DO 130 N=1,12
  130 S(M,N)=S(M,N)-C(M)*R(N)
      DO 135 N=1,4
      SFI=SF(I,N)
      DO 135 M=1,12
  135 SF(M,N)=SF(M,N)-C(M)*SFI
  136 KK=KK-KD
  140 KD=KD/10
  145 CONTINUE
C
C     OBTAIN SA(12,12) RELATING ELEMENT END FORCES (LOCAL) AND
C     JOINT DISPLACEMENTS (GLOBAL).
C
      DO 31 I=1,288
   31 SA(I)=0.
      DO 150 LA=1,10,3
      LB=LA+2
      DO 150 MA=1,10,3
      MB=MA-1
      DO 150 I=LA,LB
      DO 150 JM=1,3
      J=JM+MB
      XX=0.
      DO 151 K=1,3
  151 XX=XX+S(I,K+MB)*T(K,JM)
  150 SA(I,J)=XX
C
C     ELEM STIFF ASA(12,12) AND FIXED END FORCES RF(12) IN GLOBAL COORDS
C
      DO 32 I=1,576
   32 ASA(I)=0.
      DO 160 LA=1,10,3
      LB=LA-1
      DO 160 MA=1,10,3
      MB=MA+2
      DO 160 IL=1,3
      I=IL+LB
      DO 160 J=MA,MB
      XX=0.
      DO 161 K=1,3
  161 XX=XX+T(K,IL)*SA(K+LB,J)
  160 ASA(I,J)=XX
C
      DO 165 LA=1,10,3
      LB=LA-1
      DO 165 IL=1,3
      I=IL+LB
      DO 165 N=1,4
      XX=0.
      DO 162 K=1,3
  162 XX=XX-T(K,IL)*SF(K+LB,N)
  165 RF(I,N)=XX
C
C     FORM MASS MATRIX
C
      XXM=RO(MATTYP)*AX*DL/2.
      DO 190 M=1,3
      XM(M)=XXM
      XM(M+3)=0.
      XM(M+9)=0.
  190 XM(M+6)=XXM
      RETURN
      END
```

```
      SUBROUTINE SLAVE (X,Y,Z,ID,NUMNP,NI,NJ)
C
C     PERFORMS SLAVE...MASTER DISPLACEMENT TRANSFORMATION
C       ( FOR NODES CONNECTED TO BEAM ELEMENTS ONLY)
C
      DIMENSION X(1),Y(1),Z(1),ID(NUMNP,1)
      COMMON /EM/ LM(24),ND,NS,S(24,24),R(96),XM(24),SA(12,24) ,TT(12,4)
C
C     DETERMINE REQUIRED TRANSLATION DEGREES OF FREEDOM
C
      DO 54 NF=1,12,6
      NOD=NI
      IF (NF.EQ.7) NOD=NJ
      DO 30 K=1,3
      I=K+NF-1
      IF (LM(I).GE.0) GO TO 30
      M=-LM(I)
      LM(I)=ID(M,K)
      IF(K-2) 35,45,55
   35 D1=-(Y(NOD)-Y(M))
      D2=   Z(NOD)-Z(M)
      LM(ND+1)=ID(M,6)
      LM(ND+2)=ID(M,5)
      GO TO 50
   45 D1=-(Z(NOD)-Z(M))
      D2=   X(NOD)-X(M)
      LM(ND+1)=ID(M,4)
      LM(ND+2)=ID(M,6)
      GO TO 50
   55 D1=-(X(NOD)-X(M))
      D2=   Y(NOD)-Y(M)
      LM(ND+1)=ID(M,5)
      LM(ND+2)=ID(M,4)
   50 CONTINUE
C
C     TRANSFORMATION...ARRAYS INCREASE IN SIZE
C
      DO 60 II=1,ND
      S(ND+1,II)=S(I,II)*D1
      S(ND+2,II)=S(I,II)*D2
      XM(ND+1)=XM(I)*D1*D1
      XM(ND+2)=XM(I)*D2*D2
      S(II,ND+1)=S(II,I)*D1
      S(II,ND+2)=S(II,I)*D2
   60 CONTINUE
C
      DO 70 II=1,NS
      SA(II,ND+1)=SA(II,I)*D1
   70 SA(II,ND+2)=SA(II,I)*D2
C
      S(ND+1,ND+1)=S(I,I)*D1**2
      S(ND+2,ND+2)=S(I,I)*D2**2
      S(ND+1,ND+2)=S(I,I)*D1*D2
      S(ND+2,ND+1)=S(ND+1,ND+2)
      ND=ND+2
   30 CONTINUE
C
C     SET ROTATIONS
C
      DO 54 J=1,3
      K=NF+J+2
      IF(LM(K).GE.0)   GO TO 54
      M=-LM(K)
      LM(K)=ID(M,J+3)
   54 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE PLANE
      COMMON A(1)
      COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
      COMMON /EM/ NS,ND,LM(48),B(48,48),TT(48,4)
      COMMON /JUNK/ LT,LH,L,SG(20),SIG(7),EXTRA(150)
      DIMENSION STRLAB(5)
      DATA STRLAB/3HCEN,3HL-I,3HJ-K,3HI-J,3HK-L/
      IF(NPAR(1).EQ.0) GO TO 200
      IF(NPAR(1).EQ.3) NPAR(5)=2
      IF(NPAR(5).EQ.0) WRITE (6,2000)
      IF(NPAR(5).EQ.1) WRITE (6,2001)
      IF(NPAR(5).EQ.2) WRITE (6,2002)
      IF (NPAR(1).EQ.3) WRITE (6,2003)
      IF (NPAR(6).NE.0) WRITE (6,2004)
      IF(NPAR(4).EQ.0) NPAR(4)=1
      N6=N5+NUMNP
      N7=N6+NPAR(3)
      N8=N7+NPAR(3)
      N9=N8+NPAR(3)
      N10=N9+NPAR(3)
      MM=N10+11*NPAR(4)*NPAR(3)-MTOT
      IF(MM.GT.0) CALL ERROR(MM)
      CALL PLNAX(A(N1),A(N2),A(N3),A(N4),A(N5),A(N6),A(N7),A(N8),A(N9),
     1A(N10),  NPAR(4),NUMNP)
      RETURN
C
  200 NUME=NPAR(2)
      IF (NPAR(5).EQ.0) WRITE(6,2000)
      IF (NPAR(5).EQ.1) WRITE(6,2001)
      IF (NPAR(5).EQ.2) WRITE(6,2002)
      WRITE(6,2005)
      DO 800 MM=1,NUME
      CALL STRSC(A(N1),A(N3),NEQ,0)
      IF(NS.EQ.1) GO TO 800
      WRITE (6,3000) MM
      DO 700 L=LT,LH
      CALL STRSC(A(N1),A(N3),NEQ,1)
      ITAG=0
  510 DO 600 KK=1,NS,4
      ITAG=ITAG+1
      DO 520 I=1,4
      II=KK-1+I
  520 SIG(I)=SG(II)
      CC=(SIG(1)+SIG(2))/2.0
      BB=(SIG(1)-SIG(2))/2.
      CR=SQRT(BB**2+SIG(4)**2)
      SIG(5)=CC+CR
      SIG(6)=CC-CR
      SIG(7)=0.0
      IF ((BB.EQ.0.0).AND.(SIG(4).EQ.0.0)) GO TO 600
      SIG(7)=28.648*ATAN2(SIG(4),BB)
  600 WRITE (6,3001) L,STRLAB(ITAG),(SIG(I),I=1,7)
  700 CONTINUE
  800 CONTINUE
      RETURN
 2000 FORMAT (22H1AXISYMMETRIC ANALYSIS )
 2001 FORMAT (22H1PLANE STRAIN ANALYSIS )
 2002 FORMAT (22H1PLANE STRESS ANALYSIS )
 2003 FORMAT (18H MEMBRANE ELEMENTS )
 2004 FORMAT (30H INCOMPATIBLE MODES SUPPRESSED )
 2005 FORMAT(14H0STRESS OUTPUT/)
 3000 FORMAT(1X,14HELEMENT NUMBER,I5,5X,84H(CENTER STRESSES IN LOCAL Y-Z
     . COORDS,BOUNDARY STRESSES NORMAL AND PARALLEL TO SIDES),
     .        /1X,5H LOAD,17X,3HS11,12X,3HS22,12X,3HS33,12X,3HS12,10X,
     . 5HS-MAX,10X,5HS-MIN,5X,5HANGLE)
 3001 FORMAT(1X,I5,2X,A3,1P6E15.6, 0P 1F10.3)
      END
```

```fortran
      SUBROUTINE PLNAX(ID,X,Y,Z,T,NTC,WT,RO,WANG,E,NUMTC,NUMNP)
      DIMENSION X(1),Y(1),Z(1),ID(NUMNP,1),NTC(1),WT(1),RO(1),WANG(1),
     .  E(NUMTC,11,1),T(1)
      COMMON /ELPAR/ NPAR(14),NUMNN,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
      COMMON /EM/ LM(12),S(12,12),P(12,4),XM(12),B(20,12),BB(20,12),
     1 TI(20,4),IX(4),IE(5),D(4,4),EMUL(4,5),RR(4),ZZ(4),H(6),HS(6),
     2 HT(6),HR(6),HZ(6),FAC,XMM,PRESS,NS,EE(10),TTI(4),PP(12,4),THICK
     3 ,TMP(4),TP(12),ALP(4)
      COMMON /JUNK/ MAT,NT,TEMP,REFT,BETA,U(4),V(4),W(4),G(4)
C
      NUME=NPAR(2)
      NUMMAT=NPAR(3)
      WRITE (6,2000) NUME,NUMMAT,NUMTC
C
C     READ AND PRINT OF MATERIAL PROPERTIES
      DO 60 M=1,NUMMAT
      READ  (5,1010) MAT,NTC(MAT),WT(MAT),RO(MAT),WANG(MAT)
        IF (NTC(MAT).EQ.0)  NTC(MAT)=1
      WRITE (6,2020) MAT,NTC(MAT),WT(MAT),RO(MAT),WANG(MAT)
      NT=NTC(MAT)
      READ  (5,1005) ((E(I,J,MAT),J=1,11),I=1,NT)
      WRITE (6,2010) ((E(I,J,MAT),J=1,11),I=1,NT)
      IF ( NPAR(5) .NE. 2 ) GO TO 60
   60 CONTINUE
      READ (5,1002)  ((EMUL(I,J),J=1,5),I=1,4)
      IF (NPAR(1).EQ.3) WRITE (6,2004) ((EMUL(I,J),J=1,5),I=1,4)
      IF (NPAR(1).EQ.4) WRITE (6,2005) ((EMUL(I,J),J=1,5),I=1,4)
C
C       READ AND PRINT OF ELEMENT PROPERTIES
C
      WRITE (6,2002)
      N=0
  130 READ(5,1003) M,(IE(I),I=1,5),REFT,PRESS,NS,KG,THICK
      MAT=IE(5)
      IF(KG.EQ.0) KG=1
      IF (NPAR(5).EQ.1) THICK=1.0
      IF(NS.EQ.0) NS=4
      IF(NS.LT.4) NS=1
      IF( (IE(3) .EQ. IE(4)) .AND. (NS.EQ. 20) ) NS=16
  140 N=N+1
      IF(M.EQ.N) GO TO 145
      DO 142 I=1,4
  142 IX(I)=IX(I)+KG
      GO TO 149
  145 DO 148 I=1,4
  148 IX(I)=IE(I)
C
C        FORM CONSTITUTIVE LAW AND COMPUTE THERMAL STRESSES
C
  149 NT=NTC(MAT)
      WRITE (6,2003) N,IX,MAT,REFT,PRESS,NS,KG,THICK
      I=IX(1)
      J=IX(2)
      K=IX(3)
      L=IX(4)
      TEMP = (T(I)+T(J)+T(K)+T(L))/4.0
      BETA=WANG(MAT)
      XMM=RO(MAT)
      WGT=WT(MAT)
      CALL ELAW (NUMTC,EE,E,D,TTI,ALP)
C
C     CALCULATE ELEMENT STIFFNESS MATRIX
C
      IF(NPAR(1).EQ.3) GO TO 160
      ND=8
      DO 155 I=1,4
      II=IX(I)
      RR(I)=Y(II)
      ZZ(I)=Z(II)
      TMP(I)=T(II)
      LM(I)=ID(II,2)
  155 LM(I+4)=ID(II,3)
      CALL QUAD
      DO 158 I=1,4
      DO 157 L=1,4
      P(I,L)=P(I,L)+XM(I)*RO(MAT)*RR(I)*EMUL(L,3)**2
      P(I,L)=P(I,L)+XM(I)*WGT*EMUL(L,4)
  157 P(I+4,L)=P(I+4,L)+XM(I)*WGT*EMUL(L,5)
      XM(I)=XM(I)*XMM
  158 XM(I+4)=XM(I)
      GO TO 300
C
  160 ND=12
      CALL VECTOR(V,X(I),Y(I),Z(I),X(J),Y(J),Z(J))
      CALL VECTOR(G,X(I),Y(I),Z(I),X(L),Y(L),Z(L))
      CALL CROSS(V,G,W)
      CALL CROSS(W,V,U)
      CALL VECTOR(W,X(I),Y(I),Z(I),X(K),Y(K),Z(K))
      RR(1)=0.0
      ZZ(1)=0.0
      RR(2)=V(4)
      ZZ(2)=0.0
      RR(3)=W(4)*DOT(W,V)
      ZZ(3)=W(4)*DOT(W,U)
      RR(4)=G(4)*DOT(G,V)
      ZZ(4)=G(4)*DOT(G,U)
C
      DO 170 I=1,4
      II=IX(I)
      TMP(I)=T(II)
      LM(I)=ID(II,1)
      LM(I+4)=ID(II,2)
  170 LM(I+8)=ID(II,3)
      CALL QUAD
C
      DO 190 I=1,3
      DO 190 K=1,4
      KK=4*(I-1)+K
      DO 180 L=1,4
  180 PP(KK,L)=V(I)*P(K,L)+U(I)*P(K+4,L)
      DO 190 J=I,3
      DO 190 L=1,4
      LL=4*(J-1)+L
  190 BB(KK,LL)=V(I)*(S(K,L)*V(J)+S(K,L+4)*U(J))
     1 +U(I)*(S(K+4,L)*V(J)+S(K+4,L+4)*U(J))
C
      DO 196 I=1,12
      DO 194 L=1,4
  194 P(I,L)=PP(I,L)
      DO 196 J=I,12
      S(I,J)=BB(I,J)
  196 S(J,I)=S(I,J)
C
      DO 210 K=1,NS
      DO 200 L=1,4
      DO 200 J=1,3
      LL=4*(J-1)+L
  200 BB(K,LL)=B(K,L)*V(J)+B(K,L+4)*U(J)
      DO 210 J=1,12
  210 B(K,J)=BB(K,J)
C
      DO 220 I=1,4
      DO 215 L=1,4
      P(I  ,L)=P(I  ,L)+XM(I)*WGT*EMUL(L,3)
      P(I+4,L)=P(I+4,L)+XM(I)*WGT*EMUL(L,4)
  215 P(I+8,L)=P(I+8,L)+XM(I)*WGT*EMUL(L,5)
      XM(I)=XM(I)*XMM
      XM(I+4)=XM(I)
  220 XM(I+8)=XM(I)
C
C     CALCULATION OF BAND WIDTH AND WRITES ELEMENT MATRICES ON TAPES
C
  300 CALL CALBAN(MBAND,NDIF,LM,XM,S,P,ND,12)
      WRITE (1) ND,NS,(LM(I),I=1,ND),(( B(I,J),I=1,NS),J=1,ND),
     1 ((TTI(I,J),I=1,NS),J=1,4)
      IF(N.EQ.NUME) RETURN
      IF(N.EQ.M) GO TO 130
      GO TO 140
C
 1002 FORMAT (5F10.0)
```

```
1003 FORMAT (6I5,2F10.0,2I5,F10.0)
1005 FORMAT (8F10.0/3F10.0)
1010 FORMAT (2I5,3F10.0)
2000 FORMAT (34HONUMBER OF ELEMENTS                    = I5/
    1        34HONUMBER OF MATERIALS                   = I5/
    2        34HOMAXIMUM NUMBER OF TEMPERATURES  = I5)
2002 FORMAT ( /50H2   EL.NO.    I     J     K     L TYPE     TEMPERATURE
    1 ,7X,8HPRESSURE ,14H  NO. STRESSES  20H    KG      THICKNESS    )
2003 FORMAT (I10,5I5,2E15.6,2I10,E15.5)
2004 FORMAT(23H2ELEMENT LOAD FRACTIONS /70H LOAD CASE TEMPERATURE  PRES
    1SURE   X-DIRECTION Y-DIRECTION Z-DIRECTION /9X 1HA 5F12.3/
    2 9X 1HB 5F12.3/ 9X 1HC 5F12.3/ 9X 1HD 5F12.3)
2005 FORMAT(23H2ELEMENT LOAD FRACTIONS /70H LOAD CASE TEMPERATURE  PRES
    1SURE  ANG.VELOCITY Y-DIRECTION Z-DIRECTION /9X 1HA 5F12.3/
    2 9X 1HB 5F12.3/ 9X 1HC 5F12.3/ 9X 1HD 5F12.3)
2010 FORMAT (15HO  TEMPERATURE 11X 4HE(N),11X,4HE(S),11X,4HE(T),
    1 9X,5HNU(NS),9X,5HNU(NT), 9X,6HNU(ST),10X,5HG(NS)/26X,4HA(N),
    2 11X,4HA(S),11X,4HA(T),
    3 (/F15.2,3F15.2,3F15.4,F15.2/22X,3F15.9))
2020 FORMAT (////16H MATERIAL NUMBER I3//
    1 30H NUMBER OF TEMPERATURE CARDS =I3,5X,15H WGT. DENSITY =E12.4,
    2 15H MASS DENSITY =E12.4,13H ANGLE BETA =F6.1)
     END
```

```
      SUBROUTINE QUAD
      COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
      COMMON /EM/ LM(12),S(12,12),P(12,4),XM(12),B(20,12),BB(20,12),
     1 TI(20,4),IX(4),IE(5),D(4,4),EMUL(4,5),RR(4),ZZ(4),H(6),HS(6),
     2 HT(6),HR(6),HZ(6),FAC,XMM,PRESS,NS,EE(10),TTI(4),PP(12,4),THICK
     3 ,TMP(4),TP(12),ALP(4)
      COMMON /JUNK/ MAT,NT,TEMP,REFT,BETA
      DIMENSION SS(2),TT(2),HH(2),SSS(5),TTT(5),IVECT(4),JVECT(4),V(4)
      DATA SSS/0.,-1.,1.,0.,0./, TTT/0.,0.,0.,-1.,1./
      DATA SS/-0.57735026918963,0.57735026918963/
      DATA TT/-0.57735026918963,0.57735026918963/
      DATA HH/1.0,1.0/,IVECT/4,2,1,3/,JVECT/1,3,2,4/
C
      DO 170 J=1,12
      XM(J)=0.0
      TP(J)=0.
      DO 160 I=1,20
      BB(I,J)=0.0
  160 B(I,J)=0.0
      DO 170 I=1,12
  170 S(I,J)=0.0
C
      DO 500 II=1,2
      DO 500 JJ=1,2
      CALL FORMB(SS(II),SS(JJ),B)
      TEMP=0.
      DO 200 I=1,4
  200 TEMP=TEMP+H(I)*TMP(I)
      FAK=FAC*HH(JJ)*HH(II)
      FTP=TEMP-REFT
      DO 400 J=1,12
      D1=(D(1,1)*B(1,J)+D(1,2)*B(2,J)+D(1,3)*B(3,J)+D(1,4)*B(4,J))*FAK
      D2=(D(2,1)*B(1,J)+D(2,2)*B(2,J)+D(2,3)*B(3,J)+D(2,4)*B(4,J))*FAK
      D3=(D(3,1)*B(1,J)+D(3,2)*B(2,J)+D(3,3)*B(3,J)+D(3,4)*B(4,J))*FAK
      D4=(D(4,1)*B(1,J)+D(4,2)*B(2,J)+D(4,3)*B(3,J)+D(4,4)*B(4,J))*FAK
      TP(J)=TP(J)+FTP*(D1*ALP(1)+D2*ALP(2)+D3*ALP(3)+D4*ALP(4))
      DO 400 I=J,12
      S(I,J)=S(I,J)+B(1,I)*D1+B(2,I)*D2+B(3,I)*D3+B(4,I)*D4
  400 S(J,I)=S(I,J)
      DO 450 I=1,4
  450 XM(I)=XM(I)+FAK*H(I)
  500 CONTINUE
C
C     FORM STRESS DISPLACEMENT MATRIX
C
      LL=NS/4
      DO 530 L=1,LL
      CALL FORMB(SSS(L),TTT(L),BB)
C
      TEMP=0.
      DO 515 K=1,4
  515 TEMP=TEMP+H(K)*TMP(K)
      FTP=TEMP-REFT
      DO 530II=1,4
      I=II+4*(L-1)
      TI(I,4)=-TTI(II)*FTP
      DO 530 J=1,12
      B(I,J)=0.0
      DO 530 K=1,4
  530 B(I,J)=B(I,J)+D(II,K)*BB(K,J)
C
C     ELIMINATE EXTRA DEGREES OF FREEDOM
C
      IF ( IX(3) .EQ. IX(4) ) GO TO 560
      IF(NPAR(6).NE.0) GO TO 560
      DO 550 NN=1,4
      L=12-NN
      K=L+1
      C=TP(K)/S(K,K)
      DO 535 J=1,NS
  535 TI(J,4)=TI(J,4)+C*B(J,K)
      DO 550 I=1,L
      C=S(I,K)/S(K,K)
      TP(I)=TP(I)-C*TP(K)
      DO 540 J=1,NS
  540 B(J,I)=B(J,I)-C*B(J,K)
      DO 550 J=1,L
  550 S(I,J)=S(I,J)-C*S(K,J)
C
C     ROTATE STRESS-DISPLACEMENT TRANSFORMATION TO GIVE STRESSES
C     NORMAL AND PARALLEL TO SIDES - SIMILARILY ROTATE INITIAL STRESSES
C
  560 NSET=LL-1
      IF ( NSET .LE. 0 ) GO TO 730
      DO 720 L=1,NSET
      IV=IVECT(L)
      JV=JVECT(L)
      CALL VECTOR(V,RR(IV),ZZ(IV),0.0,RR(JV),ZZ(JV),0.0)
      S2=V(1)*V(1)
      C2=V(2)*V(2)
      SC=-V(1)*V(2)
      I1=4*L+1
      I2=I1+1
      I4=I1+3
      T1=TI(I1,4)
      T2=TI(I2,4)
      T4=TI(I4,4)
      T5=2.0*SC*T4
      TI(I1,4)=C2*T1+S2*T2+T5
      TI(I2,4)=S2*T1+C2*T2-T5
      TI(I4,4)=SC*(T2-T1)+(C2-S2)*T4
      DO 710 J=1,8
      B1=B(I1,J)
      B2=B(I2,J)
      B4=B(I4,J)
      B5=2.0*SC*B4
      B(I1,J)=C2*B1+S2*B2+B5
      B(I2,J)=S2*B1+C2*B2-B5
  710 B(I4,J)=SC*(B2-B1)+(C2-S2)*B4
  720 CONTINUE
  730 CONTINUE
C
      DO 660 L=1,4
      DO 600 I=1,NS
  600 TI(I,L)=TI(I,4)*EMUL(L,1)
      DO 660 I=1,8
  660 P(I,L)=TP(I)*EMUL(L,1)
C
C     CALCULATE PRESSURE LOADS ON I-J FACE
C
```

```
      DR=RR(2)-RR(1)
      DZ=ZZ(1)-ZZ(2)
      RI=PRESS*(2.*RR(1)+RR(2))/6.
      RJ=PRESS*(2.*RR(2)+RR(1))/6.
      IF(NPAR(5).EQ.0) GO TO 670
      RI=PRESS*THICK/2.
      RJ=RI
  670 DO 700 L=1,4
      P(1,L)=P(1,L)+DZ*RI*EMUL(L,2)
      P(5,L)=P(5,L)+DR*RI*EMUL(L,2)
      P(2,L)=P(2,L)+DZ*RJ*EMUL(L,2)
  700 P(6,L)=P(6,L)+DR*RJ*EMUL(L,2)
      RETURN
      END
```

```
      SUBROUTINE FORMB(S,T,B)
      COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
      COMMON /EM/ LM(12),U(12,12),P(12,4),XM(12),Q(20,12),BB(20,12),
     1 TI(20,4),IX(4),IE(5),D(4,4),EMUL(4,5),RR(4),ZZ(4),H(6),HS(6),
     2 HT(6),HR(6),HZ(6),FAC,XMM,PRESS,NS,EE(10),TTI(4),PP(12,4),THICK
     3 ,TMP(4),QP(12),ALP(4)
      DIMENSION B(20,12)
      DIMENSION II(6),JJ(6)
      DATA II/1,2,3,4,9,10/,JJ/5,6,7,8,11,12/
C
      SM=1.0-S
      SP=1.0+S
      TM=1.0-T
      TP=1.0+T
C
      H(1)=SM*TM/4.
      H(2)=SP*TM/4.
      H(3)=SP*TP/4.
      H(4)=SM*TP/4.
      H(5)=(1.0-S*S)
      H(6)=(1.0-T*T)
C
      HS(1)=-TM/4.
      HS(2)=-HS(1)
      HS(3)=TP/4.
      HS(4)=-HS(3)
      HS(5)=-2.*S
      HS(6)=0.0
C
      HT(1)=-SM/4.
      HT(2)=-SP/4.
      HT(3)=-HT(2)
      HT(4)=-HT(1)
      HT(5)=0.0
      HT(6)=-2.*T
C
      PZT=HT(1)*ZZ(1)+HT(2)*ZZ(2)+HT(3)*ZZ(3)+HT(4)*ZZ(4)
      PZS=HS(1)*ZZ(1)+HS(2)*ZZ(2)+HS(3)*ZZ(3)+HS(4)*ZZ(4)
      PRS=HS(1)*RR(1)+HS(2)*RR(2)+HS(3)*RR(3)+HS(4)*RR(4)
      PRT=HT(1)*RR(1)+HT(2)*RR(2)+HT(3)*RR(3)+HT(4)*RR(4)
      XJ=PRS*PZT-PRT*PZS
C
      PSR=PZT/XJ
      PTR=-PZS/XJ
      PSZ=-PRT/XJ
      PTZ=PRS/XJ
C
      DO 100 I=1,6
      HR(I)=PSR*HS(I)+PTR*HT(I)
  100 HZ(I)=PSZ*HS(I)+PTZ*HT(I)
      R=H(1)*RR(1)+H(2)*RR(2)+H(3)*RR(3)+H(4)*RR(4)
      IF(NPAR(5).NE.0) R=THICK
C
C     FORM STRAIN DISPLACEMENT MATRIX
C
      DO 200 K=1,6
      I=II(K)
      J=JJ(K)
      B(1,I)=HR(K)
      B(2,J)=HZ(K)
      IF(NPAR(5).EQ.0) B(3,I)=H(K)/R
      B(4,I)=HZ(K)
  200 B(4,J)=HR(K)
C
      FAC=XJ*R
      RETURN
      END
```

```
      SUBROUTINE ELAW (NUMTC,EE,E,C,P,ALP)
      COMMON/JUNK/MAT,NT,TEMP,REFT,BETA,TAU(4),D(4,4),CC(4,4),XX(4)
      DIMENSION E(NUMTC,11,1),EE(10),C(4,4),P(4),ALP(4)
      COMMON/ELPAR/NPAR(14)
C
C          STRESS-STRAIN LAW IN  N-S-T  SYSTEM
        IF (NT.NE.1)  GO TO 220
      DO 210 KK=1,10
  210 EE(KK)=E(1,KK+1,MAT)
      GO TO 260
  220 DO 230 I=2,NT
      III=I
      T1=E(I-1,1,MAT)
      T2=E(I,1,MAT)
      IF(T2.GE.TEMP) GO TO 240
  230 CONTINUE
  240 CONTINUE
      RI=(T2-TEMP)/(T2-T1)
      RJ=(TEMP-T1)/(T2-T1)
      I=III
      DO 250 KK=1,10
  250 EE(KK)=E(I-1,KK+1,MAT)*RI+E(I,KK+1,MAT)*RJ
  260 CONTINUE
      DO 265 II=1,4
      DO 265 KK=1,4
      C(II,KK)=0.
  265 D(II,KK)=0.
      C(1,1)=1.0/EE(1)
      C(2,2)=1.0/EE(2)
      C(3,3)=1.0/EE(3)
      C(1,2)=-EE(4)/EE(2)
      C(1,3)=-EE(5)/EE(3)
      C(2,3)=-EE(6)/EE(3)
      C(2,1)=C(1,2)
      C(3,1)=C(1,3)
      C(3,2)=C(2,3)
      C(4,4)=1.0/EE(7)
      DO 270 I=1,3
  270 ALP(I)=EE(I+7)
      ALP(4)=0.0
C
C     ROTATE PROPERTIES TO R-Z-T SYSTEM
C
      IF (BETA.EQ.0.0) GO TO 500
      ANG=BETA/57.2957795
      SS=SIN(ANG)
      CC=COS(ANG)
      C2=CC*CC
      S2=SS*SS
      SC=SS*CC
C     SET D FOR SIG(O)=D*SIG(G)
      D(1,1)=C2
      D(1,2)=S2
      D(1,4)=2.*SC
      D(2,1)=S2
```

```
      D(2,2)=C2
      D(2,4)=-2.*SC
      D(3,3)=1.0
      D(4,1)=-SC
      D(4,2)=SC
      D(4,4)=C2-S2
C
      DO 300 I=1,4
      DO 300 J=1,4
      SUM=0.
      DO 280 M=1,4
  280 SUM=SUM+D(M,I)*C(M,J)
  300 CC(I,J)=SUM
      DO 350 I=1,4
      DO 350 J=I,4
      SUM=0.
      DO 330 M=1,4
  330 SUM=SUM+CC(I,M)*D(M,J)
      C(I,J)=SUM
  350 C(J,I)=SUM
C
      XX(1)=C2*ALP(1)+S2*ALP(2)
      XX(2)=S2*ALP(1)+C2*ALP(2)
      XX(3)=ALP(3)
      XX(4)=2.0*SC*(ALP(1)-ALP(2))
      DO 430 I=1,4
  430 ALP(I)=XX(I)
C
  500 CONTINUE
      CALL POSINV (C,4,4)
C     MODIFY FOR PLANE STRESS
      IF (NPAR(5).NE.2) GO TO 560
      C(1,1)=C(1,1)-C(3,1)*C(1,3)/C(3,3)
      C(1,2)=C(1,2)-C(3,2)*C(1,3)/C(3,3)
      C(1,4)=C(1,4)-C(3,4)*C(1,3)/C(3,3)
      C(2,2)=C(2,2)-C(3,2)*C(2,3)/C(3,3)
      C(2,4)=C(2,4)-C(3,4)*C(2,3)/C(3,3)
      C(4,4)=C(4,4)-C(3,4)*C(4,3)/C(3,3)
      DO 650 I=1,4
      DO 600 J=I,4
  600 C(J,I)=C(I,J)
      C(I,3)=0.
  550 C(3,I)=0.
  560 DO 670 I=1,4
      P(I)=0.0
      DO 670 M=1,4
  670 P(I)=P(I)+C(I,M)*ALP(M)
  700 RETURN
C
      END
```

```
      SUBROUTINE THREED
C
      COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
      COMMON / EM / NS,ND,LM(48),B(48,48),TT(48,4)
      EQUIVALENCE (IS1,TT(4)) , (IS2,TT(6))
      COMMON / JUNK / LT,LH,L,SIG(24)
      COMMON A(1)
      DIMENSION SPR(6)
C
      IF(NPAR(1).EQ.0) GO TO 500
      N6=N5+NUMNP
      N7=N6+NPAR(3)
      N8=N7+NPAR(3)
      N9=N8+NPAR(3)
      N10=N9+NPAR(3)
      N11=N10+NPAR(4)
      N12=N11+NPAR(4)
      N13=N12+NPAR(4)
      N14=N13+NPAR(4)
      N15=N14+33*33
      N16=N15+12*33
      IF(N16.GT.MTOT) CALL ERROR (N16-MTOT)
      CALL BRICK8 (A(N14),A(N15),
     .             NPAR(2),NPAR(3),NPAR(4),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),NUMNP)
      RETURN
C
  500 WRITE (6,2005)
      NUME=NPAR(2)
      DO 800 MM=1,NUME
      CALL STRSC (A(N1),A(N3),NEQ,0)
      WRITE (6,2000)
      DO 800 L=LT,LH
      CALL STRSC (A(N1),A(N3),NEQ,1)
      CALL PRIST (NS,IS1,IS2,SIG,SPR)
      WRITE (6,3005) MM,L,IS1,(SIG(I),I=1,6),(SPR(I),I=1,3)
      IF(NS.EQ.12) WRITE (6,3015) IS2,(SIG(I),I=7,12),(SPR(I),I=4,6)
  800 CONTINUE
C
      RETURN
C
 2000 FORMAT (/)
 2005 FORMAT (36H .....8-NODE SOLID ELEMENT STRESSES   //
     . 24H ELEMENT  LOAD NO.  FACE    ,5X,
     .   104H SIG-XX         SIG-YY         SIG-ZZ         SIG-XY         SIG-YZ
     .   SIG-ZX        SIG-MAX        SIG-MIN     S2/ANGLE)
 3005 FORMAT (I6,I9,I8,2X,1P9E12.2)
 3015 FORMAT (15X,  I8,2X,1P9E12.2)
      END
```

```
      SUBROUTINE BRICK8 (S,STR,NBRK8,NMAT,NLD,ID,X,Y,Z,T,EE,ENU,RHO,
     .                   ALPT,KTYPE,PR,YREF,NFACE,NUMNP)
C
C     STIFFNESS SUBROUTINE FOR 24 D.F. ISOPARAMETRIC HEXAHEDRON
C     LINEAR ELASTIC ISOTROPIC MATERIAL
C     *NINT*NINT*NINT* GAUSSIAN INTEGRATION RULE USED (NINT=1,2,3,4)
C
      DIMENSION KTYPE(1),PR(1),YREF(1),NFACE(1)
      DIMENSION T(1)
      DIMENSION X(1),Y(1),Z(1),ID(NUMNP,6)
      COMMON/EM/LM(24),ND,NS, SS(24,24),RF(24,4),XM(24),SA(12,24),
     .   SF(12,4)
      EQUIVALENCE (IS1,SF(4)) , (IS2,SF(6))
      DIMENSION EE(1),ENU(1),RHO(1),ALPT(1)
      COMMON /GASS/ XK(4,4),WGT(4,4),IPERM(3)
      COMMON /JUNK/ E1,E2,E3,DET,MLD(4),KLD(4),MULT(4),NP(8),INP(8),
     .              A(3,3),P(11,3),B(3,3),XX(8,3),Q(11),OL(8),
     .              TT(24),XLF(4),YLF(4),ZLF(4),TLF(4),PLF(4),
```

```
     .                  REFT,INEL,ININT,IMAT,IINC,TTEMP,NEL,ML,NINT,MAT,
     .                  INC,TAG,TEMP,SKIP,I,J,K,L,FAC,CC1,CC2,CC3,CC4,G,
     .                  DEN,FACT,GT,GG,C1,C2,C3,C,K1,K2
      COMMON /ELPAR/ NPAR(14),NUMN ,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
      DIMENSION S(33,33),STR(12,33)
      DIMENSION E(3,3)
      DIMENSION IS(2),ISP(2)
C
      DATA XK /      0.,               0.,               0.,               0.,
     1 -.5773502691896, .5773502691896,               0.,               0.,
     2 -.7745966692415, .0000000000000, .7745966692415,               0.,
     3 -.8611363115941,-.3399810435849, .3399810435849, .8611363115941/
      DATA WGT / 2.000,              0.,               0.,               0.,
     1 1.0000000000000,1.0000000000000,               0.,               0.,
     2  .5555555555556, .8888888888889, .5555555555556,               0.,
     3  .3478548451375, .6521451548625, .6521451548625, .3478548451375/
      DATA IPERM / 2,3,1 /
      DIMENSION STPTS(7,3)
      DATA STPTS / 0. , 1. ,-1. , 0. , 0. , 0. , 0. ,
     .             0. , 0. , 0. , 1. ,-1. , 0. , 0. ,
     .             0. , 0. , 0. , 0. , 0. , 1. ,-1. /
C
C
C
C     ZERO EM
C
      WRITE (6,3000) NBRK8,NMAT,NLD
      DO 9 I=1,1058
    9 LM(I)=0
C
C     MATERIAL PROPERTIES
C
      WRITE (6,1300)
      DO 1 I=1,NMAT
      READ  (5,1001) N,EE(N),ENU(N),RHO(N),ALPT(N)
    1 WRITE (6,2001) N,EE(N),ENU(N),RHO(N),ALPT(N)
C
C     ELEMENT DISTRIBUTED LOAD CARDS
C
      IF(NLD) 23,23,15
   15 WRITE (6,1302)
      DO 16 I=1,NLD
      READ  (5,1002) N,KTYPE(N),PR(N),YREF(N),NFACE(N)
   16 WRITE (6,2002) N,KTYPE(N),PR(N),YREF(N),NFACE(N)
C
   23 READ  (5,1003) GRAV,PLF,TLF,XLF,YLF,ZLF
      WRITE (6,2003)      PLF,TLF,XLF,YLF,ZLF
      GRAV=1.0
      WRITE (6,1301)
      NEL=0
   30 READ  (5,1000) INEL,INP,ININT,IMAT,IINC,MLD,ISP,TTEMP
      DO 39 I=1,4
   39 MULT(I)=1
      IF(IINC.EQ.0) IINC=1
      IF(IMAT.EQ.0) IMAT=1
   40 NEL=NEL+1
      ML=INEL-NEL
      IF(ML) 50,55,60
   50 WRITE (6,4003) INEL
      STOP
   55 DO 56 I=1,8
   56 NP(I)=INP(I)
      NINT=ININT
      MAT=IMAT
      INC=IINC
      TAG=1H*
      REFT=TTEMP
      IS(1)=ISP(1)
      IS(2)=ISP(2)
      SKIP=99999.
      IF(NINT) 33,33,57
   33 NINT=IABS(NINT)
      SKIP=1.
      IF(NINT.EQ.0) SKIP=0.
   57 CONTINUE
      DO 59 I=1,4
      KLD(I)=IABS(MLD(I))
      IF(MLD(I)) 58,58,59
   58 MULT(I)=0
   59 CONTINUE
      GO TO 62
C
   60 DO 61 I=1,8
   61 NP(I)=NP(I)+INC
      TAG=1H
      DO 64 I=1,4
   64 KLD(I)=KLD(I)*MULT(I)
C
   62 TEMP=0.
      DO 10 I=1,8
      K=NP(I)
      TEMP=TEMP+T(K)
      XX(I,1)=X(K)
      XX(I,2)=Y(K)
   10 XX(I,3)=Z(K)
      TEMP=TEMP*0.125
      K=MAT
      FAC = EE(K)/((1.-2.*ENU(K))*(1.+ENU(K)))
      FACT=FAC*ALPT(K)*(TEMP-REFT)*(1.+ENU(K))
      IF(SKIP) 70,70,63
   63 SKIP=SKIP-1.
      CC1=1.-ENU(K)
      CC2=ENU(K)
      CC3=.5-ENU(K)
C
      L3=33*33
      DO 100 I=1,L3
  100 S(I)=0.
      DO 110 I=1,24
  110 TT(I)=0.
      DO 120 I=1,8
  120 DL(I)=0.
C
C     LOOP OVER NINT**3 INTEGRATION POINTS
C
      DO 300  LX = 1,NINT
      E1=XK(LX,NINT)
      DO 300  LY = 1,NINT
      E2=XK(LY,NINT)
      DO 300  LZ = 1,NINT
      E3=XK(LZ,NINT)
C
      CALL DERIV(1,SA)
C
      GT= WGT(LX,NINT)*WGT(LY,NINT)*WGT(LZ,NINT)*DET
      GG=GT*RHO(MAT)
      G=GT*FAC
      C1=G*CC1
      C2=G*CC2
      C3=G*CC3
C
      L=0
      DO 310 I=1,8
      DL(I)=DL(I) + GG*Q(I)
      DO 310 K=1,3
      L=L+1
  310 TT(L)=TT(L) + GT*SA(I,K)
C
C     ADD CONTRIBUTION TO STIFFNESS MATRIX
C
      DO 300 I=1,11
      K3 = 3*I
      K2 = K3 - 1
      K1 = K2 - 1
      UI=SA(I,1)
      VI=SA(I,2)
      WI=SA(I,3)
      DO 300 J=I,11
      L3 = 3*J
      L2 = L3 - 1
```

```
      L1 = L2 - 1
      UJ=SA(J,1)
      VJ=SA(J,2)
      WJ=SA(J,3)
      UU=UI*UJ
      VV=VI*VJ
      WW=WI*WJ
      UV=UI*VJ
      VU=VI*UJ
      UW=UI*WJ
      WU=WI*UJ
      VW=VI*WJ
      WV=WI*VJ
      S(K1,L1) = S(K1,L1) + C1*UU + C3*(VV+WW)
      S(K2,L2) = S(K2,L2) + C1*VV + C3*(WW+UU)
      S(K3,L3) = S(K3,L3) + C1*WW + C3*(UU+VV)
      S(K1,L2) = S(K1,L2) + C2*UV + C3*VU
      S(K1,L3) = S(K1,L3) + C2*UW + C3*WU
      S(K2,L3) = S(K2,L3) + C2*VW + C3*WV
      IF (I.EQ.J)  GO TO 300
      S(K2,L1) = S(K2,L1) + C2*VU + C3*UV
      S(K3,L1) = S(K3,L1) + C2*WU + C3*UW
      S(K3,L2) = S(K3,L2) + C2*WV + C3*VW
  300 CONTINUE
C
C     FORM STRAIN MATRIX
C
      NSS=2
      IF(IS(2).EQ.0) NSS=1
      DO 305 I=1,12
      DO 305 J=1,33
  305 STR(I,J)=0.
      DO 405 L=1,NSS
      LL=IS(L)+1
      E1=STPTS(LL,1)
      E2=STPTS(LL,2)
      E3=STPTS(LL,3)
      CALL DERIV(2,SA)
      L3=6*L-6
      DO 402 K=1,11
      K3=3*K
      K2=K3-1
      K1=K2-1
      STR(L3+1,K1) = SA(K,1)
      STR(L3+2,K2) = SA(K,2)
      STR(L3+3,K3) = SA(K,3)
      STR(L3+4,K1) = SA(K,2)
      STR(L3+4,K2) = SA(K,1)
      STR(L3+5,K2) = SA(K,3)
      STR(L3+5,K3) = SA(K,2)
      STR(L3+6,K1) = SA(K,3)
  402 STR(L3+6,K3) = SA(K,1)
  405 CONTINUE
      NS=6*NSS
C
C     STATIC CONDENSATION
C
      DO 710 M=1,9
      MN=34-M
      MO=MN-1
C       STIFFNESS MATRIX - S
      SP=S(MN,MN)
      DO 650 I=1,MO
  650 S(MN,I)=S(I,MN)/SP
      DO 700 K=1,MO
      SP=S(MN,K)
      DO 700 J=1,K
  700 S(J,K)=S(J,K) - SP*S(J,MN)
C       DERIVATIVE MATRIX - STR
      DO 710 J=1,NS
      SP=STR(J,MN)
      IF(SP.EQ.0.) GO TO 710
      DO 705 K=1,MO
  705 STR(J,K)=STR(J,K) - SP*S(MN,K)
  710 CONTINUE
C
      DO 760 I=1,24
      DO 760 J=I,24
      SS(I,J)=S(I,J)
  760 SS(J,I)=SS(I,J)
C
C     STRAIN TO STRESS MATRIX
C
      E(1,1)=CC1*FAC
      E(2,2)=E(1,1)
      E(3,3)=E(1,1)
      E(1,2)=CC2*FAC
      E(1,3)=E(1,2)
      E(2,3)=E(1,2)
      E(2,1)=E(1,2)
      E(3,1)=E(1,2)
      E(3,2)=E(1,2)
C
      DO 900 I=1,NSS
      II=I*6-6
      DO 850 J=1,3
      DO 850 K=1,24
      SP=0.
      DO 840 L=1,3
  840 SP = SP + E(J,L)*STR(II+L,K)
      SA(II+J,K)=SP
      JJ=II+3+J
  850 SA(JJ,K)=CC3*FAC*STR(JJ,K)
C
C
      DO 860 J=1,3
      JJ=J+3
      DO 860 K=1,4
      SF(II+J,K)=-FACT*TLF(K)
  860 SF(II+JJ,K)=0.
C
      IF (IS(I).LE.0) GO TO 900
      LL=IS(I)+1
      E1=STPTS(LL,1)
      E2=STPTS(LL,2)
      E3=STPTS(LL,3)
      CALL DERIV (4,SA)
      CALL LOSTR (IS,A,B,SA,SF,I)
C
  900 CONTINUE
C
 70   CONTINUE
C
C     DISTRIBUTED LOAD
C
      DO 410 J=1,24
      DO 410 I=1,4
  410 RF(J,I)=0.
      CALL LOAD (KTYPE,PR,YREF,NFACE)
C
C     SELF WQT.
C
      DO 460 II=1,8
      K=3*II
      J=K-1
      I=J-1
      DO 460 L=1,4
      RF(I,L) = RF(I,L)*PLF(L) + DL(II)*XLF(L)
      RF(J,L) = RF(J,L)*PLF(L) + DL(II)*YLF(L)
  460 RF(K,L) = RF(K,L)*PLF(L) + DL(II)*ZLF(L)
C
C     THERMAL LOADS
C
      DO 470 I=1,24
      GT=TT(I)*FACT
      DO 470 J=1,4
  470 RF(I,J)=RF(I,J) + GT*TLF(J)
C
C     MASS ARRAY
C
```

```
      L=0
      DO 465 I=1,8
      DO 465 J=1,3
      L=L+1
  465 XM(L)=DL(I)/GRAV
C
      IJ=0
      DO 550 I=1,8
      II=NP(I)
      DO 550 J=1,3
      IJ=IJ+1
  550 LM(IJ)=ID(II,J)
      ND=24
C
      IS1=IS(1)
      IS2=IS(2)
      NDM=24
      CALL CALBAN (MBAND,NDIF,LM,XM,SS,RF,ND,NDM)
      WRITE (1) ND,NS,(LM(I),I=1,ND),((SA(I,J),I=1,NS),J=1,ND),
     1 ((SF(I,J),I=1,NS),J=1,4)
      WRITE (6,2000) NEL,NP,NINT,MAT,TAG,KLD,REFT,IS,NDIF
C
C     CHECK IF LAST ELEMENT
C
      IF(NBRK8-NEL) 50,600,590
  590 IF(ML) 30,30,40
C
C
  600 RETURN
C
C
 1000 FORMAT (12I5,4I2,2I1,F10.2)
 1001 FORMAT (I5,4F10.0)
 1002 FORMAT (2I5,2F10.2,I5)
 1003 FORMAT(10X,F10.2/(4F10.2))
 2000 FORMAT (I6,X,8I5,I9,I12,8X,A1,3X,4I5,F9.1,5X,2I3,I8)
 2001 FORMAT (X,I5,4E15.4)
 2002 FORMAT (I5,I9,2F13.3,I12)
 2003 FORMAT (/////
     . 38H LOAD FACTORS FOR 4 ELEMENT LOAD CASES  //
     . 46X 17HELEMENT LOAD CASE /
     . 36X 1HA 9X 1HB 9X 1HC 9X 1HD  /
     . 30H PRESSURE LOAD FACTORS . .       4F10.3/
     . 30H THERMAL  LOAD FACTORS . .       4F10.3//
     . 30H PERCENT GRAVITY IN +X DIRN.   4F10.3/
     . 30H PERCENT GRAVITY IN +Y DIRN.   4F10.3/
     . 30H PERCENT GRAVITY IN +Z DIRN.   4F10.3/ )
 1300 FORMAT (9HOMATERIAL 10X 1HE 12X 2HNU 10X 3HRHO 11X 7HALPHA-T /
     . 8H  NUMBER /)
 1301 FORMAT (30H1....8 NODE SOLID ELEMENT DATA ///
     . 8H ELEMENT 10X 15HCONNECTED NODES 17X 28HINTEGRATION  MATERIAL  I
     .NPUT 7X 13HELEMENT LOADS 5X 7HELEMENT ,5X,6HSTRESS /
     . 8H  NUMBER 3X,36H1    2    3    4    5    6    7    8 6X,5HORDER
     . 7X,3HNO. 6X 3HTAG 7X 15H1    2    3    4 4X  5HTEMP. ,6X,6HPOINTS
     . 5X,4HBAND /)
 1302 FORMAT (/////26H ELEMENT DISTRIBUTED LOADS //
     . 52H NUMBER     KTYPE       PR            YREF            FACE )
 3000 FORMAT ( 31H1......8 - NODE SOLID ELEMENTS  ///
     . 24H  NUMBER OF ELEMENTS.... ,I5 //
     . 24H  NUMBER OF MATERIALS... ,I5 //
     . 24H  NUMBER OF LOAD TYPES.. ,I5 ///)
 4003 FORMAT (36HOELEMENT CARD ERROR, ELEMENT NUMBER= I6)
C
      END
```

```
      SUBROUTINE DERIV(KK,D)
C
      DIMENSION D(12,1)
      COMMON /GASS/ XK(4,4),WGT(4,4),IPERM(3)
      COMMON /JUNK/ R ,S ,T ,DET,MLD(4),KLD(4),MULT(4),NP(8),INP(8),
     .              A(3,3),P(3,11),B(3,3),XX(8,3),Q(11),DL(8)
C
      RP=(1.+R)*.125
      RM=(1.-R)*.125
      SP=1.+S
      SM=1.-S
      TP=1.+T
      TM=1.-T
      IF (KK.EQ.2.OR.KK.EQ.4) GO TO 100
C
C     SHAPE FUNCTIONS
C
      Q(1) = RP*SM*TM
      Q(2) = RP*SP*TM
      Q(3) = RM*SP*TM
      Q(4) = RM*SM*TM
      Q(5) = RP*SM*TP
      Q(6) = RP*SP*TP
      Q(7) = RM*SP*TP
      Q(8) = RM*SM*TP
C
C     DERIVATIVES OF SHAPE FUNCTIONS
C

  100 P(1,1) = SM*TM*.125
      P(1,2) = SP*TM*.125
      P(1,3) = -P(1,2)
      P(1,4) = -P(1,1)
      P(1,5) = SM*TP*.125
      P(1,6) = SP*TP*.125
      P(1,7) = -P(1,6)
      P(1,8) = -P(1,5)
      P(1,9) = -R
      P(1,10)= 0.
      P(1,11)= 0.
C
      P(2,1) = -RP*TM
      P(2,2) = -P(2,1)
      P(2,3) = RM*TM
      P(2,4) = -P(2,3)
      P(2,5) = -RP*TP
      P(2,6) = -P(2,5)
      P(2,7) = RM*TP
      P(2,8) = -P(2,7)
      P(2,9) = 0.
      P(2,10)= -S
      P(2,11)= 0.
C
      P(3,1) = -RP*SM
      P(3,2) = -RP*SP
      P(3,3) = -RM*SP
      P(3,4) = -RM*SM
      P(3,5) = -P(3,1)
      P(3,6) = -P(3,2)
      P(3,7) = -P(3,3)
      P(3,8) = -P(3,4)
      P(3,9) = 0.
      P(3,10)= 0.
      P(3,11)= -T
C
C     JACOBIAN MATRIX A
C
      DO 200 I=1,3
      DO 200 J=1,3
      C=0.
      DO 150 L=1,8
  150 C = C + P(I,L)*XX(L,J)
  200 A(I,J) = C
C
C     INVERT JACOBIAN
```

```
C
      IF(KK.EQ.3) GO TO 500
      DO 250 I=1,3
      J = IPERM(I)
      K = IPERM(J)
      B(I,I) = A(J,J)*A(K,K) - A(K,J)*A(J,K)
      B(I,J) = A(K,J)*A(I,K) - A(I,J)*A(K,K)
  250 B(J,I) = A(J,K)*A(K,I) - A(J,I)*A(K,K)
      IF (KK.EQ.4) GO TO 500
      DET = A(1,1)*B(1,1) + A(1,2)*B(2,1) + A(1,3)*B(3,1)
C
C     MATRIX OF X-Y-Z DERIVATIVES
      DO 400 I=1,3
      DO 400 J=1,11
      C = 0.
      DO 350 K=1,3
  350 C = C + B(I,K)*P(K,J)
  400 D(J,I)=C/DET
C
  500 RETURN
C
      END
```

```
      SUBROUTINE LOAD (KTYPEE,PRR,YREFF,NFACE)
C
      COMMON/EM/LM(24),ND,NS, ES(24,24),RF(24,4),XM(24),SA(12,24),
     .      SF(12,4)
      COMMON /JUNK/ ETA(3)   ,DET,MLD(4),KLD(4),MULT(4),NP(8),INP(8),
     .                  A(3,3),P(11,3),B(3,3),XX(8,3),Q(11),DL(8)
      DIMENSION KTYPEE(1),PRR(1),YREFF(1),NFACE(1)
      COMMON /GASS / DUM(12),XK(4),DDUM(12),WGT(4),IPERM(3)
      DIMENSION KCRD(6),FVAL(6),KFACE(6,4)
C
      DATA KFACE /  1, 4, 2, 1, 6, 2,
     1              2, 3, 3, 4, 7, 3,
     2              6, 7, 7, 8, 8, 4,
     3              5, 8, 6, 5, 5, 1/
      DATA KCRD / 1,1,2,2,3,3/
      DATA FVAL /1.,-1.,1.,-1.,1.,-1./
C
C
      DO 700 KK=1,4
      NNN=KLD(KK)
      IF(NNN) 700,700,10
   10 KTYPE=KTYPEE(NNN)
      PR=PRR(NNN)
      YREF=YREFF(NNN)
      KF=NFACE(NNN)
C
C        INTEGRATE OVER THE SURFACE
C
      ML = KCRD(KF)
      MM = IPERM(ML)
      MN = IPERM(MM)
      ETA(ML) = FVAL(KF)
      DO 300  LX = 1,4
      ETA(MM) = XK(LX)
      DO 300  LY = 1,4
      ETA(MN) = XK(LY)
      CALL DERIV(3,SA)
C
C        COMPUTE DIRECTION COSINES OF NORMAL TO SURFACE
C
      A1 = (A(MM,2)*A(MN,3)-A(MM,3)*A(MN,2))
      A2 = (A(MM,3)*A(MN,1)-A(MM,1)*A(MN,3))
      A3 = (A(MM,1)*A(MN,2)-A(MM,2)*A(MN,1))
      AA=SQRT(A1**2+A2**2+A3**2)
      A1 = A1/AA
      A2 = A2/AA
      A3 = A3/AA
C
C        COMPUTE FIRST FUND. FORM    (SIN / )
C
      AA = 0.
      BB = 0.
      CC = 0.
      DO 200  I = 1,3
      AA=AA+A(MM,I)**2
      CC=CC+A(MN,I)**2
  200 BB = BB + A(MM,I)*A(MN,I)
      C=SQRT (AA*CC - BB*BB)
C
C        COMPUTE PRESSURE,LOAD COMPONENTS, STORE IN R
C
      IF (KTYPE.EQ.2) GO TO 170
      FORCE = PR
      GO TO 185
  170 YY = 0.
      DO 180  I = 1,8
  180 YY = YY + Q(I)*XX(I,2)
      YY = YY - YREF
      FORCE = -PR*YY
      IF(YY.GT.0.) FORCE = 0.
  185 CONTINUE
      TS=FORCE*WGT(LX)*WGT(LY)*C
C
      DO 190  I = 1,4
      N = KFACE(KF,I)
      QQ=TS*Q(N)
      K=3*N
      RF(K-2,KK) = RF(K-2,KK) + QQ*A1
      RF(K-1,KK) = RF(K-1,KK) + QQ*A2
      RF(K  ,KK) = RF(K  ,KK) + QQ*A3
  190 CONTINUE
C
  300 CONTINUE
C
  700 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE LOSTR (IS,A,B,SA,SF,L)
      DIMENSION IS(2),A(3,3),B(3,3),SA(12,24),SF(12,4),IRF(6,2),TC(6,24)
     1,TR(6,6)
      DATA IRF /1,1,2,2,3,3,
     1          2,2,3,3,1,1/
C
      LL=IS(L)
      I=IRF(LL,1)
      TT=B(1,I)*B(1,I)+B(2,I)*B(2,I)+B(3,I)*B(3,I)
      TT=SQRT(TT)
      TC(3,1)=B(1,I)/TT
      TC(3,2)=B(2,I)/TT
      TC(3,3)=B(3,I)/TT
      I=IRF(LL,2)
      TT=A(I,1)*A(I,1)+A(I,2)*A(I,2)+A(I,3)*A(I,3)
      TT=SQRT(TT)
      TC(1,1)=A(I,1)/TT
      TC(1,2)=A(I,2)/TT
      TC(1,3)=A(I,3)/TT
      TC(2,1)=TC(3,2)*TC(1,3)-TC(3,3)*TC(1,2)
      TC(2,2)=TC(3,3)*TC(1,1)-TC(3,1)*TC(1,3)
      TC(2,3)=TC(3,1)*TC(1,2)-TC(3,2)*TC(1,1)
C
      TR(1,1)=TC(1,1)*TC(1,1)
      TR(1,2)=TC(1,2)*TC(1,2)
      TR(1,3)=TC(1,3)*TC(1,3)
```

```
      TR(1,4)=TC(1,1)*TC(1,2)*2.
      TR(1,5)=TC(1,2)*TC(1,3)*2.
      TR(1,6)=TC(1,1)*TC(1,3)*2.
      TR(2,1)=TC(2,1)*TC(2,1)
      TR(2,2)=TC(2,2)*TC(2,2)
      TR(2,3)=TC(2,3)*TC(2,3)
      TR(2,4)=TC(2,1)*TC(2,2)*2.
      TR(2,5)=TC(2,2)*TC(2,3)*2.
      TR(2,6)=TC(2,1)*TC(2,3)*2.
      TR(3,1)=TC(3,1)*TC(3,1)
      TR(3,2)=TC(3,2)*TC(3,2)
      TR(3,3)=TC(3,3)*TC(3,3)
      TR(3,4)=TC(3,1)*TC(3,2)*2.
      TR(3,5)=TC(3,2)*TC(3,3)*2.
      TR(3,6)=TC(3,1)*TC(3,3)*2.
      TR(4,1)=TC(1,1)*TC(2,1)
      TR(4,2)=TC(1,2)*TC(2,2)
      TR(4,3)=TC(1,3)*TC(2,3)
      TR(4,4)=TC(1,1)*TC(2,2)+TC(1,2)*TC(2,1)
      TR(4,5)=TC(1,2)*TC(2,3)+TC(1,3)*TC(2,2)
      TR(4,6)=TC(1,1)*TC(2,3)+TC(1,3)*TC(2,1)
      TR(5,1)=TC(2,1)*TC(3,1)
      TR(5,2)=TC(2,2)*TC(3,2)
      TR(5,3)=TC(2,3)*TC(3,3)
      TR(5,4)=TC(2,1)*TC(3,2)+TC(2,2)*TC(3,1)
      TR(5,5)=TC(2,2)*TC(3,3)+TC(2,3)*TC(3,2)
      TR(5,6)=TC(2,1)*TC(3,3)+TC(2,3)*TC(3,1)
      TR(6,1)=TC(3,1)*TC(1,1)
      TR(6,2)=TC(3,2)*TC(1,2)
      TR(6,3)=TC(3,3)*TC(1,3)
      TR(6,4)=TC(3,1)*TC(1,2)+TC(3,2)*TC(1,1)
      TR(6,5)=TC(3,2)*TC(1,3)+TC(3,3)*TC(1,2)
      TR(6,6)=TC(3,1)*TC(1,3)+TC(3,3)*TC(1,1)
C
      IL=(L-1)*6
      DO 100 I=1,6
      DO 100 J=1,24
      TC(I,J)=0.
      DO 100 K=1,6
  100 TC(I,J)=TC(I,J)+TR(I,K)*SA(IL+K,J)
      DO 110 I=1,6
      DO 110 J=1,24
  110 SA(IL+I,J)=TC(I,J)
C
      DO 120 I=1,6
      DO 120 J=1,4
      TC(I,J)=0.
      DO 120 K=1,6
  120 TC(I,J)=TC(I,J)+TR(I,K)*SF(IL+K,J)
      DO 130 I=1,6
      DO 130 J=1,4
  130 SF(IL+I,J)=TC(I,J)
C
      RETURN
      END
```

```
      SUBROUTINE PRIST (NS,IS1,IS2,SIG,SPR)
      DIMENSION SIG(12),SPR(6),IS(2),SG(6)
C
      IS(1)=IS1
      IS(2)=IS2
      NNS=1
      IF (NS.EQ.12) NNS=2
      DO 900 N=1,NNS
      IN=3*N-3
      II=IN*2
      IF (IS(N).EQ.0) GO TO 200
C
      CC=(SIG(II+1)+SIG(II+2))/2.
      BB=(SIG(II+1)-SIG(II+2))/2.
      CR=SQRT(BB**2+SIG(II+4)**2)
      SPR(IN+1)=CC+CR
      SPR(IN+2)=CC-CR
      SPR(IN+3)=0.
      IF (BB.NE.0.)SPR(IN+3)=28.648*ATAN2(SIG(II+4),BB)
      GO TO 900
C
  200 CC=(SIG(II+1)+SIG(II+2)+SIG(II+3))/3.
      DO 210 I=1,3
      SG(I)=SIG(II+I)-CC
  210 SG(I+3)=SIG(II+I+3)
      C2=(SG(1)**2+SG(2)**2+SG(3)**2)*.5+SG(4)**2+SG(5)**2+SG(6)**2
      C3=SG(1)*(SG(2)*SG(3)-SG(5)*SG(5))+SG(4)*(SG(5)*SG(6)-SG(4)*SG(3))
     1+SG(6)*(SG(4)*SG(5)-SG(2)*SG(6))
      T=SQRT(C2/1.5)
      A=C3*1.414214/T**3
      IF ( A .LT. -1.0 ) A=-1.0
      IF ( A .GT.  1.0 ) A= 1.0
      A=ACOS(A)/3.
      T=T*1.414214
      SPR(IN+1)=T*COS(A)
      SPR(IN+2)=T*COS(A+2.0944)
      SPR(IN+3)=T*COS(A-2.0944)
      DO 220 I=2,3
      IF (SPR(IN+1).GT.SPR(IN+I)) GO TO 220
      C3=SPR(IN+1)
      SPR(IN+1)=SPR(IN+I)
      SPR(IN+I)=C3
  220 CONTINUE
      IF (SPR(IN+2).LE.SPR(IN+3)) GO TO 230
      C3=SPR(IN+2)
      SPR(IN+2)=SPR(IN+3)
      SPR(IN+3)=C3
  230 DO 240 I=1,3
  240 SPR(IN+I)=SPR(IN+I)+CC
  900 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE SHELL
      COMMON A(1)
      COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
      COMMON / JUNK / LT,LH,L,SIG(20)
C
      IF (NPAR(1).EQ.0) GO TO 500
C     PROTECT NODAL TEMPERATURES
      N6=N5+NUMNP+12*NPAR(3)
      IF (N6.GT.MTOT) CALL ERROR (N6-MTOT)
      CALL TPLATE(NPAR(2),NPAR(3),A(N1),A(N2),A(N3),A(N4),A(N5),NUMNP,
     1  MBAND)
      RETURN
C
  500 WRITE (6,2002)
      NUME=NPAR(2)
      DO 800 MM=1,NUME
      CALL STRSC (A(N1),A(N3),NEQ,0)
      WRITE (6,2001)
      DO 800 L=LT,LH
      CALL STRSC (A(N1),A(N3),NEQ,1)
      WRITE (6,3002) MM,L,(SIG(I),I=1,6)
  800 CONTINUE
C
      RETURN
C
 2001 FORMAT (/)
 2002 FORMAT (24H1 SHELL ELEMENT STRESSES//
     1 102H               ELEMENT       LOAD      MEMBRANE STRESS COMPONENTS
     2              BENDING MOMENT COMPONENTS       /
     3 102H                NUMBER       CASE       SXX           SYY           S
     4XY          MXX          MYY           MXY   //)
 3002 FORMAT (10X,2I10,6E12.4)
      END


      SUBROUTINE TPLATE(NUMEL,NUMMAT,ID,X,Y,Z,C,NUMNP,MBAND)
C
      COMMON/QTSARG/
     1XX(5),YY(5),ZZ(5),HM(5),HP(5),CM(3,3),ALFA(3),HW(5),RHO(5,3),P(5)
     2,T(5),DT(5),SM(5,3),BM(5,3),TDIS(3,3,4),TROT(3,3,4),S(42,42),R(42)
      COMMON/EM/LM(24),ND,NS,ASA(24,24),RF(24,4),XM(24),SA(12,24)
     1,SF(12,4)
      DIMENSION X(1),Y(1),Z(1),ID(NUMNP,1),C(12,1),IX(7),IY(7),EL(4)
     1,TO(3,3),TLO(5,4)
C
      LL=4
C
      DO 4 I=1,12
      DO 4 J=1,4
    4 SF(I,J)=0.
      DO 5 I=1,24
      DO 5 J=1,4
    5 RF(I,J)=0.0
      NPF=6
      NS=6
      MID=4
      MID=0
      IDIS=0
      IROT=0
      ISTOP=0
      MTYPE=6
      WRITE (6,2000) MTYPE,NUMEL,NUMMAT
C
C *** READ AND PRINT OF MATERIAL PROPERTIES
C
      WRITE (6,2001)
      DO 10 J=1,NUMMAT
      READ  (5,1000) N,(C(I,N),I=1,12)
   10 WRITE (6,2002) N,(C(I,N),I=3,12)
C
C *** READ AND PRINT OF ELEMENT LOAD MULTIPLIERS
C
      READ  (5,1002) ((TLO(I,J),J=1,4),I=1,5)
      WRITE (6,2006)
      WRITE (6,2007) (J,(TLO(I,J),I=1,5),J=1,4)
C
C *** READ AND PRINT OF ELEMENT DATA
C
      WRITE (6,2003)
      NN=0
  100 READ  (5,1001) MM,IY,EL
  110 NN=NN+1
      IF (MM-NN) 440,50,60
   50 DO 45 I=1,7
   45 IX(I)=IY(I)
      INCL=IY(7)
      IF (IY(6).EQ.0) IY(6)=1
      IM=IY(6)
      IF(INCL.EQ.0) INCL=1
      NNS=5
      IF (IY(5).EQ.0) NNS=4
      IF (IY(4).EQ.0) NNS=3
      RHOM=C(3,IM)
      ALFA(1)=C(4,IM)
      ALFA(2)=C(5,IM)
      ALFA(3)=C(6,IM)
      CM(1,1)=C(7,IM)
      CM(1,2)=C(8,IM)
      CM(1,3)=C(9,IM)
      CM(2,2)=C(10,IM)
      CM(2,3)=C(11,IM)
      CM(3,3)=C(12,IM)
      CM(2,1)=CM(1,2)
      CM(3,1)=CM(1,3)
      CM(3,2)=CM(2,3)
C
      DO 30 I=1,NNS
      HM(I)=EL(1)
      HP(I)=EL(1)
      HW(I)=EL(1)
      SM(I,1)=0.0
      SM(I,2)=0.0
      SM(I,3)=0.0
      BM(I,1)=0.0
      BM(I,2)=0.0
   30 BM(I,3)=0.0
      GO TO 70
C
   60 DO 65 I=1,NNS
   65 IX(I)=IX(I)+INCL
C
   70 DO 40 I=1,NNS
      J=IX(I)
      XX(I)=X(J)
      YY(I)=Y(J)
   40 ZZ(I)=Z(J)
C
      DO 550 IL=1,LL
      DO 520 I=1,NNS
      RHO(I,1)=TLO(3,IL)*RHOM
      RHO(I,2)=TLO(4,IL)*RHOM
      RHO(I,3)=TLO(5,IL)*RHOM
      P(I)=TLO(1,IL)*EL(2)
      T(I)=TLO(2,IL)*EL(3)
  520 DT(I)=TLO(2,IL)*EL(4)
      IF (IL.GE.2) GO TO 530
C
      CALL QTSHEL  (0,NNS,NPF,MID,IDIS,IROT,ND ,NTF)
C
      DO 510 I=1,ND
      RF(I)=R(I)
      XM(I)=0.
      DO 510 J=1,ND
  510 ASA(I,J)=S(I,J)
      GO TO 550
```

```
C
  530 CALL QTSHEL  (1,NNS,NPF,MID,IDIS,IROT,ND ,NTF)
C
      DO 540 I=1,ND
  540 RF(I,IL)=R(I)
  550 CONTINUE
      WRITE (6,2004) NN,(IX(I),I=1,6),HM(1),P(1),T(1),DT(1)
C
C *** FORM LM ARRAY AND COMPUTE BANDWIDTH
C
      DO 410 I=1,NNS
      J=NPF*I-NPF
      N=IX(I)
      DO 410 K=1,NPF
  410 LM(J+K)=ID(N,K)
C
      CALL QDCOS (4,XX,YY,ZZ,TO)
      H=HM(1)
C
      CALL STRETR (TO,XX,YY,ZZ,CM,SA,H,RHOM,XM)
C
      NDM=24
      CALL CALBAN (MBAND,NDIF,LM,XM,ASA,RF,ND,NDM)
      WRITE (1) ND,NS,(LM(I),I=1,ND),((SA(I,J),I=1,NS),J=1,ND),
     1 ((SF(I,J),I=1,NS),J=1,4)
C
      GO TO 500
  440 WRITE (6,2005) MM
      ISTOP=1
  500 IF (NN.LT.MM) GO TO 110
      IF (NN.EQ.NUMEL) RETURN
      IF (ISTOP.EQ.1) STOP
      GO TO 100
C
 2000 FORMAT (34H1...THIN ELASTIC SHELL ELEMENTS...//
     1 25H ELEMENT TYPE............I5//
     2 25H NUMBER OF ELEMENTS......I5//
     3 25H NUMBER OF MATERIALS.....I5)
 1000 FORMAT (I10,6F10.0/6F10.0)
 2001 FORMAT (36H1...MATERIAL PROPERTY INFORMATION...//
     1 131H                                    MASS   THERMAL EXPANSION COEF
     2FICIENTS     ....................ELASTIC COEFFICIENTS.............
     3..../
     4 131H TYPE                          DENSITY          AX          AY
     5         AZ          CXX         CXY          CXG         CYY          CYG
     6GXY //)
 2002 FORMAT (1H I4,22X,4E11.4,5F10.4)
 1002 FORMAT (4F10.0)
 2006 FORMAT (31H1...ELEMENT LOAD MULTIPLIERS...//
     1 90H                                                       ..........GRAVI
     2TY ACCELERATION...........  /
     3 90H    LOAD CASE     LATERAL LOAD     TEMPERATURE     X-COMPONENT
     4Y-COMPONENT     Z-COMPONENT  /)
 2007 FORMAT (1X,I10,5X,5F15.5)
 2003 FORMAT (15H1*ELEMENT DATA*//
     1 96H ELEMENT            NODAL POINT NUMBERS          MATL
     2                          TEMPERATURE/
     3 95H  NUMBER           I     J     K     L     O       TYPE    THICKNESS
     4  PRESSURE TEMPERATURE   GRADIENT /)
 1001 FORMAT (8I5,4F10.0)
 2004 FORMAT (3X,I5,5X,5I5,5X,I5,4F12.4)
 2005 FORMAT (23H0 ELEMENT CARD IN ERROR,I5)
C
      END
```

```
      SUBROUTINE STRETR (T,X,Y,Z,CM,SA,H,RHOM,XM)
C
C *** FORM DISPLACEMENT-STRESS TRANSFORMATION MATRIX
C
      DIMENSION T(3,3),X(1),Y(1),Z(1),CM(3,3),SA(12,24),AA(12,12),MM(12)
     1,A(12,12),AM(4),S1(3,9),S2(3,9),S3(3,12),V(3,3),XY(3,2),XM(1)
C
      DO 5 I=1,12
      DO 5 J=1,24
    5 SA(I,J)=0.
      DO 10 I=1,3
      V(I,1)=X(I+1)-X(1)
      V(I,2)=Y(I+1)-Y(1)
   10 V(I,3)=Z(I+1)-Z(1)
      DO 20 I=1,3
      DO 20 J=1,2
      XY(I,J)=0.
      DO 20 K=1,3
   20 XY(I,J)=XY(I,J)+V(I,K)*T(J,K)
      XO=(XY(1,1)+XY(2,1)+XY(3,1))/4.0
      YO=(XY(1,2)+XY(2,2)+XY(3,2))/4.0
C
C *** FORM LUMP MASS MATRIX
C
      AM1=XY(1,1)*YO-XO*XY(1,2)
      AM2=(XY(1,1)-XO)*(XY(2,2)-YO)-(XY(2,1)-XO)*(XY(1,2)-YO)
      AM3=(XY(2,1)-XO)*(XY(3,2)-YO)-(XY(3,1)-XO)*(XY(2,2)-YO)
      AM4=XO*XY(3,2)-XY(3,1)*YO
      AC=0.25*H*RHOM
      AM(1)=AC*(AM1+AM4)
      AM(2)=AC*(AM1+AM2)
      AM(3)=AC*(AM2+AM3)
      AM(4)=AC*(AM3+AM4)
C
      DO 25 L=1,4
      XX=AM(L)
      DO 25 J=1,3
      JS=6*(L-1)+J
      XM(JS)=XX
   25 XM(JS+3)=0.
C
C *** FORM STRESS TRANSFORMATION FOR IN PLANE STRESS
C
      S=0.5
      R=0.5
      DO 100 I=1,3
      DO 100 J=1,8
  100 S1(I,J)=0.0
C
      XYJ=XY(1,1)*XY(3,2)-XY(3,1)*XY(1,2)+(XY(1,1)*(XY(2,2)-XY(3,2))
     1 -(XY(2,1)-XY(3,1))*XY(1,2))*S-((XY(1,1)-XY(2,1))*XY(3,2)
     2 -XY(3,1)*(XY(1,2)-XY(2,2)))*R
      X1=XY(1,2)-XY(3,2)-(XY(2,2)-XY(3,2))*S-(XY(1,2)-XY(2,2))*R
      X2=XY(3,2)+(XY(2,2)-XY(3,2))*S-XY(3,2)*R
      X3=-XY(1,2)*S+XY(3,2)*R
      X4=-XY(1,2)+XY(1,2)*S+(XY(1,2)-XY(2,2))*R
      Y1=-XY(1,1)+XY(3,1)+(XY(2,1)-XY(3,1))*S+(XY(1,1)-XY(2,1))*R
      Y2=-XY(3,1)-(XY(2,1)-XY(3,1))*S+XY(3,1)*R
      Y3=XY(1,1)*S-XY(3,1)*R
      Y4=XY(1,1)-XY(1,1)*S-(XY(1,1)-XY(2,1))*R
C
      S1(1,1)=X1/XYJ
      S1(1,3)=X2/XYJ
      S1(1,5)=X3/XYJ
      S1(1,7)=X4/XYJ
      S1(2,2)=Y1/XYJ
      S1(2,4)=Y2/XYJ
      S1(2,6)=Y3/XYJ
      S1(2,8)=Y4/XYJ
      S1(3,1)=S1(2,2)
      S1(3,2)=S1(1,1)
      S1(3,3)=S1(2,4)
      S1(3,4)=S1(1,3)
      S1(3,5)=S1(2,6)
      S1(3,6)=S1(1,5)
```

```
      S1(3,7)=S1(2,8)
      S1(3,8)=S1(1,7)
      DO 105 I=1,3
      DO 105 J=1,8
      S2(I,J)=0.
      DO 104 K=1,3
  104 S2(I,J)=S2(I,J)+CM(I,K)*S1(K,J)
  105 S2(I,J)=S2(I,J)*H
C
      H=H**3/12.
      DO 110 L=1,4
      DO 110 I=1,3
      DO 110 J=1,3
      JS=6*(L-1)+J
      KA=2*(L-1)+1
  110 SA(I,JS)=S2(I,KA)*T(1,J)+S2(I,KA+1)*T(2,J)
C
C *** FORM STRESS TRANSFORMATION FOR MOMENT
C
      DO 30 I=1,12
      DO 30 J=1,12
      AA(I,J)=0.
   30 A(I,J)=0.
C
      DO 40 I=1,3
      J=3*I+1
      A(1,J)=1.
      A(2,J)=XY(I,1)
      A(3,J)=XY(I,2)
      A(4,J)=XY(I,1)*XY(I,1)
      A(5,J)=XY(I,1)*XY(I,2)
      A(6,J)=XY(I,2)*XY(I,2)
      A(7,J)=A(4,J)*XY(I,1)
      A(8,J)=A(5,J)*XY(I,1)
      A(9,J)=A(5,J)*XY(I,2)
      A(10,J)=A(6,J)*XY(I,2)
      A(11,J)=XY(I,1)*XY(I,1)*XY(I,1)*XY(I,2)
      A(12,J)=XY(I,1)*XY(I,2)*XY(I,2)*XY(I,2)
      A(3,J+1)=1.0
      A(5,J+1)=XY(I,1)
      A(6,J+1)=2.0*XY(I,2)
      A(8,J+1)=XY(I,1)*XY(I,1)
      A(9,J+1)=2.0*XY(I,1)*XY(I,2)
      A(10,J+1)=3.0*XY(I,2)*XY(I,2)
      A(11,J+1)=XY(I,1)*XY(I,1)*XY(I,1)
      A(12,J+1)=3.0*XY(I,1)*XY(I,2)*XY(I,2)
      A(2,J+2)=-1.0
      A(4,J+2)=-2.0*XY(I,1)
      A(5,J+2)=-XY(I,2)
      A(7,J+2)=-3.0*XY(I,1)*XY(I,1)
      A(8,J+2)=-2.0*XY(I,1)*XY(I,2)
      A(9,J+2)=-XY(I,2)*XY(I,2)
      A(11,J+2)=-3.0*XY(I,1)*XY(I,1)*XY(I,2)
      A(12,J+2)=-XY(I,2)*XY(I,2)*XY(I,2)
   40 CONTINUE
      A(1,1)=1.0
      A(3,2)=1.0
      A(2,3)=-1.0
C
      CALL INVERT (A,12,12,MM,AA)
C
      DO 210 I=1,3
      DO 210 J=1,9
  210 S1(I,J)=0.
C
      S1(1,1)=2.0
      S1(1,4)=6.0*XO
      S1(1,5)=2.0*YO
      S1(1,8)=6.0*XO*YO
      S1(2,3)=2.0
      S1(2,6)=2.0*XO
      S1(2,7)=6.0*YO
      S1(2,9)=6.0*XO*YO
      S1(3,2)=2.0
      S1(3,5)=4.0*XO
      S1(3,6)=4.0*YO
      S1(3,8)=6.0*XO*XO
      S1(3,9)=6.0*YO*YO
C
      DO 220 I=1,3
      DO 220 J=1,9
      S2(I,J)=0.
      DO 215 K=1,3
  215 S2(I,J)=S2(I,J)+CM(I,K)*S1(K,J)
  220 S2(I,J)=-S2(I,J)*H
C
      DO 230 I=1,3
      DO 230 J=1,12
      S3(I,J)=0.
      DO 230 K=1,9
  230 S3(I,J)=S3(I,J)+S2(I,K)*A(J,3+K)
C
      DO 240 L=1,4
      DO 240 I=1,3
      DO 240 J=1,3
      JS=6*(L-1)+J
      KA=3*(L-1)+1
      SA(I+3,JS)=S3(I,KA)*T(3,J)
      JS=JS+3
      SA(I+3,JS)=S3(I,KA+1)*T(1,J)+S3(I,KA+2)*T(2,J)
  240 CONTINUE
      RETURN
      END

      SUBROUTINE INVERT(A,NN,N,M,C)
C
C           GENERAL MATRIX INVERSION SUBROUTINE
C
      DIMENSION A(1),M(1),C(1)
C
      DO 90 I=1,NN
   90 M(I)=-I
C
      DO 140 I=1,NN
C
C     LOCATE LARGEST ELEMENT
C
      D=0.0
      DO 112 L=1,NN
      IF (M(L)) 100,100,112
  100 J=L
      DO 110 K=1,NN
      IF (M(K)) 103,103,108
  103 IF (ABS(D)-ABS(A(J))) 105,105,108
  105 LD=L
      KD=K
      D=A(J)
  108 J=J+N
  110 CONTINUE
  112 CONTINUE
C
C     INTERCHANGE ROWS
C
      TEMP=-M(LD)
      M(LD)=M(KD)
      M(KD)=TEMP
      L=LD
      K=KD
      DO 114 J=1,NN
      C(J)=A(L)
      A(L)=A(K)
      A(K)=C(J)
      L=L+N
  114 K=K+N
```

```
C
C       DIVIDE COLUMN BY LARGEST ELEMENT
C
        NR=(KD-1)*N+1
        NH=NR+N-1
        DO 115 K=NR,NH
  115   A(K)=A(K)/D
C
C       REDUCE REMAINING ROWS AND COLUMNS
C
        L=1
        DO 135 J=1,NN
        IF (J-KD) 130,125,130
  125   L=L+N
        GO TO 135
  130   DO 134 K=NR,NH
        A(L)=A(L)-C(J)*A(K)
  134   L=L+1
  135   CONTINUE
C
C       REDUCE ROW
C
        C(KD)=-1.0
        J=KD
        DO 140 K=1,NN
        A(J)=-C(K)/D
  140   J=J+N
C
C       INTERCHANGE COLUMNS
C
        DO 200 I=1,NN
        L=0
  150   L=L+1
        IF(M(L)-I) 150,160,150
  160   K=(L-1)*N+1
        J=(I-1)*N+1
        M(L)=M(I)
        M(I)=I
        DO 200 L=1,NN
        TEMP=A(K)
        A(K)=A(J)
        A(J)=TEMP
        J=J+1
  200   K=K+1
C
        RETURN
C
        END
```

```
        SUBROUTINE QTSHEL (KKK,NNS,NPF,MID,IDIS,IROT,NEF,NTF)
C
C       THIS SUBROUTINE CAN EVALUATE
C               ... ELEMENT STIFFNESS MATRIX       ...
C               ... CONSISTENT NODAL FORCE VECTOR ...
C               ... INTERNAL STRESSES AND MOMENTS ...
C       OF A SHALLOW QUADRILATERAL SHELL ELEMENT ASSEMBLED WITH 4 FLAT
C       TRIANGLES, OR OF A SINGLE TRIANGULAR SHELL ELEMENT.
C
C * * * * * * * * * * * * CALLING ARGUMENTS * * * * * * * * * * * * * *
C
C       INPUTS
C
C        KKK          INTEGER FLAG SPECIFYING OPERATION TO BE PERFORMED
C                      IF KKK =-1, FORM STIFFNESS MATRIX ONLY.
C                      IF KKK = 0, FORM STIFFNESS MATRIX AND LOAD VECTOR.
C                      IF KKK = 1, FORM LOAD VECTOR ONLY.
C                      IF KKK = 2, EVALUATE STRESSES AND MOMENTS.
C        NNS          NUMBER OF SUPPLIED NODAL POINTS
C                      IF NNS = 5, QTSHEL FORMS A QUADRILATERAL, AND THE
C                      PROPERTIES AT THE INTERNAL NODE 5 MUST BE INPUT.
C                      IF NNS = 4, QTSHEL FORMS A QUADRILATERAL, AND THE
C                      PROPERTIES AT THE INTERNAL NODE 5 ARE SET BY QTSHEL
C                      TO BE THEIR CORNER AVERAGE.
C                      IF NNS = 3, QDSTIF FORMS A SINGLE TRIANGLE.
C
C        NPF          NUMBER OF GLOBAL DEGREES OF FREEDOM AT EACH
C                     EXTERNAL NODE (3, 5 OR 6)
C                      IF NPF = 6, THE 3 DISPLACEMENTS U, V AND W AND THE
C                      3 ROTATIONS RX, RY AND RZ ARE INCLUDED AS D.O.F.
C                      IF NPF = 5, THE ROTATION RZ IS IGNORED.
C                      IF NPF = 3, ONLY U, V AND W ARE CONSIDERED AND
C                      THE BENDING STIFFNESS IS NOT INCLUDED (MEMBRANE
C                      SHELL ELEMENT)
C
C        MID          NUMBER OF INTERNAL MIDPOINTS IN QUADRILATERAL (0 OR 4)
C                      IF MID = 0, THE MEMBRANE ELEMENTS ARE    CST
C                              AND THE BENDING  ELEMENTS ARE    LCCT-9
C                      IF MID = 4, THE MEMBRANE ELEMENTS ARE    LST-10
C                              AND THE BENDING  ELEMENTS ARE    LCCT-11
C                     IF NNS = 3 (SINGLE TRIANGLE) MID IS ASSUMED TO BE 0
C
C        IDIS         INTEGER FLAG FOR THE NODAL DISPLACEMENTS U,V,W
C                      IF IDIS = 0, U,V,W ARE SPECIFIED IN THE GLOBAL SYSTEM
C                      IF IDIS = 1, U,V,W ARE SPECIFIED IN THE NODAL DISPL
C                      SYSTEMS DEFINED BY THE DIRECTION COSINE ARRAY TDIS.
C
C        IROT         INTEGER FLAG FOR THE NODAL ROTATIONS RX,RY,RZ.
C                      IF IROT = 0, RX,RY,RZ ARE SPECIFIED IN GLOBAL SYSTEM
C                      IF IROT = 1, RX,RY,RZ ARE SPECIFIED IN THE NODAL ROT
C                      SYSTEMS DEFINED BY THE DIRECTION COSINE ARRAY TROT.
C
C       OUTPUTS
C
C        NEF          NUMBER OF EXTERNAL DEGREES OF FREEDOM (NEF = NPF*NEN,
C                     WHERE NEN=4 FOR QUADRILATERAL, =3 FOR SINGLE TRIANGLE)
C
C        NTF          TOTAL NUMBER OF DEGREES OF FREEDOM (EXTERNAL+INTERNAL)
C
C * * * * * * * * * * ARRAYS IN  COMMON /QTSARG/  * * * * * * * * * * *
C
C X(I),Y(I),Z(I)  I=1...NNS   GLOBAL NODAL COORDINATES
C
C    CM(I,J)      I=1...3, J=1...3   PLANE STRESS MATERIAL MATRIX
C                 RELATING STRESSES TO STRAINS IN THE LOCAL SYSTEM
C
C    ALFA(I)      I=1...3   DILATATION COEFFICIENTS RELATING IN-PLANE
C                 THERMAL STRAINS IN THE LOCAL SYSTEM TO TEMPERATURES
C
C     HM(I)       I=1...NNS   THICKNESS RESISTING MEMBRANE STRESSES
C
C     HP(I)       I=1...NNS   THICKNESS RESISTING BENDING MOMENTS
C
C    RHO(I,J)     I=1...NNS, J=1...3  GLOBAL COMPONENTS RHOX (J=1),
C                 RHOY (J=2) AND RHOZ (J=3) OF BODY FORCES PER UNIT
C                 OF VOLUME
C
C     HW(I)       I=1...NNS   THICKNESS FOR COMPUTING BODY FORCES
C                 RHO*HW PER UNIT OF ELEMENT AREA
C
C     P(I)        I=1...NNS   LATERAL PRESSURE (NORMAL TO THE FACES OF
C                 THE COMPONENT TRIANGLES)
C
C     T(I)        I=1...NNS   MEAN TEMPERATURE VARIATIONS
C
C     DT(I)       I=1...NNS   MEAN TEMPERATURE THICKNESS GRADIENTS
C
C    SM(I,J)      I=1...NNS, J=1...3   ARRAY OF MEMBRANE STRESS
C                 COMPONENTS IN THE LOCAL SYSTEM  SIG-XX (J=1), SIG-YY
C                 (J=2) AND SIG-XY (J=3). SM CONTAINS
C                  MEMBRANE STRESSES IN THE INITIAL POSITION AS INPUT
C                  WHEN KKK=0,1,2 (EXCLUDING THERMAL ACTIONS)
```

```
C                        MEMBRANE STRESSES IN THE DEFORMED POSITION AS OUTPUT
C                        WHEN KKK=2 (INCLUDING THERMAL ACTIONS)
C
C     BM(I,J)           I=1...NNS, J=1...3   ARRAY OF BENDING MOMENT
C                       COMPONENTS IN THE LOCAL SYSTEM  MOM-XX (J=1), MOM-YY
C                       (J=2) AND MOM-XY (J=3). BM CONTAINS
C                        BENDING MOMENTS IN THE INITIAL POSITION AS INPUTS
C                        WHEN KKK=0,1,2 (EXCLUDING THERMAL ACTIONS)
C                        BENDING MOMENTS IN THE DEFORMED POSITION AS OUTPUT
C                        WHEN KKK=2 (INCLUDING THERMAL ACTION)
C
C   TDIS(I,J,K)         I=1...3, J=1...3, K=1...NEN    NOT REQUIRED IF
C                       IDIS=0. IF IDIS=1, TDIS(1..3,1..3,K) MUST CONTAIN
C                       THE (3,3) DIRECTION COSINE MATRIX OF THE NODAL
C                       DISPLACEMENT SYSTEM AT THE K-TH ELEMENT NODE WITH
C                       RESPECT TO THE GLOBAL SYSTEM
C
C   TROT(I,J,K)         I=1...3, J=1...3, K=1...NEN    NOT REQUIRED IF
C                       IROT=0. IF IROT=1, TROT(1..3,1..3,K) MUST CONTAIN
C                       THE (3,3) DIRECTION COSINE MATRIX OF THE NODAL
C                       ROTATION SYSTEM AT THE K-TH ELEMENT NODE WITH
C                       RESPECT TO THE GLOBAL SYSTEM
C
C     S(I,J)            I=1...NEF, J=1...NEF  EXTERNAL STIFFNESS MATRIX
C                       (OUTPUT IF KKK=-1,0)
C
C     S(I,J)            I=1...NTF, J=NEF+1...NTF   REDUCED INTERNAL STIFFNESS
C                       OF QUADRILATERAL ELEMENT. OUTPUT IF KKK=-1,0.
C                       REQUIRED INPUT IF KKK=1,2. NOT USED FOR SINGLE
C                       TRIANGLE.
C
C     R(I)              I=1...NEF     OUTPUT EXTERNAL NODAL FORCES IF KKK=0,1.
C                       INPUT EXTERNAL NODAL DISPLACEMENTS IF KKK=2.
C
C     R(I)              I=NEF+1...NTF   REDUCED INTERNAL NODAL FORCE VECTOR
C                       OF QUADRILATERAL ELEMENT. OUTPUT IF KKK=0,1.
C                       REQUIRED INPUT IF KKK=2 (RETURNS INTERNAL NODAL
C                       DISPLACEMENTS). NOT USED FOR SINGLE TRIANGLE.
C
C * * * * * * * * * * ROLE OF ARRAYS IN COMMON /QTSARG/ * * * * * * * *
C
C       ARRAYS       OPERATION  KKK =-1    KKK = 0    KKK = 1    KKK = 2
C                                Q  T       Q  T       Q  T       Q  T
C
C X,Y,Z,CM,ALFA,HM,HP (*)        I  I       I  I       I  I       I  I
C    RHO,HW,P                    -  -       I  I       I  I       -  -
C       T,DT  (*)                -  -       I  I       I  I       I  I
C       SM,BM (*)                -  -       I  I       I  I      I/O I/O
C     TDIS,TROT (**)             I  I       I  I       I  I       I  I
C  S(1..NEF,1..NEF)              O  O       O  O       -  -       -  -
C S(1..NTF,NEF+1..NTF)           O  -       O  -       I  -       I  -
C      R(1..NEF)                 -  -       O  O       O  O       I  I
C   R(NEF+1..NTF)                -  -       O  -       O  -      I/O -
C
C      WHERE   Q=QUADRILATERAL (NNS=4,5), T=SINGLE TRIANGLE (NNS=3)
C              I=INPUT, O=OUTPUT, I/O=INPUT/OUTPUT, -=NOT USED.
C
C   NOTES   (*)  HP,DT AND BM ARE NOT USED IF NPF=3.
C          (**) TDIS IS NOT USED IF IDIS=0, AND TROT IS NOT USED
C                 IF IROT=0.
C
      COMMON /QTSARG/ X(5),Y(5),Z(5), HM(5),HP(5), CM(3,3),ALFA(3),
     1 HW(5),RHO(5,3),P(5), T(5),DT(5), SM(5,3),BM(5,3), TDIS(3,3,4),
     2 TROT(3,3,4), S(42,42), R(42)
      COMMON /TRIARG/ A(3),B(3), HMT(3),HPT(3), C(3,3), SMT(3,3),
     1 BMT(3,3), FT(12),      P1(3),P2(3),P3(3),RM(3), ST(12,12)
      COMMON /TRANSF/  T1(3),T2(3),T3(3), TO(3,3)
      DIMENSION F(1), IPERMQ(4), MFR(5), LOC(5), NC(3), CA(3), WGT(3),
     1 TD1(13),TD2(13),TD3(9), TR1(9),TR2(9),TR3(9), U(1),V(1),W(1),
     2 RX(1),RY(1)
      EQUIVALENCE (T11,T1),(T12,T1(2)),(T13,T1(3)),(T21,T2),(T22,T2(2)),
     1 (T23,T2(3)),(T31,T3),(T32,T3(2)),(T33,T3(3)), (R,F),
     2  (U,FT), (V,FT(7)), (W,P1), (RX,P2), (RY,P3)
      DATA  IPERMQ /2,3,4,1/, MFR /3,3,3,2,2/, WGT /.50,.50,.25/
      LOGICAL QUAD, TRIG, NOMP, NOST, SIST, NOLD, SILD, NOSM, SISM, SKMP
C
C     INITIALIZE
C
      SIST = KKK.LE.0
      NOST = .NOT.SIST
      SILD = KKK.EQ.0.OR.KKK.EQ.1
      NOLD = .NOT.SILD
      SISM = KKK.GE.2
      NOSM = .NOT.SISM
      IF ((NNS.NE.3).AND.(NNS.NE.5))  NNS = 4
      IF ((NPF.NE.3).AND.(NPF.NE.6))  NPF = 5
      NEN = MINO (NNS,4)
      QUAD = NEN.EQ.4
      TRIG = NEN.EQ.3
      WG = 1.
      N3 = 2*NEN - 3
      NTRI = 3*NEN - 8
      NEF = NEN*NPF
      NSF = NEF + (NEN-3)*NPF
      IF (MID.NE.4)  MID = 0
      MIDP = MID
      IF (TRIG)  MIDP = 0
      NFM = 3
      IF (NPF.EQ.3)  NFM = 2
      NTF = NSF + NFM*MIDP
      NOMP = MIDP.LE.0
      SKMP = NOMP.OR.NOST
      IF (NNS.NE.4)  GO TO 130
      X(5) = 0.25*(X(1)+X(2)+X(3)+X(4))
      Y(5) = 0.25*(Y(1)+Y(2)+Y(3)+Y(4))
      Z(5) = 0.25*(Z(1)+Z(2)+Z(3)+Z(4))
      HM(5) = 0.25*(HM(1)+HM(2)+HM(3)+HM(4))
      HP(5) = 0.25*(HP(1)+HP(2)+HP(3)+HP(4))
      IF (KKK.LT.0)  GO TO 130
      T(5) = 0.25*(T(1)+T(2)+T(3)+T(4))
      DT(5) = 0.25*(DT(1)+DT(2)+DT(3)+DT(4))
      DO 110  J = 1,3
      SM(5,J) = 0.25*(SM(1,J)+SM(2,J)+SM(3,J)+SM(4,J))
  110 BM(5,J) = 0.25*(BM(1,J)+BM(2,J)+BM(3,J)+BM(4,J))
      IF (NOLD)  GO TO 130
      P(5) = 0.25*(P(1)+P(2)+P(3)+P(4))
      HW(5) = 0.25*(HW(1)+HW(2)+HW(3)+HW(4))
      DO 120  J = 1,3
  120 RHO(5,J) = 0.25*(RHO(1,J)+RHO(2,J)+RHO(3,J)+RHO(4,J))
  130 IF (NOST)  GO TO 150
      DO 140  I = 1,NTF
      DO 140  J = 1,NTF
  140 S(I,J) =  0.
  150 IF (SISM)  GO TO 170
      DO 160  I = 1,NTF
  160 F(I) = 0.
  170 IF (NOSM.OR.TRIG)  GO TO 200
      NEF1 = NEF + 1
      DO 180  L = NEF1,NTF
      M = L - 1
      DO 180  I = 1,M
  180 R(L) = R(L) - S(I,L)*R(I)
  200 DO 210  I = 1,63
  210 A(I) = 0.
      DO 220  I = 1,3
      CA(I) = CM(1,I)*ALFA(1)+CM(2,I)*ALFA(2)+CM(3,I)*ALFA(3)
      DO 220  J = 1,3
  220 C(I,J) = CM(I,J)
C
C     COMPUTE DIRECTION COSINE MATRIX TO OF LOCAL ELEMENT SYSTEM
C
      CALL QDCOS (NTRI,X,Y,Z,TO)
C
C     LOOP OVER THE NTRI TRIANGLE COMPONENTS
C
      DO 700  NT = 1,NTRI
      N1 = NT
      N2 = IPERMQ(N1)
```

```
      NC(1) = N1
      NC(2) = N2
      NC(3) = N3
      MT = MIDP/2
      NOD = 3 + MT
C
C     COMPUTE DIRECTION COSINES OF LOCAL TRIANGLE SYSTEM
C     AND THE TRIANGLE PROJECTIONS A,B ONTO IT
C
      CALL TDCOS (N1,N2,N3,X,Y,Z,A,B)
C
C     SET UP INPUTS FOR TRIANGLE SUBROUTINES
C
      DO 240  I = 1,3
      L = NC(I)
      LOC(I) = NPF*(L-1)
      HMT(I) = HM(L)
      HPT(I) = HP(L)
      IF (NOLD)  GO TO 240
      ROX = RHO(L,1)
      ROY = RHO(L,2)
      ROZ = RHO(L,3)
      RO1 = T11*ROX+T12*ROY+T13*ROZ
      RO2 = T21*ROX+T22*ROY+T23*ROZ
      RO3 = T31*ROX+T32*ROY+T33*ROZ
      H1 = HW(L)
      P1(I) = RO1*H1
      P2(I) = RO2*H1
      P3(I) = RO3*H1 + P(L)
      TEMP = T(L)
      TMOM = DT(L)*HP(L)**3/12.
      DO 230  J = 1,3
      SMT(I,J) = SM(L,J) - CA(J)*TEMP
  230 BMT(I,J) = BM(L,J) - CA(J)*TMOM
  240 CONTINUE
C
C     FORM TRANSFORMATIONS BETWEEN ELEMENT AND NODAL SYSTEMS
C
      L1 = 9*N1 - 8
      L2 = 9*N2 - 8
      CALL TRFPRD (IDIS,NEN,TDIS(L1),TDIS(L2),TDIS(19),TD1,TD2,TD3)
      IF (NPF.NE.3)
     1CALL TRFPRD (IROT,NEN,TROT(L1),TROT(L2),TROT(19),TR1,TR2,TR3)
      DO 250  I = 7,8
      TD1(I+3) = TD1(I)
      TD1(I+5) = TD1(I)
      TD2(I+3) = TD2(I)
  250 TD2(I+5) = TD2(I)
      LOC(4) = NSF + NFM*(N2-1)
      LOC(5) = NSF + NFM*(N1-1)
      N4 = LOC(4) + 3
      N5 = LOC(5) + 3
C
C     MEMBRANE CONTRIBUTION
C
  260 IF (SISM)  GO TO 320
C     MEMBRANE STIFFNESS AND/OR LOAD VECTOR
      CALL SLST (MT,KKK)
      LT = 0
      DO 300  JJ = 1,NOD
      J = JJ + JJ
      M = LOC(JJ)
      LL = MFR(JJ)
      DO 300  L = 1,LL
      M = M + 1
      LT = LT + 1
      C1 = TD1(LT)
      C2 = TD2(LT)
      IF (SILD)  F(M) = F(M) + FT(J-1)*C1 + FT(J)*C2
      IF (NOST)  GO TO 300
      KT = 0
      DO 290  II = 1,JJ
      I = II + II
      KK = MFR(II)
      IF (II.EQ.JJ)  KK = L
      H1 = ST(I-1,J-1)*C1 + ST(I-1,J)*C2
      H2 = ST(I  ,J-1)*C1 + ST(I  ,J)*C2
      N = LOC(II)
      N = LOC(II)
      DO 290  K = 1,KK
      N = N + 1
      KT = KT + 1
      SQ = S(N,M) + TD1(KT)*H1 + TD2(KT)*H2
      S(N,M) = SQ
  290 S(M,N) = SQ
  300 CONTINUE
  320 CONTINUE
  400 IF (NPF.EQ.3)  GO TO 560
C
C     PLATE BENDING CONTRIBUTION
C
      IF (SISM)  GO TO 600
C     BENDING STIFFNESS AND/OR LOAD VECTOR
      CALL SLCCT (MT,KKK)
      DO 500  JJ = 1,3
      JT = 3*JJ-3
      J = JT + 1
      DO 450  L = 1,NPF
      M = LOC(JJ) + L
      L3 = L - 3
      IF (L3.GT.0)  GO TO 420
      C3 = TD3(JT+L)
      IF (SILD)  F(M) = F(M) + FT(J)*C3
      IF (SKMP)  GO TO 450
      S4 = S(M,N4) + ST(J,10)*C3
      S5 = S(M,N5) - ST(J,11)*C3
      GO TO 430
  420 C1 = TR1(JT+L3)
      C2 = TR2(JT+L3)
      IF (SILD)  F(M) = F(M) + FT(J+1)*C1 + FT(J+2)*C2
      IF (SKMP)  GO TO 450
      S4 = S(M,N4) + ST(J+1,10)*C1 + ST(J+2,10)*C2
      S5 = S(M,N5) - ST(J+1,11)*C1 - ST(J+2,11)*C2
  430 S(M,N4) = S4
      S(N4,M) = S4
      S(M,N5) = S5
      S(N5,M) = S5
  450 CONTINUE
      IF (NOST)  GO TO 500
      DO 480  II = 1,JJ
      IT = 3*II-3
      I = IT + 1
      KK = NPF
      DO 480  L = 1,NPF
      IF (II.EQ.JJ)  KK = L
      M = LOC(JJ) + L
      L3 = L - 3
      IF (L3.GT.0)  GO TO 460
      C3 = TD3(JT+L)
      H1 = ST(I  ,J)*C3
      H2 = ST(I+1,J)*C3
      H3 = ST(I+2,J)*C3
      GO TO 470
  460 C1 = TR1(JT+L3)
      C2 = TR2(JT+L3)
      H1 = ST(I  ,J+1)*C1 + ST(I  ,J+2)*C2
      H2 = ST(I+1,J+1)*C1 + ST(I+1,J+2)*C2
      H3 = ST(I+2,J+1)*C1 + ST(I+2,J+2)*C2
  470 N = LOC(II)
      DO 480  K = 1,KK
      N = N + 1
      K3 = K - 3
      K1 = IT + K
      K2 = IT + K3
      IF (K3.LE.0)  SQ = S(N,M) + TD3(K1)*H1
      IF (K3.GT.0)  SQ = S(N,M) + TR1(K2)*H2 + TR2(K2)*H3
      S(N,M) = SQ
  480 S(M,N) = SQ
  500 CONTINUE
      IF (NOMP)  GO TO 700
```

```
      IF (NOLD)  GO TO 540
      F(N4) = F(N4) + FT(10)
      F(N5) = F(N5) - FT(11)
  540 IF (NOST)  GO TO 700
      S(N4,N4) = S(N4,N4) + ST(10,10)
      S(N5,N5) = S(N5,N5) + ST(11,11)
      S(N4,N5) = S(N4,N5) - ST(10,11)
      S(N5,N4) = S(N4,N5)
      GO TO 700
  560 IF (NOLD)  GO TO 700
      FL = (P3(1)+P3(2)+P3(3))*(A(3)*B(2)-A(2)*B(3))/12.
      JT = 0
      DO 580  JJ = 1,2
      DO 580 L=1,3
      JT=JT+1
      M = LOC(JJ) + L
  580 F(M) = F(M) + FL*TD3(JT)
  600 CONTINUE
  700 CONTINUE
      IF (SISM.OR.TRIG)  GO TO 900
C
C    CHECK FOR POSSIBLE INTERNAL STIFFNESS SINGULARITY (FLAT
C     OR NEARLY FLAT QUADRILATERAL WHEN NPF = 3 OR 6)
      IF ((NPF.EQ.5).OR.NOST)  GO TO 730
      DO 720  N = 3,6,3
      IF (NEF.NE.N)  GO TO 720
      M = 5*N
      M1 = M - 1
      M2 = M - 2
      IF (S(M,M).GT.(S(M1,M1)+S(M2,M2))*1.0E-08)  GO TO 720
      DO 710  I = 1,NTF
      S(I,M) = 0.
  710 S(M,I) = 0.
  720 CONTINUE
C
C     CONDENSATION OF INTERNAL DEGREES OF FREEDOM
C
  730 NIF = NTF - NEF
      DO 900  N = 1,NIF
      K = NTF - N
      L = K + 1
      PIVOT = S(L,L)
      FL = F(L)
      IF (PIVOT.LE.0.)  GO TO 900
      F(L) = F(L)/PIVOT
      DO 780  I = 1,K
      G = S(I,L)
      IF (G)  740,780,740
  740 IF (NOST)  GO TO 770
      G = G/PIVOT
      S(I,L) = G
      DO 760  J = I,K
      SQ =  S(I,J) - G*S(L,J)
      S(I,J) = SQ
  760 S(J,I) = SQ
  770 F(I) = F(I) - G*FL
  780 CONTINUE
  900 CONTINUE
  900 RETURN
      END



      SUBROUTINE QDCOS (N,X,Y,Z,T)
C
C     THIS SUBROUTINE COMPUTES THE DIRECTION COSINES OF THE LOCAL
C     ELEMENT SYSTEM OF A QUADRILATERAL (N=4) OR SINGLE TRIANGLE (N=1)
C
      DIMENSION X(1), Y(1), Z(1), T(1)
      X1 = X(2)+X(3)-X(N)-X(1)
      Y1 = Y(2)+Y(3)-Y(N)-Y(1)
      Z1 = Z(2)+Z(3)-Z(N)-Z(1)
      X2 = X(3)+X(N)-X(1)-X(2)
      Y2 = Y(3)+Y(N)-Y(1)-Y(2)
      Z2 = Z(3)+Z(N)-Z(1)-Z(2)
      S1 = X1**2+Y1**2+Z1**2
      C  = (X1*X2+Y1*Y2+Z1*Z2)/S1
      X2 = X2 - C*X1
      Y2 = Y2 - C*Y1
      Z2 = Z2 - C*Z1
      S1 = SQRT (S1)
      S2 = SQRT (X2**2+Y2**2+Z2**2)
      X1 = X1/S1
      Y1 = Y1/S1
      Z1 = Z1/S1
      X2 = X2/S2
      Y2 = Y2/S2
      Z2 = Z2/S2
      T(1) = X1
      T(2) = X2
      T(3) = Y1*Z2-Y2*Z1
      T(4) = Y1
      T(5) = Y2
      T(6) = Z1*X2-Z2*X1
      T(7) = Z1
      T(8) = Z2
      T(9) = X1*Y2-X2*Y1
      RETURN
      END



      SUBROUTINE TOCOS (N1,N2,N3,X,Y,Z,A,B)
C
C     THIS SUBROUTINE COMPUTES THE DIRECTION COSINES OF THE LOCAL
C     SYSTEM AND THE PROJECTED DIMENSIONS OF A TRIANGLE COMPONENT
C
      COMMON /TRANSF/  T1(3),T2(3),T3(3), T(9)
      EQUIVALENCE (T11,T1),(T12,T1(2)),(T13,T1(3)),(T21,T2),(T22,T2(2)),
     1 (T23,T2(3)),(T31,T3),(T32,T3(2)),(T33,T3(3))
      DIMENSION  X(1), Y(1), Z(1), A(1), B(1)
      A1 = X(N1)-X(N3)
      B1 = Y(N1)-Y(N3)
      C1 = Z(N1)-Z(N3)
      A2 = X(N2)-X(N3)
      B2 = Y(N2)-Y(N3)
      C2 = Z(N2)-Z(N3)
      T31 = B1*C2-B2*C1
      T32 = C1*A2-C2*A1
      T33 = A1*B2-A2*B1
      S = SQRT (T31**2+T32**2+T33**2)
      T31 = T31/S
      T32 = T32/S
      T33 = T33/S
      T11 = T33*T(5)-T32*T(8)
      T12 = T31*T(8)-T33*T(2)
      T13 = T32*T(2)-T31*T(5)
      S = SQRT (T11**2+T12**2+T13**2)
      T11 = T11/S
      T12 = T12/S
      T13 = T13/S
      T21 = T13*T32-T12*T33
      T22 = T11*T33-T13*T31
      T23 = T12*T31-T11*T32
      A(1) = -T11*A2-T12*B2-T13*C2
      A(2) =  T11*A1+T12*B1+T13*C1
      B(1) =  T21*A2+T22*B2+T23*C2
      B(2) = -T21*A1-T22*B1-T23*C1
      A(3) = -A(1)-A(2)
      B(3) = -B(1)-B(2)
      RETURN
      END
```

```
      SUBROUTINE TRFPRD (M,NEN,Q1,Q2,Q3,P1,P2,P3)
C
C     THIS SUBROUTINE GENERATES THE TRANSFORMATIONS RELATING A LOCAL
C     SUBTRIANGLE SYSTEM TO THE NODAL DIS/ROT SYSTEMS AT ITS 3 CORNERS
C
      COMMON /TRANSF/  T1(3),T2(3),T3(3), T(9)
      DIMENSION  P1(1), P2(1), P3(1), Q1(1), Q2(1), Q3(1)
      EQUIVALENCE (T11,T1),(T12,T1(2)),(T13,T1(3)),(T21,T2),(T22,T2(2)),
     1 (T23,T2(3)),(T31,T3),(T32,T3(2)),(T33,T3(3))
      DO 300  I = 1,3
      J = I + 3
      K = I + 6
      P1(I) = T1(I)
      P1(J) = T1(I)
      P2(I) = T2(I)
      P2(J) = T2(I)
      P3(I) = T3(I)
      P3(J) = T3(I)
      IF (NEN.NE.4)  GO TO 150
      CI = T(I)
      CJ = T(J)
      CK = T(K)
      IF (M)  260,260,240
  150 IF (M)  180,180,200
  180 P1(K) = T1(I)
      P2(K) = T2(I)
      P3(K) = T3(I)
      GO TO 300
  200 CI = Q3(I)
      CJ = Q3(J)
      CK = Q3(K)
  240 P1(I) = T11*Q1(I) + T12*Q1(J) + T13*Q1(K)
      P1(J) = T11*Q2(I) + T12*Q2(J) + T13*Q2(K)
      P2(I) = T21*Q1(I) + T22*Q1(J) + T23*Q1(K)
      P2(J) = T21*Q2(I) + T22*Q2(J) + T23*Q2(K)
      P3(I) = T31*Q1(I) + T32*Q1(J) + T33*Q1(K)
      P3(J) = T31*Q2(I) + T32*Q2(J) + T33*Q2(K)
  260 P1(K) = T11*CI    + T12*CJ    + T13*CK
      P2(K) = T21*CI    + T22*CJ    + T23*CK
      P3(K) = T31*CI    + T32*CJ    + T33*CK
  300 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SLST (M,KKK)
C
C     THIS SUBROUTINE FORMS THE PLANE STRESS STIFFNESS MATRIX AND/OR
C     THE CONSISTENT LOAD VECTOR OF A LINEAR STRAIN TRIANGLE (LST) WITH
C     6, 5 OR 4 NODAL POINTS, OR OF A CONSTANT STRAIN TRIANGLE (CST).
C     LINEAR ELASTIC ANISOTROPIC MATERIAL
C
C * * * * * * * * * * * * * * * INPUTS  * * * * * * * * * * * * * * * *
C
C       M        NUMBER OF MIDPOINTS INCLUDED AS NODAL POINTS (M=3,2,1
C                FOR LST, M=0 FOR CST). NOTE.. MIDPOINTS 4-5-6 ARE
C                LOCATED ON THE SIDES 2-3, 3-1 AND 1-2, RESPECTIVELY.
C
C       KKK      OPERATION FLAG
C                  KKK LE 0 = FORM STIFFNESS MATRIX AND LOAD VECTOR.
C                  KKK GT 0 = FORM LOAD VECTOR ONLY.
C
C   A(I),B(I)    I=1...3   PROJECTIONS OF SIDES 2-3, 3-1 AND 1-2 ONTO
C                X AND -Y, RESPECTIVELY.
C
C     C(I,J)     I=1...3, J=1...3    PLANE STRESS MATERIAL MATRIX.
C
C       H(I)     I=1...3   CORNER THICKNESSES (LINEAR VARIATION ASSUMED).
C
C PX(I),PY(I)    I=1...3   CORNER VALUES OF X-Y COMPONENTS OF BODY FORCES
C                          PER UNIT OF ELEMENT AREA (LINEAR VARIATION ASSUMED).
C
C   SMT(I,J)     I=1...3, J=1...3   INITIAL MEMBRANE STRESS COMPONENTS
C                SIG-XX (J=1), SIG-YY (J=2) AND SIG-XY (J=3 AT THE
C                CORNERS I=1,2,3  (LINEAR VARIATION ASSUMED).
C
C
C * * * * * * * * * * * * * * * OUTPUTS * * * * * * * * * * * * * * * *
C
C   ST(I,J)      I=1..NDF, J=1..NDF  WITH NDF (NUMBER OF DOF) = 6+2*M, IS
C                THE ELEMENT STIFFNESS MATRIX ASSOCIATED WITH THE NODAL
C                DISPLACEMENT ORDERING
C                    U(1),V(1),U(2),V(2),U(3), ..... V(3+M)
C                WHERE U(4), ... V(3+M), IF M GT 0, ARE DEVIATIONS
C                FROM LINEARITY AT THE MIDPOINTS 1...M.
C
C    FT(I)       I=1..NDF   CONSISTENT NODAL FORCE VECTOR ASSOCIATED
C                WITH THE NODAL DISPLACEMENT ORDERING DESCRIBED ABOVE.
C
C
      COMMON /TRIARG/ A(3),B(3), H(3),  HPT(3), C(3,3), SMT(3,3),
     1 BMT(3,3), FT(12),     PX(3),PY(3),PT(3),RM(3), ST(12,12)
      DIMENSION Q(3,3), QA(3), QB(3), A4(3), B4(3), IPERM(3),
     1 SXX(3),SYY(3), SXY(3)
      EQUIVALENCE (SXX,SMT),(SYY,SMT(4)),(SXY,SMT(7))
      LOGICAL NOS
      DATA  IPERM /2,3,1/
      NOS = KKK.GT.0
      NDF = 6 + 2*M
      AREA = A(3)*B(2)-A(2)*B(3)
      SUMH = H(1)+H(2)+H(3)
      HO = SUMH/3.
      IF (HO) 500,500,140
  140 PXS = PX(1)+PX(2)+PX(3)
      PYS = PY(1)+PY(2)+PY(3)
      SXXH = 0.
      SYYH = 0.
      SXYH = 0.
      DO 150  I = 1,3
      CH = (SUMH+H(I))/24.
      SXXH = SXXH + CH*SXX(I)
      SYYH = SYYH + CH*SYY(I)
  150 SXYH = SXYH + CH*SXY(I)
      FAC = HO/(2.*AREA)
      C11 = C(1,1)*FAC
      C22 = C(2,2)*FAC
      C33 = C(3,3)*FAC
      C12 = C(1,2)*FAC
      C13 = C(1,3)*FAC
      C23 = C(2,3)*FAC
      DO 200  J = 1,3
      L = J + J
      FT(L-1) = (PXS+PX(J))*AREA/24. - (B(J)*SXXH+A(J)*SXYH)
      FT(L)   = (PYS+PY(J))*AREA/24. - (A(J)*SYYH+B(J)*SXYH)
      IF (NOS)  GO TO 200
  180 DO 190  I = 1,J
      K = I + I
      AA = A(I)*A(J)
      BB = B(I)*B(J)
      AB = A(I)*B(J)
      BA = B(I)*A(J)
      ABA = AB+BA
      ST(K-1,L-1) = C11*BB + C33*AA + C13*ABA
      ST(K  ,L  ) = C22*AA + C33*BB + C23*ABA
      ST(K-1,L  ) = C12*BA + C33*AB + C13*AA + C23*BB
  190 ST(K  ,L-1) = C12*AB + C33*BA + C13*BB + C23*AA
  200 CONTINUE
      IF (M) 350,350,220
  220 DO 240  I = 1,3
      A4(I) = 4.*A(I)
      B4(I) = 4.*B(I)
      J = IPERM(I)
      K = IPERM(J)
      R = H(I)/HO
      Q(I,I) = 0.1+R/15.
```

```
      Q(J,K) = 0.1-R/60.
  240 Q(K,J) = Q(J,K)
      DO 300  J = 1,M
      J1 = IPERM(J)
      J2 = IPERM(J1)
      L = J + J + 6
      FX = 0.
      FY = 0.
      DO 250  N = 1,3
      Q1 = Q(N,J1)
      Q2 = Q(N,J2)
      QA(N) = Q2*A4(J1)+Q1*A4(J2)
      QB(N) = Q2*B4(J1)+Q1*B4(J2)
      FX = FX - QB(N)*SXX(N)-QA(N)*SXY(N)
  250 FY = FY - QA(N)*SYY(N)-QB(N)*SXY(N)
      FT(L-1) = (PXS-PX(J))*AREA/12. + FX*HO/2.
      FT(L)   = (PYS-PY(J))*AREA/12. + FY*HO/2.
      IF (NOS)  GO TO 300
      SUMQA = QA(1)+QA(2)+QA(3)
      SUMQB = QB(1)+QB(2)+QB(3)
      JM = J + 3
      DO 290  I = 1,JM
      K = I + I
      IF (I.GT.3) GO TO 260
      AA = A(I)*SUMQA
      AB = A(I)*SUMQB
      BA = B(I)*SUMQA
      BB = B(I)*SUMQB
      GO TO 280
  260 I1 = IPERM(I-3)
      I2 = IPERM(I1)
      AA = A4(I2)*QA(I1)+A4(I1)*QA(I2)
      AB = A4(I2)*QB(I1)+A4(I1)*QB(I2)
      BA = B4(I2)*QA(I1)+B4(I1)*QA(I2)
      BB = B4(I2)*QB(I1)+B4(I1)*QB(I2)
  280 ABA = AB+BA
      ST(K-1,L-1) = C11*BB + C33*AA + C13*ABA
      ST(K  ,L  ) = C22*AA + C33*BB + C23*ABA
      ST(K-1,L  ) = C12*BA + C33*AB + C13*AA + C23*BB
  290 ST(K  ,L-1) = C12*AB + C33*BA + C13*BB + C23*AA
  300 CONTINUE
  350 DO 400  I = 2,NDF
      DO 400  J = 1,I
  400 ST(I,J) = ST(J,I)
  500 RETURN
      END
```

```
      SUBROUTINE SLCCT (M,KKK)
C
C     THIS SUBROUTINE FORMS THE PLATE BENDING STIFFNESS AND/OR THE
C     CONSISTENT LOAD VECTOR OF A LINEAR CURVATURE COMPATIBLE TRIANGLE
C     (LCCT) WITH 6, 5, 4 OR 3 NODAL POINTS
C
C * * * * * * * * * * * * * * * INPUTS  * * * * * * * * * * * * * * * *
C
C      M        NUMBER OF MIDPOINT DEGREES OF FREEDOM (M =3,2,1,0).
C               NOTE.. MIDPOINTS 4-5-6 (IF INCLUDED) ARE LOCATED ON
C               SIDES 2-3, 3-1 AND 1-2, RESPECTIVELY.
C
C     KKK       OPERATION FLAG
C                 KKK LE 0 = FORM STIFFNESS MATRIX AND LOAD VECTOR.
C                 KKK GT 0 = FORM LOAD VECTOR ONLY.
C
C  A(I),B(I)    I=1...3   PROJECTIONS OF SIDES 2-3, 3-1 AND 1-2 ONTO
C               X AND -Y, RESPECTIVELY.
C
C   C(I,J)      I=1...3, J=1...3   PLANE STRESS MATERIAL MATRIX.
C
C     H(I)      I=1...3   CORNER THICKNESSES (LINEAR VARIATION ASSUMED).
C
C     PT(I)     I=1...3   CORNER VALUES OF LATERAL DISTRIBUTED LOAD
C               (LINEAR VARIATION ASSUMED).
C
C   BMT(I,J)    I=1...3, J=1...3   INITIAL BENDING MOMENT COMPONENTS
C               MOM-XX (J=1), MOM-YY (J=2) AND MOM-XY (J=3) AT THE
C               CORNERS I=1...3  (LINEAR VARIATION ASSUMED).
C
C
C * * * * * * * * * * * * * * * OUTPUTS * * * * * * * * * * * * * * * *
C
C   ST(I,J)     I=1..NDF, J=1..NDF  WITH NDF (NUMBER OF DOF) = 9+M, IS
C               THE ELEMENT STIFFNESS MATRIX ASSOCIATED WITH THE NODAL
C               DISPLACEMENT ORDERING
C                 W(1),RX(1),RY(1),W(2), .... RY(3),RM(1), ... RM(M)
C               WHERE RM(1), ... RM(M), IF M GT O, ARE MIDPOINT
C               DEVIATIONS FROM NORMAL SLOPE LINEARITY
C
C    FT(I)      I=1..NDF   CONSISTENT NODAL FORCE VECTOR ASSOCIATED
C               WITH THE NODAL DISPLACEMENT ORDERING DESCRIBED ABOVE.
C
C
      COMMON /TRIARG/ A(3),B(3), HMT(3), H(3),  C(3,3), SMT(3,3),
     1 BMT(3,3), FT(12),     PX(3),PY(3),PT(3),RM(3), ST(12,12)
      DIMENSION  P(21,12), G(21), Q(3,6), QB(3,6), T(3), U(3), HT(3),
     1 TX(3), TY(3), IPERM(3), XM(3,3), XMO(3)
      EQUIVALENCE (CM11,C(1)),(CM12,C(2)),(CM13,C(3)),(CM22,C(5)),
     1 (CM23,C(6)),(CM33,C(9))
      LOGICAL NOS, FLAT
      DATA IPERM/2,3,1/
      HO = (H(1)+H(2)+H(3))/3.
      IF (HO.LE.O.)  GO TO 1000
      NDF = 9 + M
      NOS = KKK.GT.0
      FLAT = (H(1).EQ.H(2)).AND.(H(2).EQ.H(3))
      AREA = A(3)*B(2)-A(2)*B(3)
      FAC = HO**3*AREA/864.
      PTF = AREA/6480.
      T(3) = 1.
      DO 150  I = 1,3
      J = IPERM(I)
      K = IPERM(J)
      X = A(I)**2+B(I)**2
      U(I) = -(A(I)*A(J)+B(I)*B(J))/X
      X = SQRT(X)
      Y = 2.*AREA/X
      HT(I) =  2.*Y
      TX(I) =  Y*A(I)/X
      TY(I) = -Y*B(I)/X
      A1 = A(I)/AREA
      A2 = A(J)/AREA
      B1 = B(I)/AREA
      B2 = B(J)/AREA
      Q(1,I)   = B1*B1
      Q(2,I)   = A1*A1
      Q(3,I)   = 2.*A1*B1
      Q(1,I+3) = 2.*B1*B2
      Q(2,I+3) = 2.*A1*A2
      Q(3,I+3) = 2.*(A1*B2+A2*B1)
      DO 120  N = 1,3
  120 XM(N,I) = BMT(N,I)*AREA/72.
      IF (FLAT)  GO TO 150
      DO 140  N = 1,3
      L = IPERM(N)
      T(1) = H(N)/HO
      T(2) = H(L)/HO
      IF (T(1).GT.O.)  XM(N,I) = XM(N,I)/T(1)**3
      C1 = T(I)
      C2 = T(J)
      C3 = T(K)
      C4 = C2+C3
      C11 = C1*C1
      C23 = C2*C3
      C5 = C4*(3.*C1+C4) + 6.*C11 - 2.*C23
```

```
      C6 = C5 + 3.*C4*C4 - 4.*(C11+C23)
      QB(N,I  ) = (C1*(10.*C11-3.*C23)+C4*C5)/17.5 - 2.0
  140 QB(N,I+3) = (C1*(C11-2.*C23)+C4*C6)/35.0 - 1.0
  150 CONTINUE
      DO 200  I = 1,3
      J = IPERM(I)
      K = IPERM(J)
      II = 3*I
      JJ = 3*J
      KK = 3*K
      A1 = A(I)
      A2 = A(J)
      A3 = A(K)
      B1 = B(I)
      B2 = B(J)
      B3 = B(K)
      U1 = U(I)
      U2 = U(J)
      U3 = U(K)
      W1 = 1.-U1
      W2 = 1.-U2
      W3 = 1.-U3
      B1D = B1 + B1
      B2D = B2 + B2
      B3D = B3 + B3
      A1D = A1 + A1
      A2D = A2 + A2
      A3D = A3 + A3
      C21 = B1-B3*U3           + TX(K)
      C22 = -B1D+B2*W2+B3*U3 + TX(J)-TX(K)
      C31 = A1-A3*U3           + TY(K)
      C22 = -B1D+B2*W2+B3*U3 + TX(J)-TX(K)
      C31 = A1-A3*U3           + TY(K)
      C32 = -A1D+A2*W2+A3*U3 + TY(J)-TY(K)
      C51 = B3*W3-B2           + TX(K)
      C52 = B2D-B3*W3-B1*U1    + TX(I)-TX(K)
      C61 = A3*W3-A2           + TY(K)
      C62 = A2D-A3*W3-A1*U1    + TY(I)-TY(K)
      C81 = B3-B2D-B2*U2       + TX(J)
      C82 = B1D-B3+B1*W1       + TX(I)
      C91 = A3-A2D-A2*U2       + TY(J)
      C92 = A1D-A3+A1*W1       + TY(I)
      P1 = PT(I)*PTF
      P2 = PT(J)*PTF
      P3 = PT(K)*PTF
      U37 = 7.*U3
      W27 = 7.*W2
      W24 = 4.*W2
      U34 = 4.*U3
      C1 = 54.+W27
      C2 = 54.+U37
      C3 = 15.+W24
      C4 = 39.+U37
      C5 = 39.+W27
      C6 = 15.+U34
      TXS = TX(J)+TX(K)
      TYS = TY(J)+TY(K)
      FT(II-2) = 6.*((90.+U37+W27)*P1+(36.+U37+W24)*P2+(36.+U34+W27)*P3)
      FT(II-1) = (C1*B2-C2*B3+7.*TXS)*P1 + (C3*B2-C4*B3+4.*TXS+
     1 3.*TX(K))*P2 + (C5*B2-C6*B3+4.*TXS+3.*TX(J))*P3
      FT(II)    = (C1*A2-C2*A3+7.*TYS)*P1 + (C3*A2-C4*A3+4.*TYS+
     1 3.*TY(K))*P2 + (C5*A2-C6*A3+4.*TYS+3.*TY(J))*P3
      FT(K+9)   = (7.*(P1+P2)+4.*P3)*HT(K)
      XMO(I) = (XM(1,I)+XM(2,I)+XM(3,I))/3.
      DO 200  N = 1,3
      L = 6*(I-1) + N
      Q11 = Q(N,I)
      Q22 = Q(N,J)
      Q33 = Q(N,K)
      Q12 = Q(N,I+3)
      Q23 = Q(N,J+3)
      Q31 = Q(N,K+3)
      Q2333 = Q23-Q33
      Q3133 = Q31-Q33
      P(L   ,II-2) = 6.*(-Q11+W2*Q33+U3*Q2333)
      P(L   ,II-1) = C21*Q23+C22*Q33-B3D*Q12+B2D*Q31
      P(L   ,II  ) = C31*Q23+C32*Q33-A3D*Q12+A2D*Q31
      P(L   ,JJ-2) = 6.*(Q22+W3*Q2333)
      P(L   ,JJ-1) = C51*Q2333+B3D*Q22
      P(L   ,JJ  ) = C61*Q2333+A3D*Q22
      P(L   ,KK-2) = 6.*(1.+U2)*Q33
      P(L   ,KK-1) = C81*Q33
      P(L   ,KK  ) = C91*Q33
      P(L   ,I+9 ) = 0.
      P(L   ,J+9 ) = HT(J)*Q33
      P(L   ,K+9 ) = HT(K)*Q2333
      P(L+3 ,II-2) = 6.*(Q11+U3*Q3133)
      P(L+3 ,II-1) = C21*Q3133-B3D*Q11
      P(L+3 ,II  ) = C31*Q3133-A3D*Q11
      P(L+3 ,JJ-2) = 6.*(-Q22+U1*Q33+W3*Q3133)
      P(L+3 ,JJ-1) = C51*Q31+C52*Q33+B3D*Q12-B1D*Q23
      P(L+3 ,JJ  ) = C61*Q31+C62*Q33+A3D*Q12-A1D*Q23
      P(L+3 ,KK-2) = 6.*(1.+W1)*Q33
      P(L+3 ,KK-1) = C82*Q33
      P(L+3 ,KK  ) = C92*Q33
      P(L+3 ,I+9 ) = HT(I)*Q33
      P(L+3 ,J+9 ) = 0.
      P(L+3 ,K+9 ) = HT(K)*Q3133
      P(N+18,II-2) = 2.*(Q11+U3*Q12+W2*Q31)
      P(N+18,KK-1) = ((B1D-B2D)*Q33+C82*Q23+C81*Q31)/3.
      P(N+18,KK  ) = ((A1D-A2D)*Q33+C92*Q23+C91*Q31)/3.
  200 P(N+18,K+9 ) = HT(K)*Q12/3.
  300 DO 400  J = 1,NDF
      DO 340  L = 1,3
      II = L
      KK = L + 18
      P3 = P(KK,J)
      G(KK) = 0.
      DO 340  N = 1,3
      I = IPERM(N)
      JJ = II + 3
      P1 = P(II,J)
      P2 = P(JJ,J)
      SUM = P1 + P2 + P3
      G1 = SUM + P1
      G2 = SUM + P2
      G3 = SUM + P3
      IF (FLAT)  GO TO 320
      G1 = G1 + QB(N,1)*P1 + QB(N,6)*P2 + QB(N,5)*P3
      G2 = G2 + QB(N,6)*P1 + QB(N,2)*P2 + QB(N,4)*P3
      G3 = G3 + QB(N,5)*P1 + QB(N,4)*P2 + QB(N,3)*P3
  320 G(II) = G1
      G(JJ) = G2
      G(KK) = G3 + G(KK)
      II = II + 6
  340 FT(J) = FT(J) - XM(N,L)*G1 - XM(I,L)*G2 - XMO(L)*G3
      IF (NOS)  GO TO 400
      DO 360  N = 1,19,3
      G1 = G(N)
      G2 = G(N+1)
      G3 = G(N+2)
      G(N)   = CM11*G1 + CM12*G2 + CM13*G3
      G(N+1) = CM12*G1 + CM22*G2 + CM23*G3
  360 G(N+2) = CM13*G1 + CM23*G2 + CM33*G3
      DO 390  I = 1,J
      X = 0.
      DO 380  N = 1,21
  380 X = X + G(N)*P(N,I)
      X = X*FAC
      ST(I,J) = X
  390 ST(J,I) = X
  400 CONTINUE
 1000 RETURN
      END
```

```
      SUBROUTINE THKSHL
C
      COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
      COMMON /JUNK/ MM,L,K,NTAG,SIG(200)
      COMMON A(1)
C
      IF(NPAR(1).EQ.0) GO TO 500
      N6=N5+NPAR(3)
      N7=N6+NPAR(3)
      N8=N7+NPAR(3)
      N9=N8+NPAR(3)
      N10=N9 +NPAR(4)
      N11=N10+NPAR(4)
      N12=N11+NPAR(4)
      N13=N12+NPAR(4)
      N14=N13+63*63
      N15=N14+42*63
      IF(N15.GT.MTOT) CALL ERROR(N15-MTOT)
      CALL ELST3D (A(N13),A(N14),NPAR(2),NPAR(3),NPAR(4),A(N1),A(N2),
     1A(N3),A(N4),A(N5),A(N6),A(N7),A(N8),A(N9),A(N10),A(N11),A(N12),
     2NUMNP)
      RETURN
C
  500 CONTINUE
      CALL ST3D16 (A(N3),A(N1),NEQ,NPAR(2))
C
      RETURN
C
      END
```

```
      SUBROUTINE ELST3D (S,STR,N3DEL,NMAT,NLD,ID,X,Y,Z,EE,ENU,RHO,ALPT,
     1KTYPE,PR,YREF,NFACE,NUMNP)
C
C     STIFNESS SUBROUTINE FOR 48 D.F. ISOPARAMETRIC3D THICK SHELL ELEM.
C     LINEAR ELASTIC ISOTROPIC MATERIAL
C     NINT*NINT*(NINT-1)    GAUSSIAN INTEGRATION RULE USED
C     NINT=1,2,3,4
C
      COMMON /ELPAR/ NPAR(14),NUMN ,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
      COMMON/EM/LM(48),ND,NDM,RF(63,4),XM(48),SA(21,3),DL(16),TT(48),
     1XLF(4),YLF(4),ZLF(4),TLF(4),PLF(4),NP(16),INP(16),DETT(15)
      COMMON /JUNK/ E1,E2,E3,DET,MLD(4),KLD(4),MULT(4),A(3,3),
     1P(3,21),B(3,3),XX(16,3),Q(19),REFT,INEL,ININT,IMAT,IINC,TTEMP,NEL,
     2ML,NINT,MAT,INC,TAG,TEMP,SKIP,I,J,K,L,FAC,CC1,CC2,CC3,CC4,G,DEN,
     3FACT,GT,GG,C1,C2,C3,C,K1,K2
      COMMON/GASS/XK(4,4),WGT(4,4),IPERM(3)
C
      DIMENSION S(63,63),STR(42,63),KTYPE(1),PR(1),YREF(1),NFACE(1),
     1X(1),Y(1),Z(1),ID(NUMNP,1),EE(1),ENU(1),RHO(1),ALPT(1)
      DIMENSION STPTS(7,3)
C
      DATA IPERM / 2,3,1 /
      DATA XK /      0.,              0.,              0.,              0.,
     1 -.5773502691896, .5773502691896,              0.,              0.,
     2 -.7745966692415, .0000000000000, .7745966692415,              0.,
     3 -.8611363115941,-.3399810435849, .3399810435849, .8611363115941/
      DATA WGT / 2.000,             0.,              0.,              0.,
     1 1.0000000000000,1.0000000000000,              0.,              0.,
     2  .5555555555556, .8888888888889, .5555555555556,              0.,
     3  .3478548451375, .6521451548625, .6521451548625, .3478548451375/
      DATA STPTS / 0. , 1. ,-1. , 0. , 0. , 0. , 0. ,
     1             0. , 0. , 0. , 1. ,-1. , 0. , 0. ,
     2             0. , 0. , 0. , 0. , 0. , 1. ,-1. /
C
      DO 5 I=1,350
    5 LM(I)=0
C
C     MATERIAL PROPERTIES
C
      WRITE (6,1300)
      DO 10 I=1,NMAT
      READ  (5,1001) N,EE(N),ENU(N),RHO(N),ALPT(N)
   10 WRITE (6,2001) N,EE(N),ENU(N),RHO(N),ALPT(N)
C
C     ELEMENT DISTRIBUTED LOAD CARDS
C
      IF(NLD) 23,23,15
   15 WRITE (6,1302)
      DO 16 I=1,NLD
      READ  (5,1002) N,KTYPE(N),PR(N),YREF(N),NFACE(N)
   16 WRITE (6,2002) N,KTYPE(N),PR(N),YREF(N),NFACE(N)
C
   23 READ  (5,1003) REFT,GRAV,PLF,TLF,XLF,YLF,ZLF
      WRITE (6,2003) REFT,PLF,TLF,XLF,YLF,ZLF
      GRAV=1.0
      WRITE (6,1301)
      NEL=0
   30 READ  (5,1000)      INEL,ININT,IMAT,IINC,MLD,TTEMP,INP
      DO 39 I=1,4
   39 MULT(I)=1
      IF(IINC.EQ.0) IINC=1
      IF(IMAT.EQ.0) IMAT=1
   40 NEL=NEL+1
      ML=INEL-NEL
      IF(ML) 50,55,60
   50 WRITE (6,4003) INEL
      STOP
   55 DO 56 I=1,16
   56 NP(I)=INP(I)
      NINT=ININT
      MAT=IMAT
      INC=IINC
      TAG=1HI
      TEMP=TTEMP
      SKIP=999.
      IF(NINT) 33,33,57
   33 NINT=IABS(NINT)
      SKIP=1.
      IF(NINT.EQ.0) SKIP=0.
   57 CONTINUE
      DO 59 I=1,4
      KLD(I)=IABS(MLD(I))
      IF(MLD(I)) 58,58,59
   58 MULT(I)=0
   59 CONTINUE
      GO TO 62
C
   60 DO 61 I=1,16
   61 NP(I)=NP(I)+INC
      TAG=1H
      DO 64 I=1,4
   64 KLD(I)=KLD(I)*MULT(I)
   62 WRITE (6,2000) NEL,NP,NINT,MAT,TAG,KLD,TEMP
C
C
      DO 80  I = 1,16
      K=NP(I)
      XX(I,1)=X(K)
      XX(I,2)=Y(K)
   80 XX(I,3)=Z(K)
      K=MAT
      CC4=ALPT(K)*(TEMP-REFT)
      FAC = EE(K)/((1.-2.*ENU(K))*(1.+ENU(K)))
      FACT=FAC*ALPT(K)*(TEMP-REFT)*(1.+ENU(K))
      IF(SKIP) 70,70,63
   63 SKIP=SKIP-1.
      CC1=1.-ENU(K)
      CC2=ENU(K)
      CC3=.5-ENU(K)
      DEN = RHO(K)
C
      L3=63*63
      DO 100 I=1,L3
  100 S(I)=0.
```

```
      DO 110 I=1,48
  110 TT(I)=0.
      DO 120 I=1,16
  120 DL(I)=0.
C
C     LOOP OVER NINT**3 INTEGRATION POINTS
C
      MINT=NINT-1
      DO 300  LX = 1,NINT
      E1=XK(LX,NINT)
      DO 300  LY = 1,NINT
      E2=XK(LY,NINT)
      DO 300  LZ = 1,MINT
      E3=XK(LZ,MINT)
C
      CALL FUNCT(1,SA)
C
      G = WGT(LX,NINT)*WGT(LY,NINT)*WGT(LZ,MINT)
      GT=G
      GG=G*DET*DEN
      G=G*FAC/DET
      C1=G*CC1
      C2=G*CC2
      C3=G*CC3
      L=0
      DO 310 I=1,16
      DL(I)=DL(I) + GG*Q(I)
      DO 310 K=1,3
      L=L+1
  310 TT(L)=TT(L) + GT*SA(I,K)
C
C     ADD CONTRIBUTION TO STIFFNESS MATRIX
C
      DO 300 I=1,21
      K3 = 3*I
      K2 = K3 - 1
      K1 = K2 - 1
      UI=SA(I,1)
      VI=SA(I,2)
      WI=SA(I,3)
      DO 300 J=I,21
      L3 = 3*J
      L2 = L3 - 1
      L1 = L2 - 1
      UJ=SA(J,1)
      VJ=SA(J,2)
      WJ=SA(J,3)
      UU=UI*UJ
      VV=VI*VJ
      WW=WI*WJ
      UV=UI*VJ
      VU=VI*UJ
      UW=UI*WJ
      WU=WI*UJ
      VW=VI*WJ
      WV=WI*VJ
      S(K1,L1) = S(K1,L1) + C1*UU + C3*(VV+WW)
      S(K2,L2) = S(K2,L2) + C1*VV + C3*(WW+UU)
      S(K3,L3) = S(K3,L3) + C1*WW + C3*(UU+VV)
      S(K1,L2) = S(K1,L2) + C2*UV + C3*VU
      S(K1,L3) = S(K1,L3) + C2*UW + C3*WU
      S(K2,L3) = S(K2,L3) + C2*VW + C3*WV
      IF (I.EQ.J)  GO TO 300
      S(K2,L1) = S(K2,L1) + C2*VU + C3*UV
      S(K3,L1) = S(K3,L1) + C2*WU + C3*UW
      S(K3,L2) = S(K3,L2) + C2*WV + C3*VW
  300 CONTINUE
C
C     FORM STRAIN MATRIX
C
      DO 305 I=1,2646
  305 STR(I)=0.
      L2=1
      DO 405 L1=1,7
      E1=STPTS(L1,1)
      E2=STPTS(L1,2)
      E3=STPTS(L1,3)
      CALL FUNCT(2,SA)
      DETT(L1)=DET
C
      L3=6*L1-6
      DO 402 K=1,21
      K3=3*K
      K2=K3-1
      K1=K2-1
      STR(L3+1,K1) = SA(K,1)
      STR(L3+2,K2) = SA(K,2)
      STR(L3+3,K3) = SA(K,3)
      STR(L3+4,K1) = SA(K,2)
      STR(L3+4,K2) = SA(K,1)
      STR(L3+5,K2) = SA(K,3)
      STR(L3+5,K3) = SA(K,2)
      STR(L3+6,K1) = SA(K,3)
  402 STR(L3+6,K3) = SA(K,1)
C
  405 L2=L2+3
C
C     STATIC CONDENSATION
C
      DO 710 M=1,15
      MN=64-M
      MO=MN-1
C
C      STIFFNESS MATRIX - S
C
      SP=S(MN,MN)
      DO 650 I=1,MO
  650 S(MN,I)=S(I,MN)/SP
      DO 700 K=1,MO
      SP=S(MN,K)
      DO 700 J=1,K
  700 S(J,K)=S(J,K) - SP*S(J,MN)
C
C      DERIVATIVE MATRIX - STR
C
      DO 710 J=1,42
      SP=STR(J,MN)
      IF(SP.EQ.0.) GO TO 710
      DO 705 K=1,MO
  705 STR(J,K)=STR(J,K) - SP*S(MN,K)
  710 CONTINUE
      DO 760 I=1,48
      DO 760 J=I,48
  760 S(J,I)=S(I,J)
C
C      SAVE ELASTIC PROPERTIES
C
      DETT(8)=CC1
      DETT(9)=CC2
      DETT(10)=CC3
      DETT(11)=FAC
      DO 510 I=1,4
  510 DETT(11+I)=CC4*TLF(I)
   70 CONTINUE
C
C     DISTRIBUTED LOAD
C
      DO 410 J=1,63
      DO 410 I=1,4
  410 RF(J,I)=0.
C
  440 CALL LOCAL(KTYPE,PR,YREF,NFACE)
C
C     SELF WGT.
C
  450 DO 460 II=1,16
      K=3*II
      J=K-1
      I=J-1
      DO 460 L=1,4
```

```
      RF(I,L) = RF(I,L)*PLF(L) + DL(II)*XLF(L)
      RF(J,L) = RF(J,L)*PLF(L) + DL(II)*YLF(L)
  450 RF(K,L) = RF(K,L)*PLF(L) + DL(II)*ZLF(L)
C
C     THERMAL LOADS
C
      DO 470 I=1,48
      GT=TT(I)*FACT
      DO 470 J=1,4
  470 RF(I,J)=RF(I,J) + GT*TLF(J)
C
C     MASS ARRAY
C
      L=0
      DO 465 I=1,16
      DO 465 J=1,3
      L=L+1
  465 XM(L)=DL(I)/GRAV
C
      IJ=0
      DO 550 I=1,16
      II=NP(I)
      DO 550 J=1,3
      IJ=IJ+1
  550 LM(IJ)=ID(II,J)
      NS=42
      ND=48
      NDM=63
C
      CALL CALBAN (MBAND,NDIF,LM,XM,S,RF,ND,NDM)
      WRITE (1) ND,NS,(DETT(I),I=1,15),(LM(I),I=1,ND),((STR(I,J),I=1,NS)
     1,J=1,ND)
C
C     CHECK IF LAST ELEMENT
C
      IF(N3DEL-NEL) 50,600,590
  590 IF(ML) 30,30,40
C
C
  600 RETURN
C
C
 1000 FORMAT (4I5,4I2,2X,F10.2/16I5)
 1001 FORMAT (I5,4F10.0)
 1002 FORMAT (2I5,2F10.2,I5)
 1003 FORMAT (2F10.2/(4F10.2))
 2000 FORMAT (I6,X,8I5/7X,8I5,I9,I12,8X,A1,3X,4I5,F8.1/)
 2001 FORMAT (X,I5,4E15.4)
 2002 FORMAT (I5,I9,2F13.3,I12)
 2003 FORMAT (31H1.....STRESS FREE TEMPERATURE =  F10.3////
     . 38H LOAD FACTORS FOR 4 ELEMENT LOAD CASES  //
     . 46X 17HELEMENT LOAD CASE /
     . 36X 1HA 9X 1HB 9X 1HC 9X 1HD  /
     . 30H PRESSURE LOAD FACTORS . .      4F10.3/
     . 30H THERMAL  LOAD FACTORS . .      4F10.3//
     . 30H PERCENT GRAVITY IN +X DIRN.   4F10.3/
     . 30H PERCENT GRAVITY IN +Y DIRN.   4F10.3/
     . 30H PERCENT GRAVITY IN +Z DIRN.   4F10.3/ )
 1300 FORMAT (9H1MATERIAL 10X 1HE 12X 2HNU 10X 3HRHO 11X 7HALPHA-T /
     . 8H  NUMBER /)
 1301 FORMAT (30H1...16 NODE SOLID ELEMENT DATA ///
     . 8H ELEMENT 10X 15HCONNECTED NODES 17X 28HINTEGRATION  MATERIAL  I
     .NPUT 7X 13HELEMENT LOADS 5X 7HELEMENT /
     . 8H  NUMBER 3X,36H1    2    3    4    5    6    7    8 6X,5HORDER
     . 7X,3HNO. 6X 3HTAG 7X 16H1    2    3    4 4X  5HTEMP. /
     . 11X,36H9   10   11   12   13   14   15   16 //    )
 1302 FORMAT (/////26H ELEMENT DISTRIBUTED LOADS //
     . 52H NUMBER     KTYPE      PR          YREF          FACE )
 4003 FORMAT (36HOELEMENT CARD ERROR, ELEMENT NUMBER= I6)
C
      END
```

```
      SUBROUTINE FUNCT (KK,D)
C
      COMMON /JUNK/ R ,S ,T ,DET,MLD(4),KLD(4),MULT(4),A(3,3),
     1P(3,21),B(3,3),XZ(16,3),Q(19)
      COMMON/GASS/XK(4,4),WGT(4,4),IPERM(3)
      DIMENSION D(21,3),BB(3)
C
      R2=2.*(1.-(R**2))
      S2=2.*(1.-(S**2))
      RP=.125*(1.+R)
      RN=.125*(1.-R)
      SP=1.+S
      SN=1.-S
      TP=1.+T
      TN=1.-T
      RPSP= R+S-1.
      RPSN= R-S-1.
      RNSP=-R+S-1.
      RNSN=-R-S-1.
      XPP= 2.*R+S
      XPN= 2.*R-S
      XNP=-2.*R+S
      XNN=-2.*R-S
      YPP= R+(2.*S)
      YPN= R-(2.*S)
      YNP=-R+(2.*S)
      YNN=-R-(2.*S)
      XX=.125
      IF (KK.EQ.2) GO TO 100
C
C      SHAPE FUNCTIONS
C
      Q( 1)=RP*SP*TP*RPSP
      Q( 2)=RN*SP*TP*RNSP
      Q( 3)=RN*SN*TP*RNSN
      Q( 4)=RP*SN*TP*RPSN
      Q( 5)=RP*SP*TN*RPSP
      Q( 6)=RN*SP*TN*RNSP
      Q( 7)=RN*SN*TN*RNSN
      Q( 8)=RP*SN*TN*RPSN
      Q( 9)=R2*SP*TP*XX
      Q(10)=RN*S2*TP
      Q(11)=R2*SN*TP*XX
      Q(12)=RP*S2*TP
      Q(13)=R2*SP*TN*XX
      Q(14)=RN*S2*TN
      Q(15)=R2*SN*TN*XX
      Q(16)=RP*S2*TN
C
C      DERIVATIVES OF SHAPE FUNCTIONS
C
  100 XR=-.5*R
      XS=-4.*S
      P(1,1)= XX*SP*TP*XPP
      P(1,2)=-XX*SP*TP*XNP
      P(1,3)=-XX*SN*TP*XNN
      P(1,4)= XX*SN*TP*XPN
      P(1,5)= XX*SP*TN*XPP
      P(1,6)=-XX*SP*TN*XNP
      P(1,7)=-XX*SN*TN*XNN
      P(1,8)= XX*SN*TN*XPN
      P(1,9)= XR*SP*TP
      P(1,10)=-XX*S2*TP
      P(1,11)= XR*SN*TP
      P(1,12)= XX*S2*TP
      P(1,13)= XR*SP*TN
      P(1,14)=-XX*S2*TN
      P(1,15)= XR*SN*TN
      P(1,16)= XX*S2*TN
      P(1,17)=1.-(3.*R*R)
      P(1,18)=0.
      P(1,19)=0.
      P(1,20)=S*(1.0-(3.0*R*R))
      P(1,21)=S*(1.0-(S*S))
C
```

```
      P(2,1)= RP*TP*YPP
      P(2,2)= RN*TP*YNP
      P(2,3)=-RN*TP*YNN
      P(2,4)=-RP*TP*YPN
      P(2,5)= RP*TN*YPP
      P(2,6)= RN*TN*YNP
      P(2,7)=-RN*TN*YNN
      P(2,8)=-RP*TN*YPN
      P(2,9)= R2*TP*XX
      P(2,10)= XS*TP*RN
      P(2,11)=-R2*TP*XX
      P(2,12)= XS*TP*RP
      P(2,13)= R2*TN*XX
      P(2,14)= XS*TN*RN
      P(2,15)=-R2*TN*XX
      P(2,16)= XS*TN*RP
      P(2,17)=0.
      P(2,18)=1.-(3.*S*S)
      P(2,19)=0.
      P(2,20)=R*(1.0-(R*R))
      P(2,21)=R*(1.0-(3.0*S*S))
C
      P(3,1)= RP*SP*RPSP
      P(3,2)= RN*SP*RNSP
      P(3,3)= RN*SN*RNSN
      P(3,4)= RP*SN*RPSN
      P(3,5)=-RP*SP*RPSP
      P(3,6)=-RN*SP*RNSP
      P(3,7)=-RN*SN*RNSN
      P(3,8)=-RP*SN*RPSN
      P(3,9)= R2*SP*XX
      P(3,10)= RN*S2
      P(3,11)= R2*SN*XX
      P(3,12)= RP*S2
      P(3,13)=-R2*SP*XX
      P(3,14)=-RN*S2
      P(3,15)=-R2*SN*XX
      P(3,16)=-RP*S2
      P(3,17)=0.
      P(3,18)=0.
      P(3,19)=-2.*T
      P(3,20)=0.0
      P(3,21)=0.0
C
C       JACOBIAN MATRIX A
C
      DO 200 I=1,3
      DO 200 J=1,3
      C=0.
      DO 150 K=1,16
  150 C=C+P(I,K)*XZ(K,J)
  200 A(I,J)=C
      IF (KK.EQ.3) GO TO 600
C
C       INVERT JACOBIAN
C
      DO 300 I=1,3
      J=IPERM(I)
      K=IPERM(J)
      B(I,I)=A(J,J)*A(K,K)-A(K,J)*A(J,K)
      B(I,J)=A(K,J)*A(I,K)-A(I,J)*A(K,K)
  300 B(J,I)=A(J,K)*A(K,I)-A(J,I)*A(K,K)
      DET=A(1,1)*B(1,1)+A(1,2)*B(2,1)+A(1,3)*B(3,1)
C
C       MATRIX OF X-Y-Z DERIVATIVES
C
      DO 400 I=1,3
      DO 400 J=1,21
      C=0.
      DO 350 K=1,3
  350 C=C+B(I,K)*P(K,J)
  400 D(J,I)=C
C
  600 RETURN
C
      END


      SUBROUTINE LOCAL(KTYPEE,PRR,YREFF,NFACE)
C
      COMMON/EM/LM(48),ND,NDM,RF(63,4),XM(48),SA(21,3),DL(16)
      COMMON/JUNK/ETA(3),DET,MLD(4),KLD(4),MULT(4),A(3,3),
     1P(3,21),B(3,3),XX(16,3),Q(19)
      COMMON/GASS/DUM(12),XK(4),DDUM(12),WGT(4),IPERM(3)
C
      DIMENSION KTYPEE(1),PRR(1),YREFF(1),NFACE(1),KCRD(6),FVAL(6),
     1KFACE(6,8)
C
      DATA KFACE /  1, 2, 2, 3, 2, 6,
     1              5, 6, 6, 7,10,14,
     2             16,14,13,15, 3, 7,
     3              8, 7, 5, 8,11,15,
     4              4, 3, 1, 4, 4, 8,
     5             12,10, 9,11,12,16,
     6              0, 0, 0, 0, 1, 5,
     7              0, 0, 0, 0, 9,13/
      DATA KCRD / 1,1,2,2,3,3/
      DATA FVAL /1.,-1.,1.,-1.,1.,-1./
C
C
      DO 700 KK=1,4
      NNN=KLD(KK)
      IF(NNN) 700,700,10
   10 KTYPE=KTYPEE(NNN)
      PR=PRR(NNN)
      YREF=YREFF(NNN)
      KF=NFACE(NNN)
C
C       INTEGRATE OVER THE SURFACE
C
      ML = KCRD(KF)
      MM = IPERM(ML)
      MN = IPERM(MM)
      ETA(ML) = FVAL(KF)
      DO 300  LX = 1,4
      ETA(MM) = XK(LX)
      DO 300  LY = 1,4
      ETA(MN) = XK(LY)
      CALL FUNCT(3,SA)
C
C       COMPUTE DIRECTION COSINES OF NORMAL TO SURFACE
C
      A1 = (A(MM,2)*A(MN,3)-A(MM,3)*A(MN,2))
      A2 = (A(MM,3)*A(MN,1)-A(MM,1)*A(MN,3))
      A3 = (A(MM,1)*A(MN,2)-A(MM,2)*A(MN,1))
      AA=SQRT(A1**2+A2**2+A3**2)
      A1 = A1/AA
      A2 = A2/AA
      A3 = A3/AA
C
C       COMPUTE FIRST FUND. FORM   (SIN / )
C
      AA = 0.
      BB = 0.
      CC = 0.
      DO 200  I = 1,3
      AA=AA+A(MM,I)**2
      CC=CC+A(MN,I)**2
  200 BB = BB + A(MM,I)*A(MN,I)
      C=SQRT (AA*CC - BB*BB)
C
C       COMPUTE PRESSURE,LOAD COMPONENTS, STORE IN R
C
      IF (KTYPE.EQ.2) GO TO 170
      FORCE = PR
```

```
      GO TO 185
  170 YY = 0.
      DO 180  I = 1,16
  180 YY = YY + Q(I)*XX(I,2)
      YY = YY - YREF
      FORCE = -PR*YY
      IF(YY.GT.0.) FORCE = 0.
  185 CONTINUE
      TS=FORCE*WGT(LX)*WGT(LY)*C
C
      DO 190  I = 1,8
      N = KFACE(KF,I)
      IF (N.EQ.0) GO TO 190
      QQ=TS*Q(N)
      K=3*N
      RF(K-2,KK) = RF(K-2,KK) + QQ*A1
      RF(K-1,KK) = RF(K-1,KK) + QQ*A2
      RF(K  ,KK) = RF(K  ,KK) + QQ*A3
  190 CONTINUE
C
  300 CONTINUE
C
  700 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE ST3D16(D,STR,NEQ,NUME)
C
      COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,MEQ
      COMMON/EM/ND,NS,LM(48),ST(42,48),DETT(15)
      COMMON/JUNK/LT,LH,MM,L,K,SIG(6),STN(6),E(3,3),C1,C2,C3,I,J,IL,J1,
     1J2,J3,I1,I2,I3,SS,DET,KX,KY,KZ,DD,SX,FAC
C
      DIMENSION D(NEQ,1),STR(4,1)
C
      WRITE (6,2005)
      WRITE (6,3025)
      DO 800 MM=1,NUME
      READ  (1) ND,NS,(DETT(I),I=1,15),(LM(I),I=1,ND),((ST(I,J),I=1,NS),
     1J=1,ND)
      C1=DETT(8)
      C2=DETT(9)
      C3=DETT(10)
      FAC=DETT(11)
      DO 501 I=1,3
  501 E(I,I)=C1
      E(1,2) = C2
      E(1,3) = C2
      E(2,3) = C2
      E(2,1) = C2
      E(3,1) = C2
      E(3,2) = C2
      LL=0
      DO 700 L=LT,LH
      WRITE (6,2000)
      LL=LL+1
C
      DO 540 IL=1,7
      DO 503 I=1,3
      J=I+3
      SS=0.
      DO 502 KK=1,4
  502 SS=SS-DETT(11+KK)*STR(KK,L)
      STN(I)=SS
  503 STN(J)=0.
      DET=DETT(IL)
C
      J1=6*(IL-1)
      DO 510 I=1,48
      I1=LM(I)
      IF(I1) 510,510,505
  505 DD=D(I1,LL)/DET
      DO 508 J=1,6
  508 STN(J)=STN(J) + ST(J1+J,I)*DD
  510 CONTINUE
C
      SX=C3*FAC
      DO 515 I=1,3
      SS=0.
      DO 514 J=1,3
  514 SS = SS + E(I,J)*STN(J)*FAC
      SIG(I)=SS
      J=I+3
  515 SIG(J)=STN(J)*SX
C
      I=IL-1
      WRITE (6,3005) MM,L,I,SIG
C
  540 CONTINUE
  590 CONTINUE
  700 CONTINUE
  800 CONTINUE
C
      RETURN
C
 2000 FORMAT (/)
 2005 FORMAT (36H1....16-NODE SOLID ELEMENT STRESSES   //
     . 31X,67H SIG-XX       SIG-YY       SIG-ZZ       SIG-XY       SIG-YZ
     .    SIG-ZX  )
 3005 FORMAT (/I6,I9,I8,5X,1P6E12.3)
 3025 FORMAT (//24H ELEMENT  LOAD NO.  FACE )
C
      END
```

APPENDIX E - ANALYSIS OF MINE STRUCTURE

A three-dimensional analysis of a typical section of a mine structure was conducted in order to illustrate the degree of difficulty of such an investigation. A quarter of a typical room and pillar mine was identified by a system of finite elements as shown in Figure E.1. The model is composed of 282 nodal points, 36 eight node elements and 32 sixteen node elements. This resulted in 613 equilibrium equations with a maximum bandwidth of 199. The total computer time required on the CDC 6400 was 218 seconds. This represents less than $50 cost at commerical rates. The preparation of data for this example involved approximately 2 man-days. Selective results are plotted in Figures E.2, E.3 and E.4.

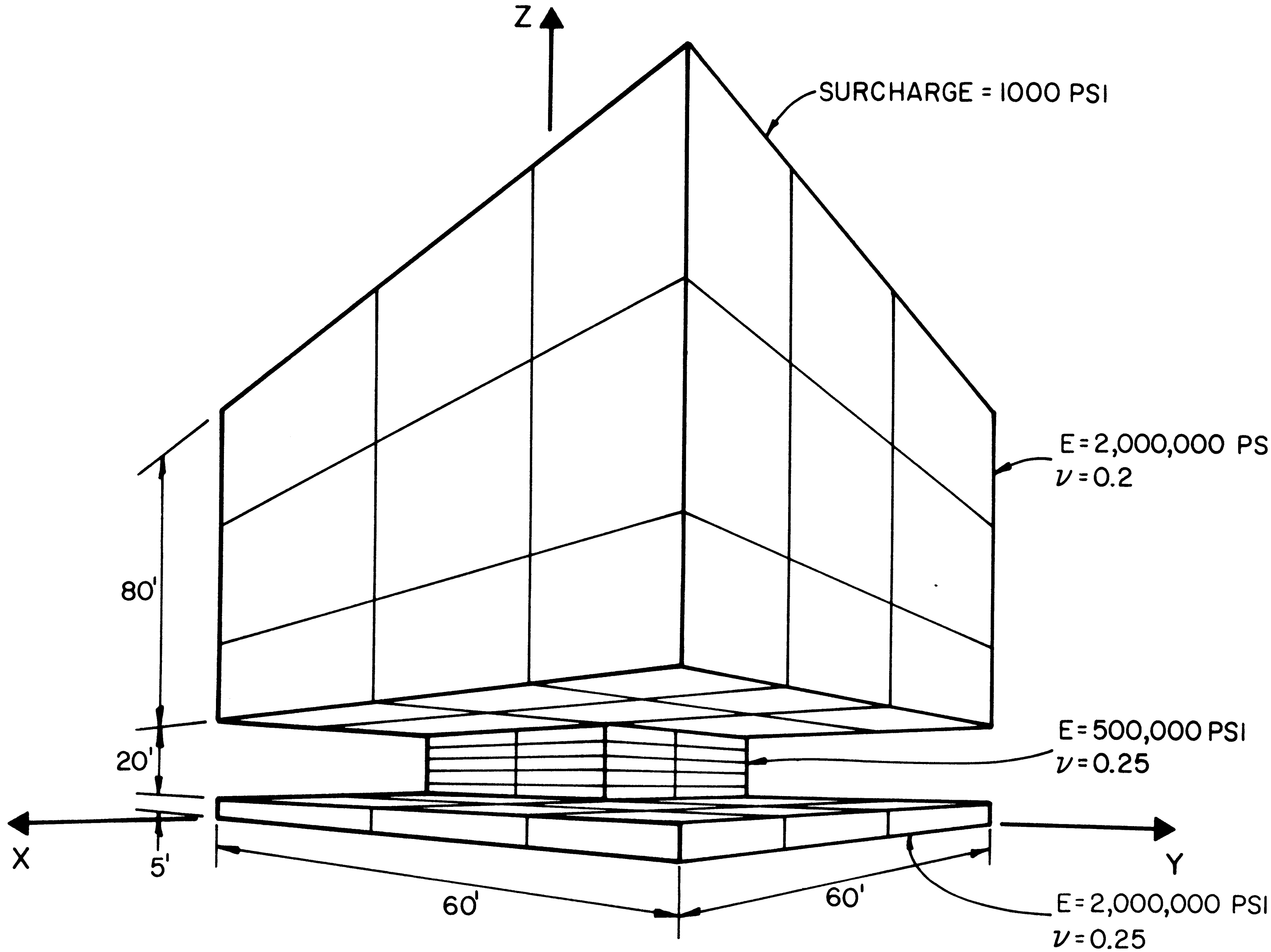


FIGURE E.1 TYPICAL MINE STRUCTURE

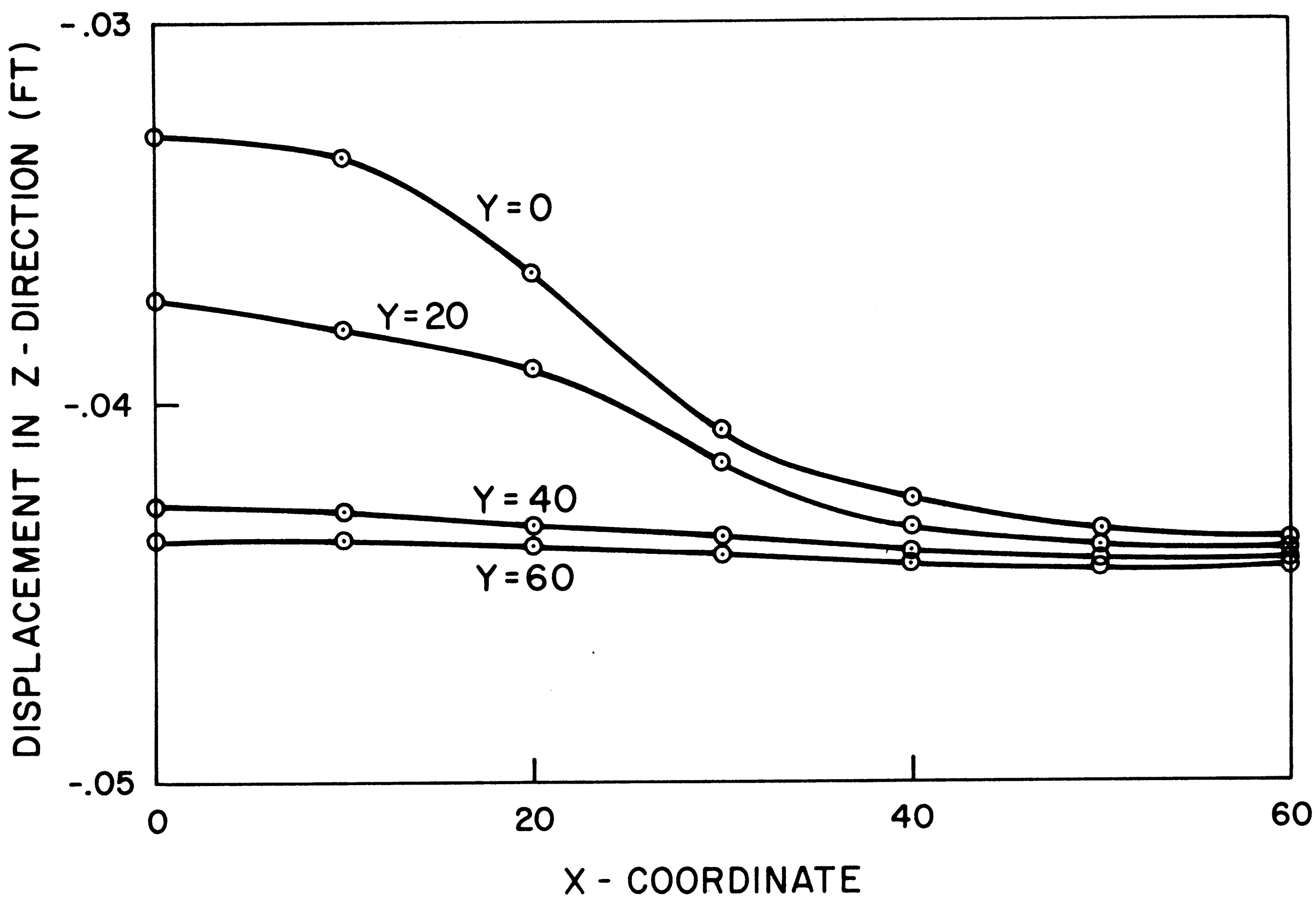


FIGURE E.2 VERTICAL DISPLACEMENTS OF THE CEILING OF THE MINE ROOM (Z=25 FT)

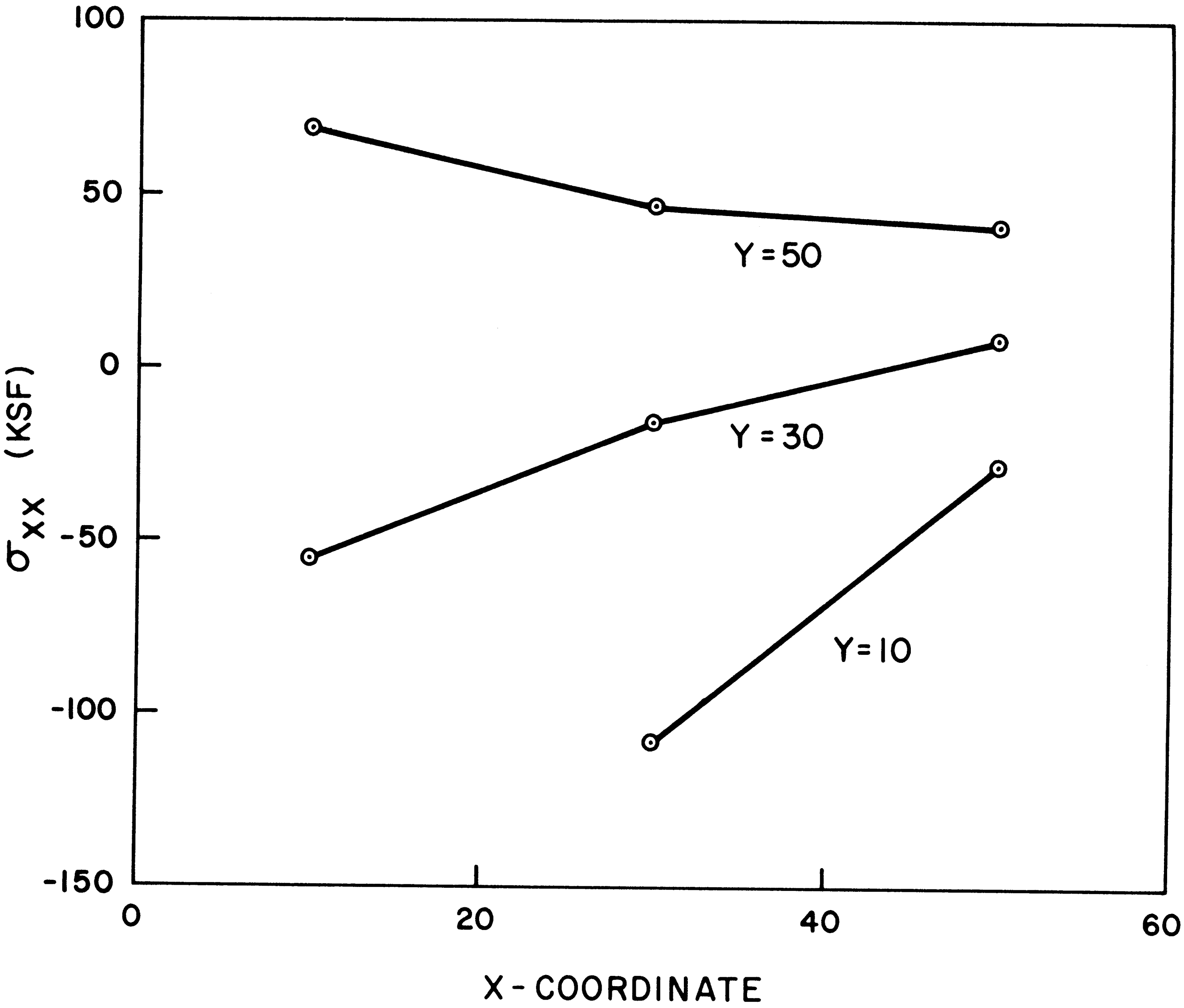


FIGURE E.3 STRESSES IN THE CEILING OF THE MINE ROOM (Z=25 FT)

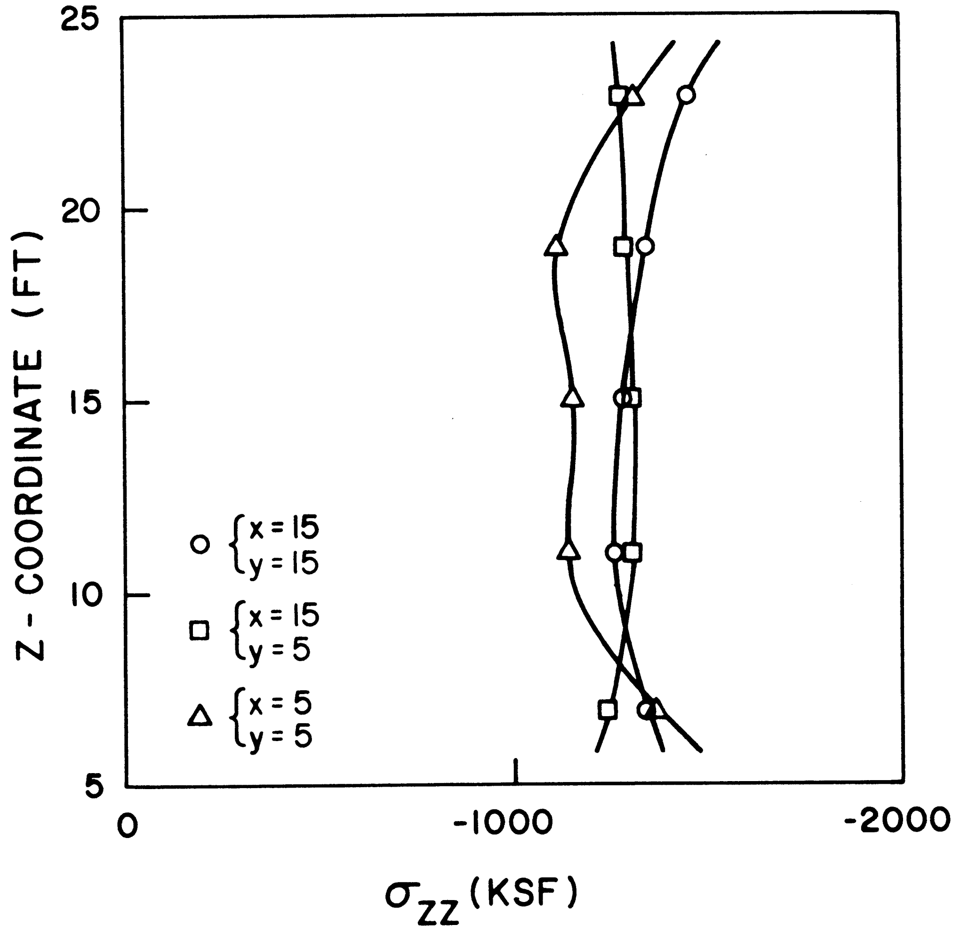


FIGURE E.4 STRESSES IN THE MINE PILAR