

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

Bureau of Mines Open File Report 15-72

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.

September 1971
edition never issued

ABSTRACT

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

	<u>Page</u>
ABSTRACT	i
ACKNOWLEDGEMENTS	ii
I. INTRODUCTION	1
II. EQUILIBRIUM EQUATIONS FOR STRUCTURAL SYSTEMS	3
2.1 The Direct Stiffness Method	3
2.2 Boundary Conditions	5
III. THREE DIMENSIONAL TRUSS ELEMENT	8
IV. THREE DIMENSIONAL BEAM ELEMENT	12
4.1 Definition of Principal Axes	12
4.2 Master and Slave Degrees of Freedom	14
V. SOLID FINITE ELEMENTS	17
5.1 Introduction	17
5.2 Source of Errors	18
5.3 Addition of Incompatible Modes For Two-Dimensional Isoparametric Elements	21
VI. PLATE AND THIN SHELL ELEMENT	44
VII. BOUNDARY ELEMENT	46
VIII. COMPUTER PROGRAM ORGANIZATION	47
6.1 Solution of Equations	48
6.2 Formation of Equilibrium Equations	48
6.3 Joint Input Data and Degrees of Freedom	49
6.4 Calculation of Element Stiffness Matrices	49
6.5 Evaluation of Element Stresses	50

	<u>Page</u>
REFERENCES	51
APPENDICES	
A THE STATIC CONDENSATION ALGORITHM	A-1
B DESCRIPTION OF INPUT DATA FOR SOLID-SAP	B-1
C PROGRAM CAPACITY.	C-1
D FORTRAN IV COMPUTER PROGRAM LISTING	D-1
E ANALYSIS OF TYPICAL MINE STRUCTURE	E-1

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:

$$\underline{K} \underline{u} = \underline{R} \quad (2.1)$$

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

$$\underline{K} = \Sigma \underline{K}_m \quad (2.2)$$

For a typical element m the element stiffness matrix is given by

$$\underline{K}_m = \int_{Vol} \underline{a}_m^T \underline{c}_m \underline{a}_m dV_m \quad (2.3)$$

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

$$\underline{\sigma}_m = \underline{c}_m \underline{\epsilon}_m + \underline{\tau}_m \quad (2.4)$$

where $\underline{\epsilon}_m$ 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:

$$\underline{\varepsilon}_m = \underline{a}_m \underline{u} \quad (2.5)$$

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

$$\underline{R} = \underline{P} + \underline{I} - \underline{F} \quad (2.6)$$

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

$$\underline{I} = \sum \underline{I}_m \quad (2.7)$$

in which

$$\underline{I}_m = \int_{\text{Area}} \underline{b}_m^T \underline{t}_m \, ds_m \quad (2.8)$$

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

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

\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

$$\underline{F} = \sum \underline{F}_m \quad (2.10)$$

in which

$$\underline{F}_m = \int_{Vol} \underline{d}_m^T \underline{\tau}_m dV_m \quad (2.11)$$

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

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

2.2 Boundary Conditions

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

$$\underline{K}_{aa} \underline{u}_a + \underline{K}_{ab} \underline{u}_b = \underline{R}_a \quad (2.13)$$

$$\underline{K}_{ba} \underline{u}_a + \underline{K}_{bb} \underline{u}_b = \underline{R}_b \quad (2.14)$$

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:

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

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:

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

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

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

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.

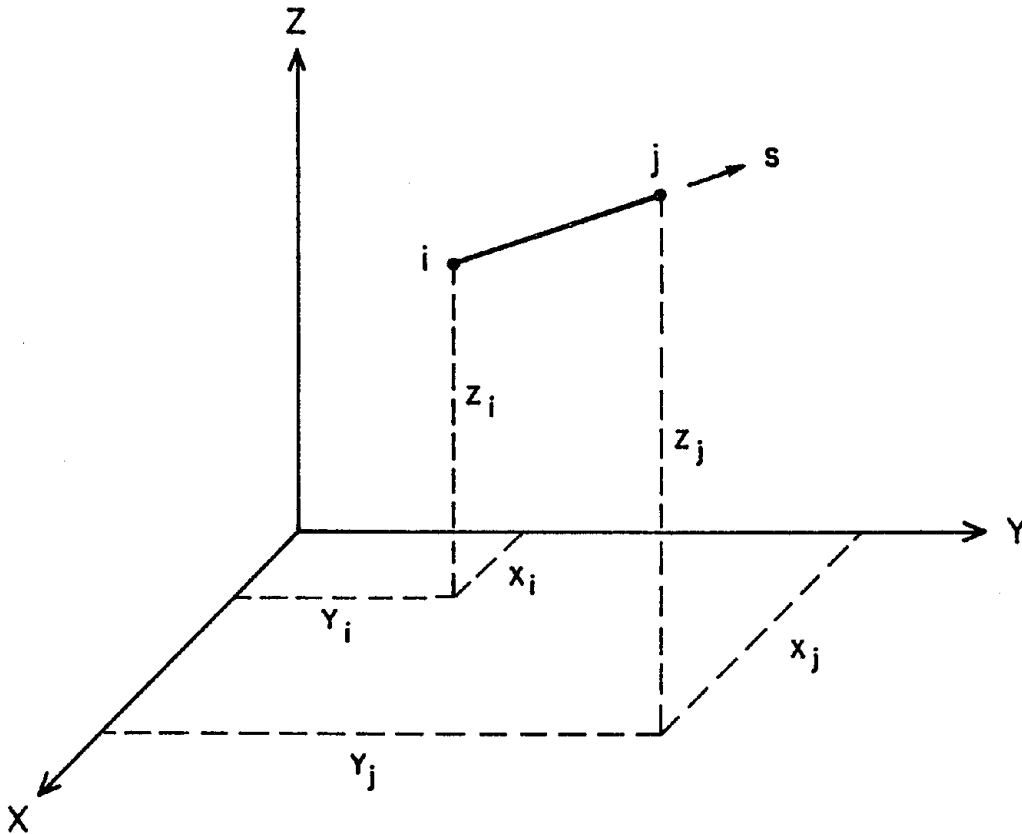


FIGURE 3-1 TYPICAL TRUSS ELEMENT

The length of the element is given by

$$L = \sqrt{L_x^2 + L_y^2 + L_z^2} \quad (3.1)$$

where

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

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

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

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

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

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

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

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:

$$\epsilon_s = \frac{1}{L^2} \begin{bmatrix} -L_x & -L_y & -L_z & L_x & L_y & L_z \end{bmatrix} \begin{bmatrix} u_{xi} \\ u_{yi} \\ u_{zi} \\ u_{xj} \\ u_{yj} \\ u_{zj} \end{bmatrix} \quad (3.6)$$

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

$$\underline{a} = \frac{1}{L^2} \begin{bmatrix} -L_x & -L_y & -L_z & L_x & L_y & L_z \end{bmatrix} \quad (3.7)$$

The axial stress is expressed in terms of axial strain by

$$\sigma_s = E \epsilon_s \quad (3.8)$$

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

$$\underline{c} = [E] \quad (3.9)$$

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:

$$K = \frac{AE}{L^3} \begin{bmatrix} -L_x \\ -L_y \\ -L_z \\ L_x \\ L_y \\ L_z \end{bmatrix} \begin{bmatrix} -L_x & -L_y & -L_z & L_x & L_y & L_z \end{bmatrix} \quad (3.10)$$

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

$$\sigma_s = \frac{E}{L^2} \begin{bmatrix} -L_x & -L_y & -L_z & L_x & L_y & L_z \end{bmatrix} \begin{bmatrix} u_{xi} \\ u_{yi} \\ u_{zi} \\ u_{xj} \\ u_{yj} \\ u_{zj} \end{bmatrix} \quad (3.11)$$

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

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

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

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

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

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

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

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.

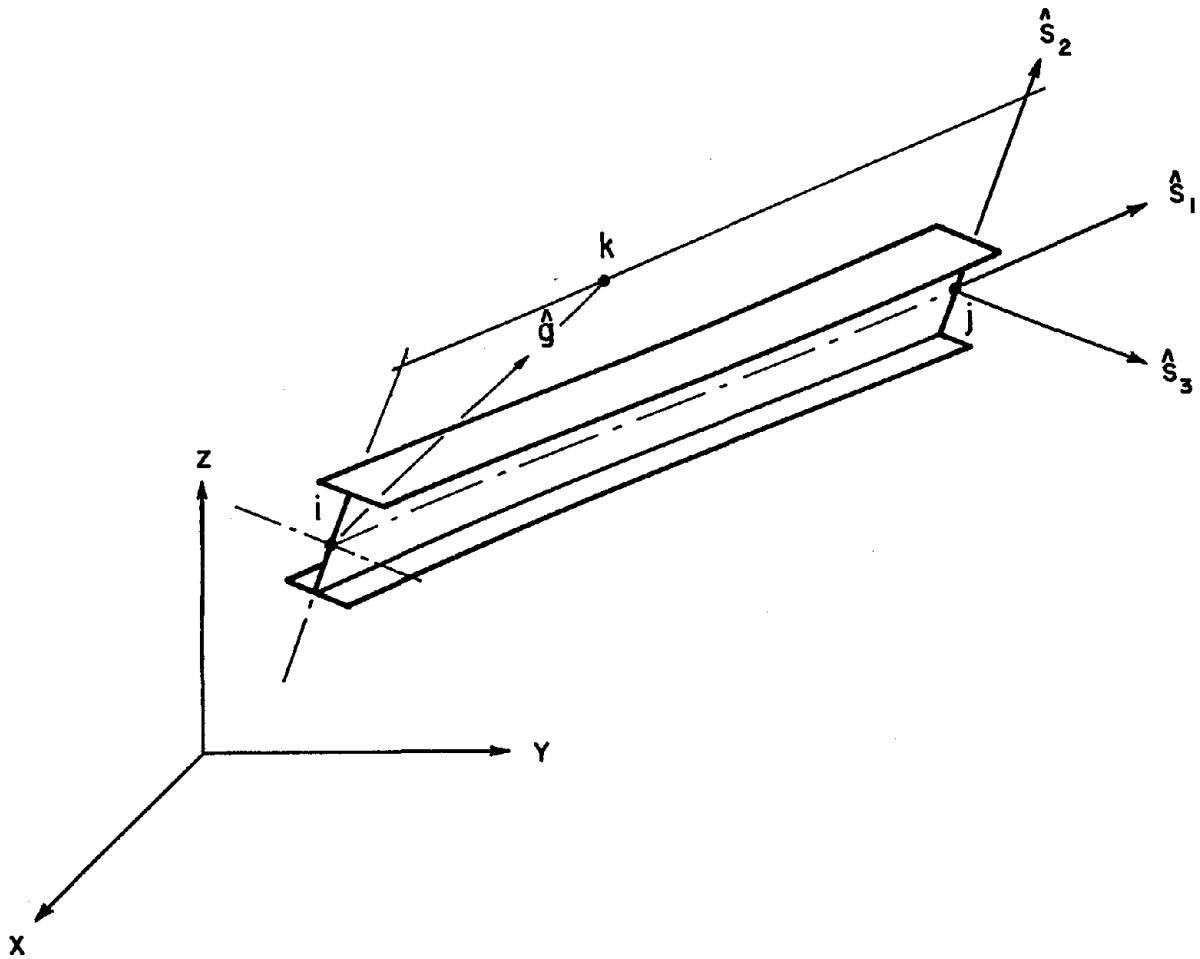


FIGURE 4-1 THREE DIMENSIONAL BEAM ELEMENT

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.

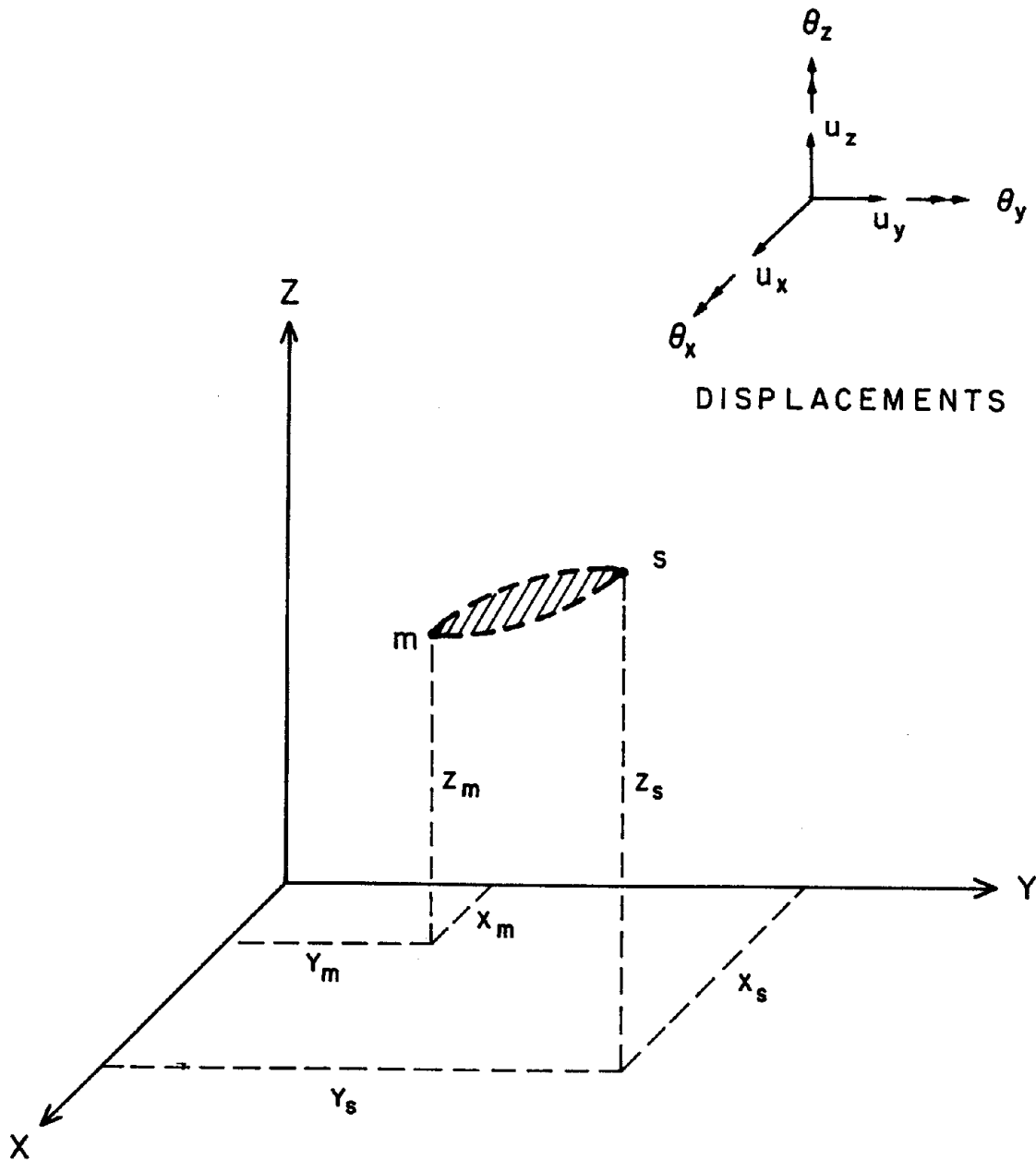


FIGURE 4-2 MASTER AND SLAVE JOINTS

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:

$$u_x = \alpha_1 xy \quad (5.1)$$

$$u_y = 1/2 \alpha_1 (a^2 - x^2) + \alpha_2 (b^2 - y^2) \quad (5.2)$$

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

$$\epsilon_{xy} = \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} = 0 \quad (5.3)$$

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

$$u_x = \beta_1 xy \quad (5.4)$$

Therefore, the form of the error in the solution is

$$u_y = \beta_2 (a^2 - x^2) + \beta_3 (b^2 - y^2) \quad (5.5)$$

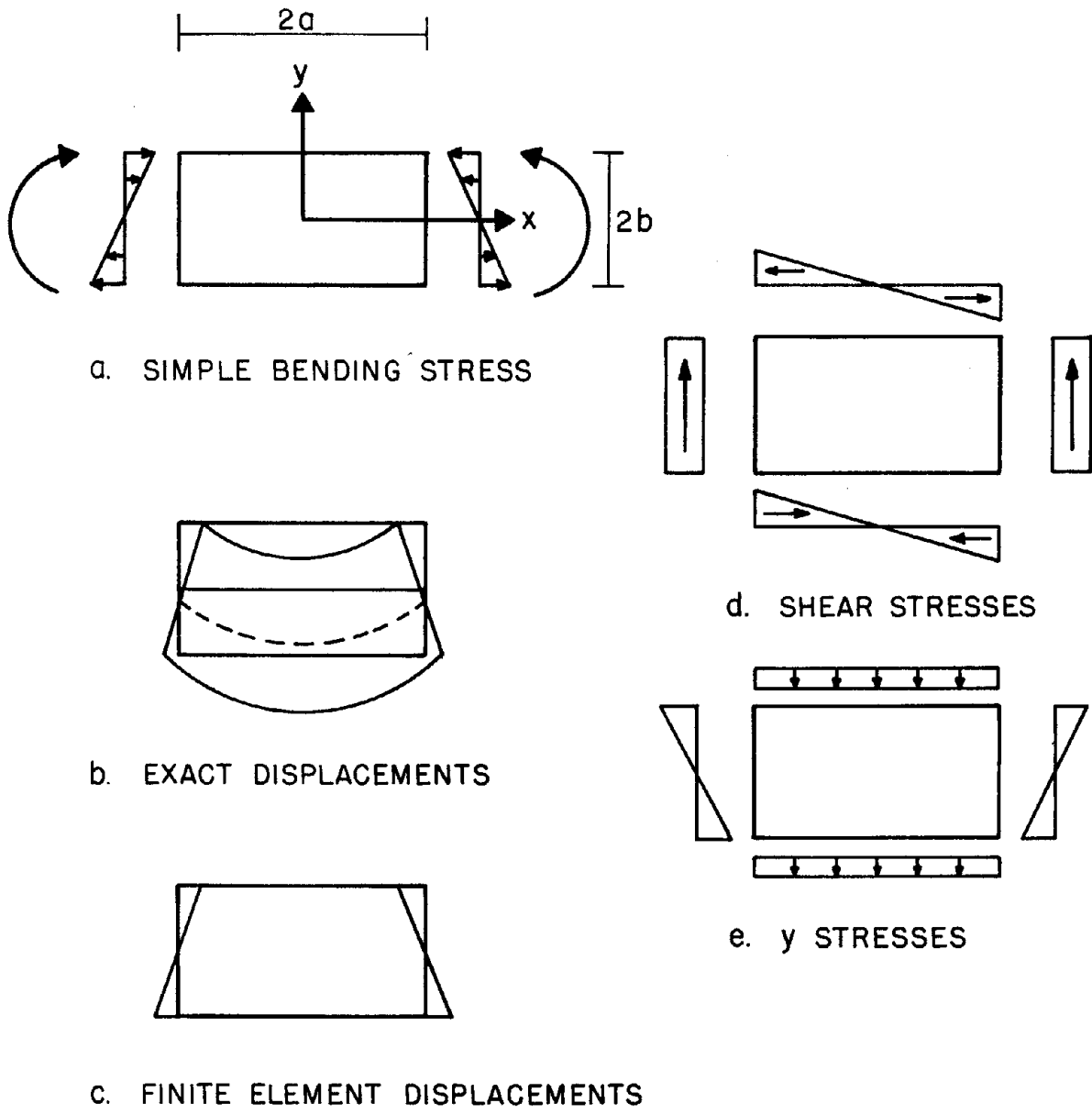


FIGURE 5-1 ERRORS DUE TO PURE BENDING STRESSES

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

$$x = \sum_{i=1}^4 h_i x_i \quad (5.6a)$$

$$y = \sum_{i=1}^4 h_i y_i \quad (5.6b)$$

where the interpolation functions are given by

$$h_1 = 1/4 (1-s) (1-t) \quad (5.7a)$$

$$h_2 = 1/4 (1+s) (1-t) \quad (5.7b)$$

$$h_3 = 1/4 (1+s) (1+t) \quad (5.7c)$$

$$h_4 = 1/4 (1-s) (1+t) \quad (5.7d)$$

Strain-Displacement Equations

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

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

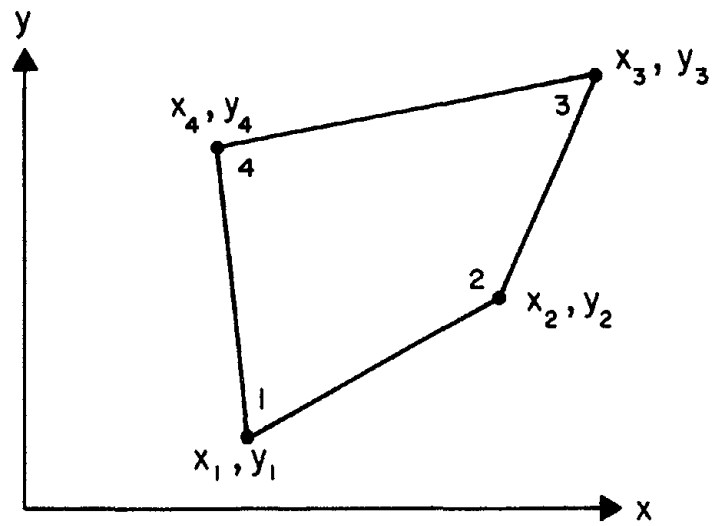
$$u_y (s,t) = \sum h_i u_{yi} \quad (5.8b)$$

For two dimensional analysis the strain-displacement equations are

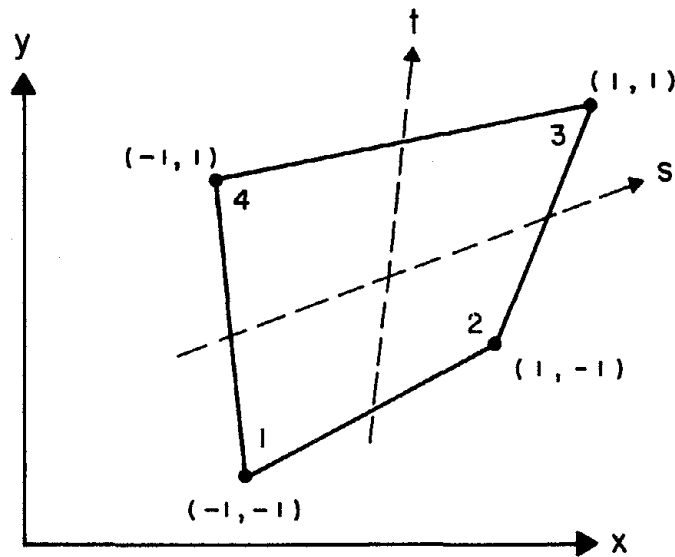
$$\epsilon_{xx} = \frac{\partial u_x}{\partial x} = \sum h_{i,x} u_{xi} \quad (5.9a)$$

$$\epsilon_{yy} = \frac{\partial u_y}{\partial y} = \sum h_{i,y} u_{yi} \quad (5.9b)$$

$$\epsilon_{xy} = \frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} = \sum h_{i,y} u_{xi} + \sum h_{i,x} u_{yi} \quad (5.9c)$$



a. GLOBAL SYSTEM



b. LOCAL SYSTEM

FIGURE 5-2 TWO-DIMENSIONAL ISOPARAMETRIC ELEMENT

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

$$\underline{\epsilon} = \underline{a}(s,t) \underline{U} = \begin{bmatrix} \underline{H}_{,x} & 0 \\ 0 & \underline{H}_{,y} \\ \underline{H}_{,y} & \underline{H}_{,x} \end{bmatrix} \begin{bmatrix} \underline{U}_x \\ \underline{U}_y \end{bmatrix} \quad (5.10)$$

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

$$\underline{H}_{,x} = [h_{1,x} \ h_{2,x} \ h_{3,x} \ h_{4,x}] \quad (5.11a)$$

$$\underline{H}_{,y} = [h_{1,y} \ h_{2,y} \ h_{3,y} \ h_{4,y}] \quad (5.11b)$$

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.

$$h_{i,x} = h_{i,s} s_{,x} + h_{i,t} t_{,x} \quad (5.12a)$$

$$h_{i,y} = h_{i,s} s_{,y} + h_{i,t} t_{,y} \quad (5.12b)$$

In general, the chain rule can be written as

$$\begin{bmatrix} \frac{\partial}{\partial s} \\ \frac{\partial}{\partial t} \end{bmatrix} = \begin{bmatrix} x_{,s} & y_{,s} \\ x_{,t} & y_{,t} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} \quad (5.13)$$

or inverted as

$$\begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} = \begin{bmatrix} s_{,x} & t_{,x} \\ s_{,y} & t_{,y} \end{bmatrix} \begin{bmatrix} \frac{\partial}{\partial s} \\ \frac{\partial}{\partial t} \end{bmatrix} \quad (5.14)$$

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

$$\begin{bmatrix} s,x & t,x \\ s,y & t,y \end{bmatrix} = \frac{1}{J} \begin{bmatrix} y,t & -y,s \\ -x,t & x,s \end{bmatrix} \quad (5.15)$$

where the Jacobian J is

$$J = x,s y,t - x,t y,s \quad (5.16)$$

From equations (6a) and (6b)

$$\begin{aligned} x,s &= \sum h_{i,s} x_i \\ x,t &= \sum h_{i,t} x_i \\ y,s &= \sum h_{i,s} y_i \\ y,t &= \sum h_{i,t} y_i \end{aligned} \quad (5.17)$$

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

$$\underline{K} = \int_{\text{Area}} \underline{a}^T \underline{c} \underline{a} \, dA \quad (5.18)$$

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

$$\underline{K} = \int_{-1}^1 \int_{-1}^1 \underline{a}^T \underline{c} \underline{a} J \, ds \, dt \quad (5.19)$$

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

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

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:

$$\begin{aligned} u_x &= \sum h_i u_{xi} + h_5 \alpha_1 + h_6 \alpha_2 \\ u_y &= \sum h_i u_{yi} + h_5 \alpha_3 + h_6 \alpha_4 \end{aligned}$$

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 12×12 . 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 8×8 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

$$\begin{aligned} h_5 &= (1-s^2) \\ h_6 &= (1-t^2) \end{aligned}$$

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.

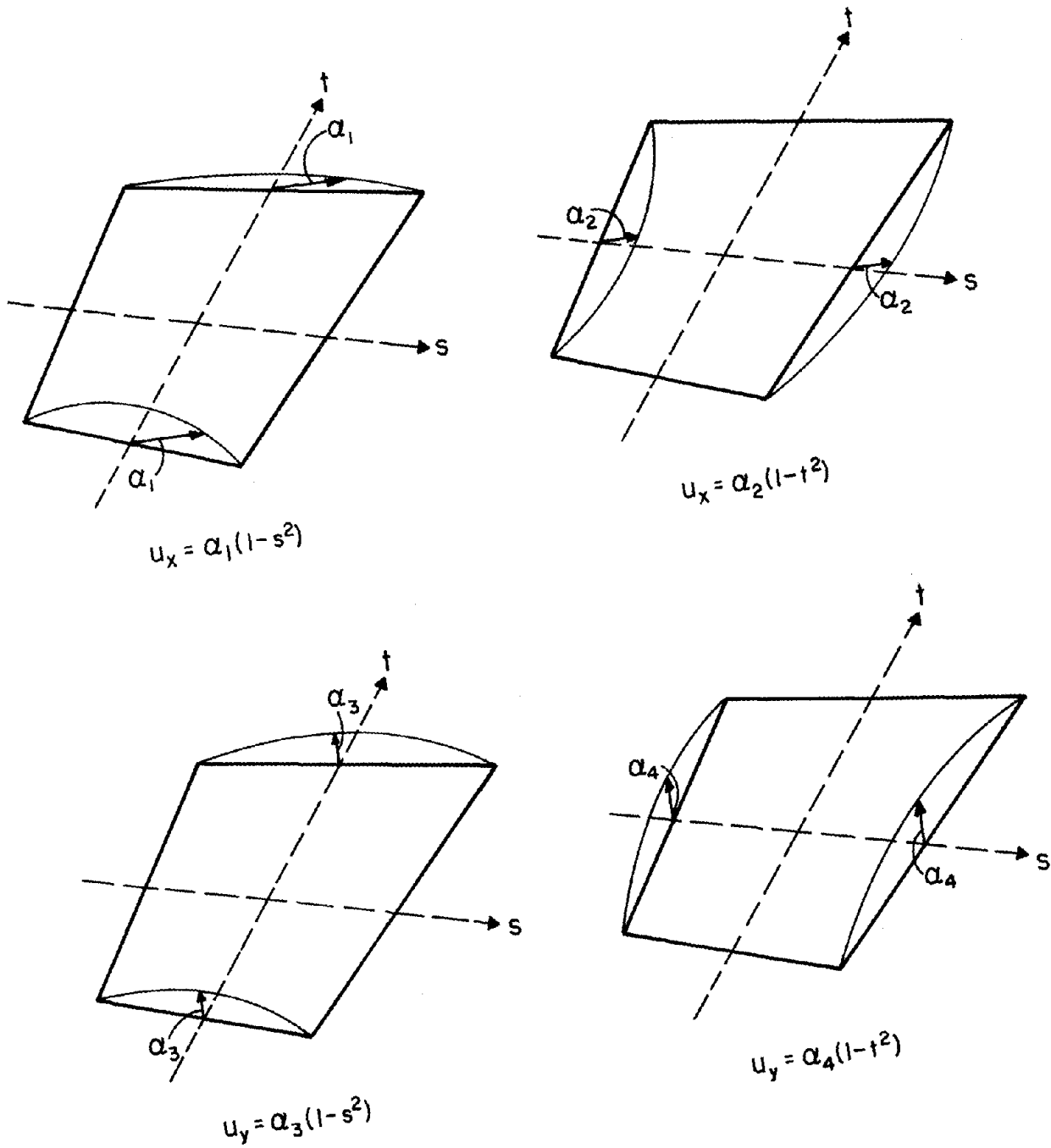


FIGURE 5-3 INCOMPATIBLE DISPLACEMENT MODES FOR GENERAL QUADRILATERAL ELEMENT

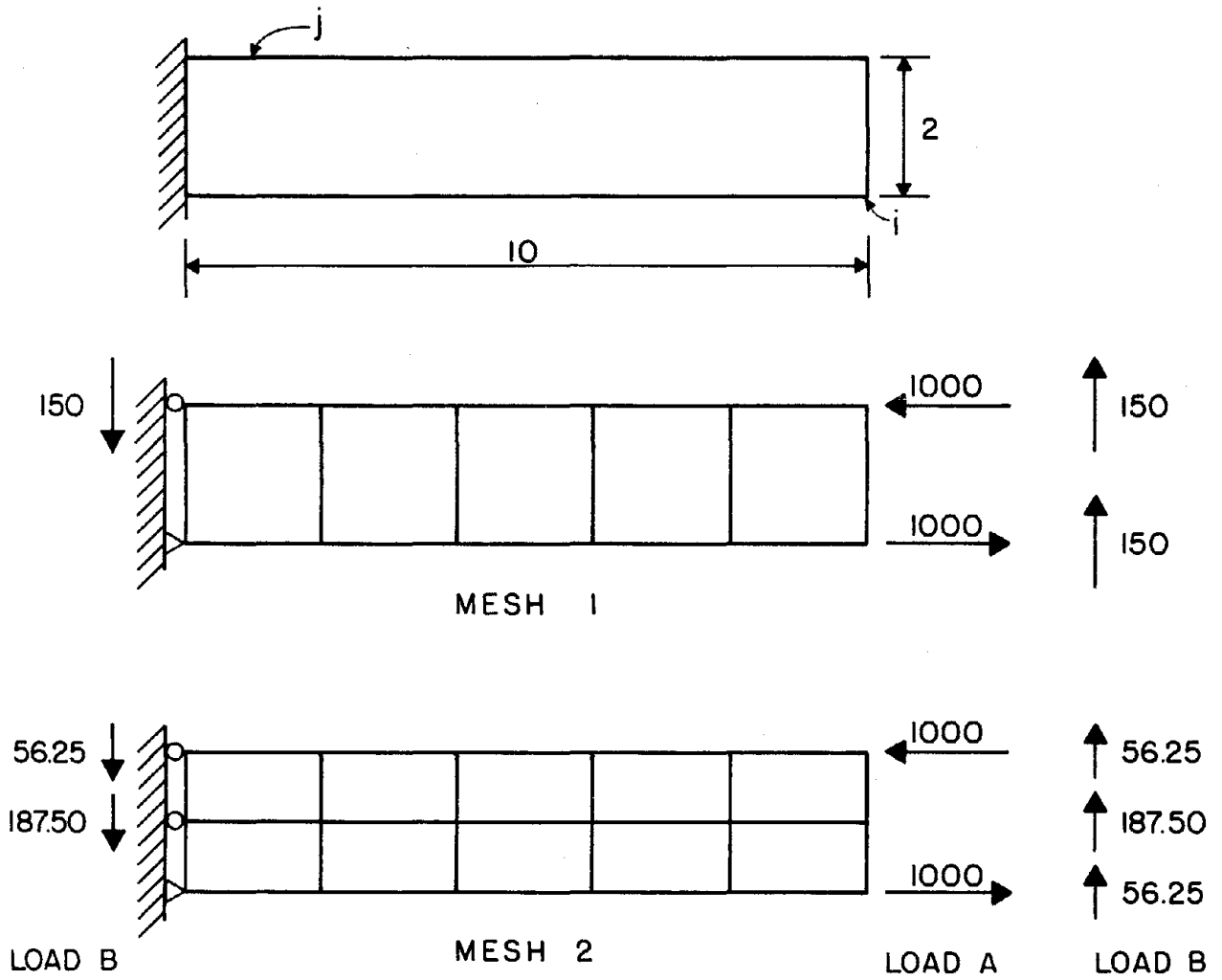


FIGURE 5-4 CANTILEVER BEAM - PLANE STRESS

	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

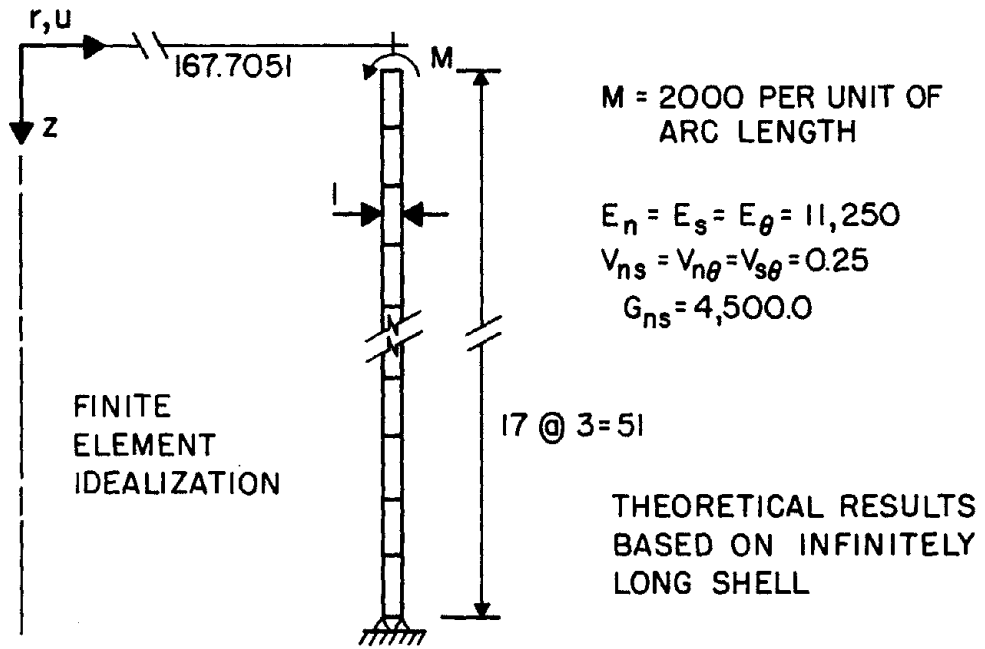


FIGURE 5-5 CYLINDRICAL SHELL ANALYSIS

LATERAL DISPLACEMENTS u					
Z	THEORY	04CST	0M5	04	06
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	04CST	0M5	04	06
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

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

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

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

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

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

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

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

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

$$h_9 = (1 - \xi^2)$$

$$h_{10} = (1 - \eta^2)$$

$$h_{11} = (1 - \zeta^2)$$

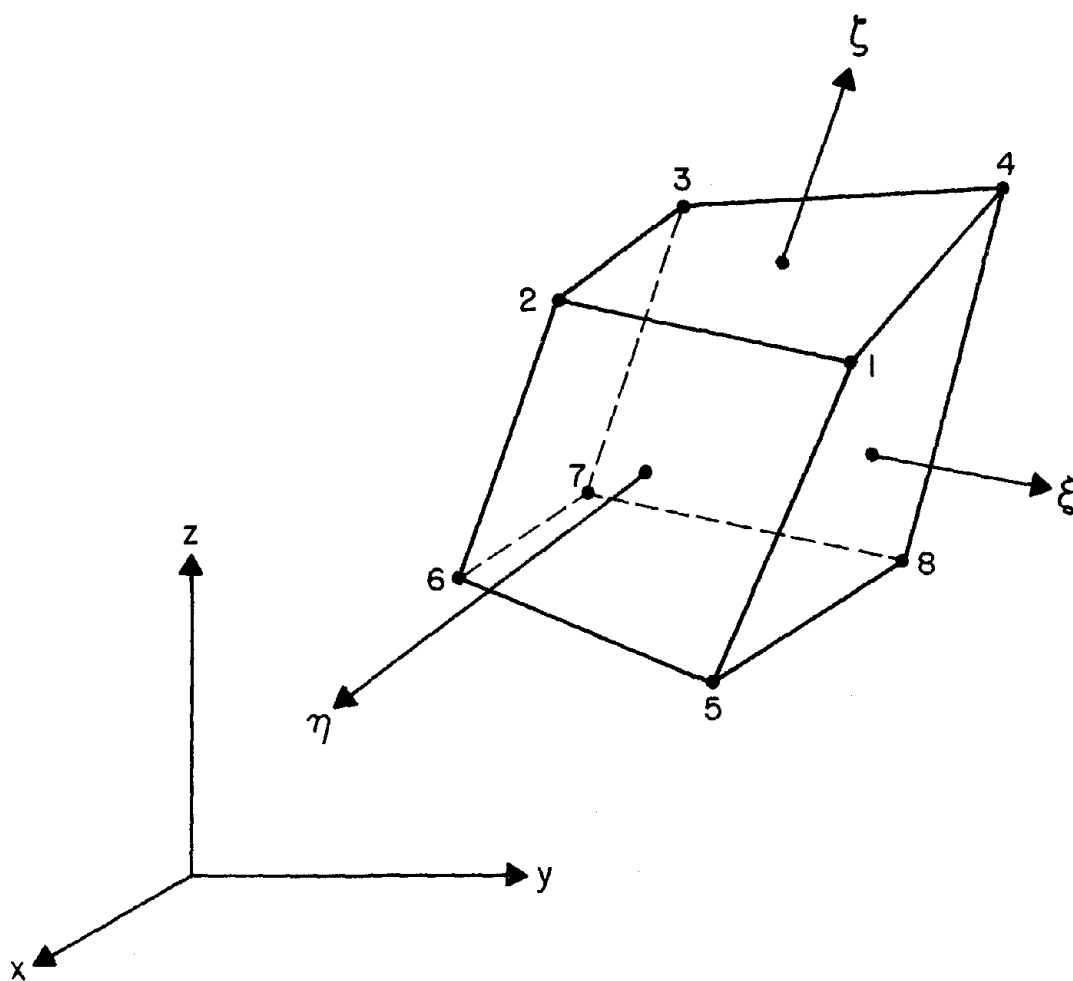


FIGURE 5-6 EIGHT POINT THREE DIMENSIONAL ELEMENT

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

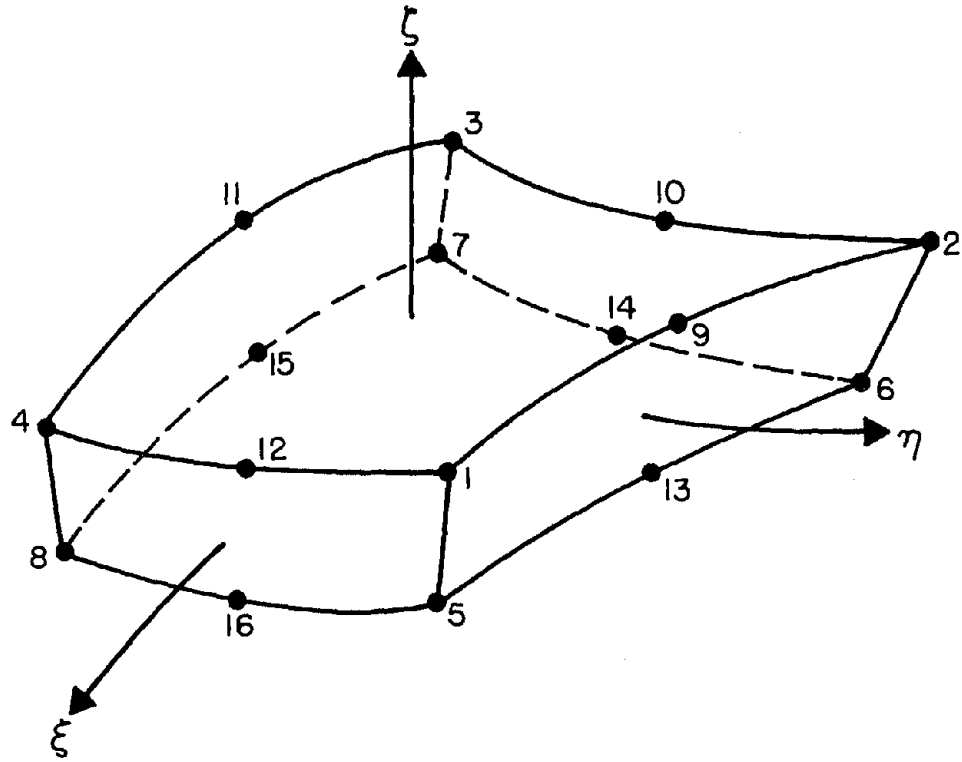


FIGURE 5-7 THREE DIMENSIONAL THICK-SHELL ELEMENT

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

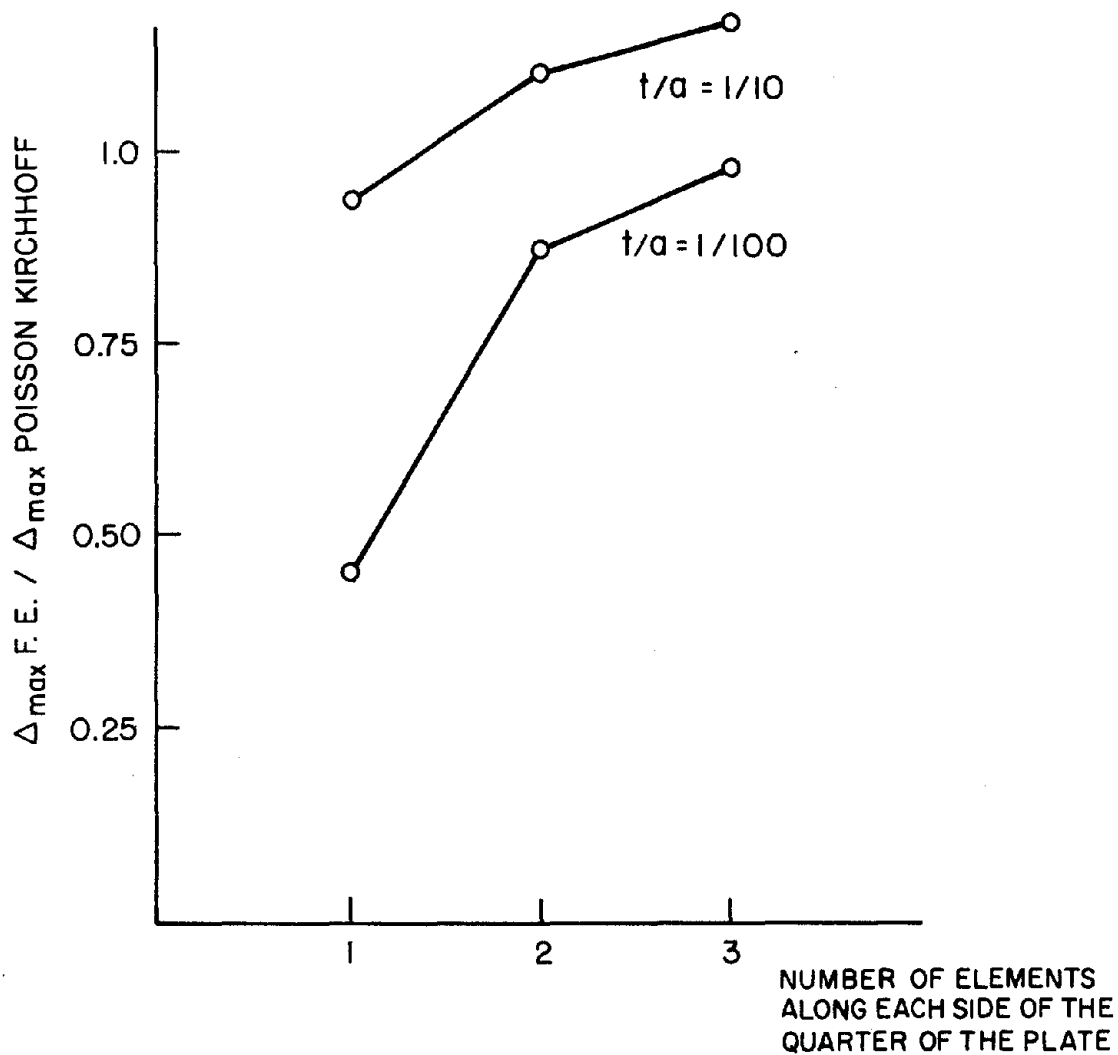
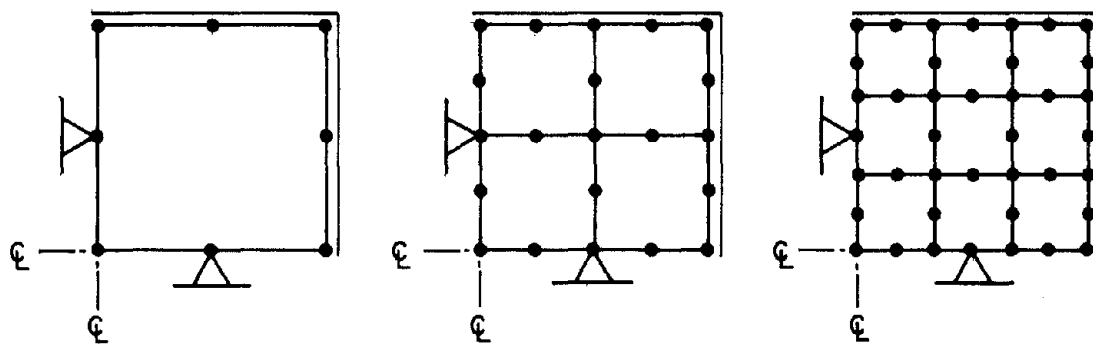


FIGURE 5-8 SIMPLY SUPPORTED SQUARE PLATE WITH A CONCENTRATED LOAD AT THE CENTER

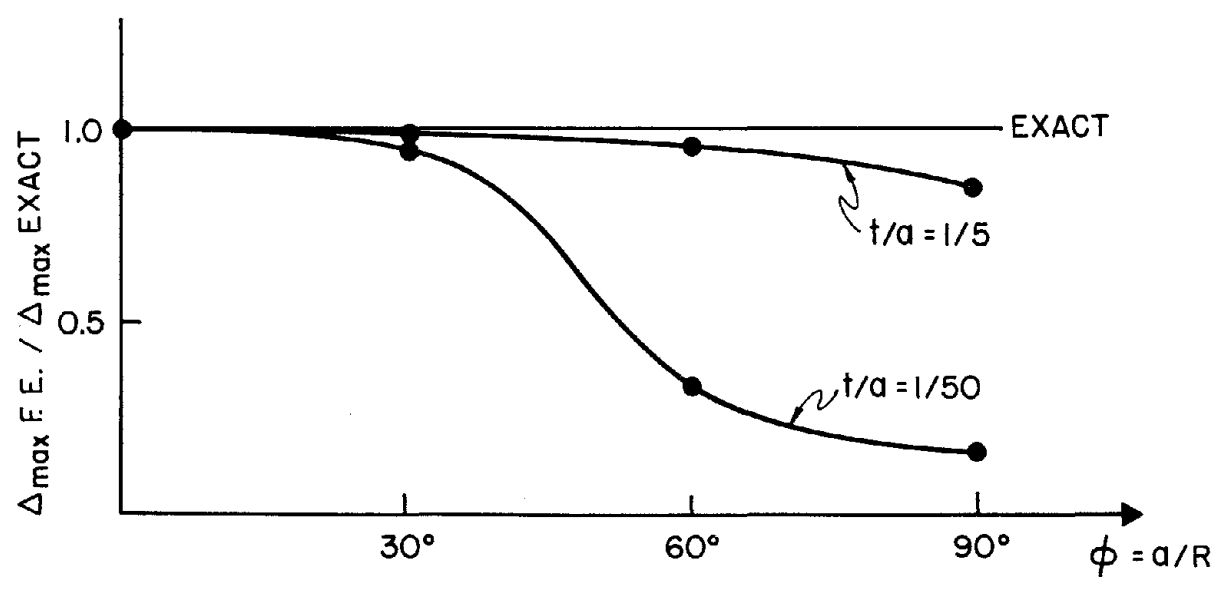
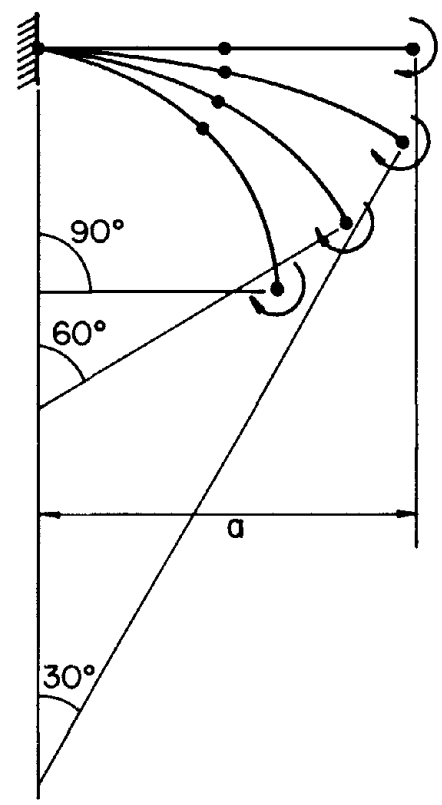


FIGURE 5-9 CURVED CANTILEVER BEAM

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 corner 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.

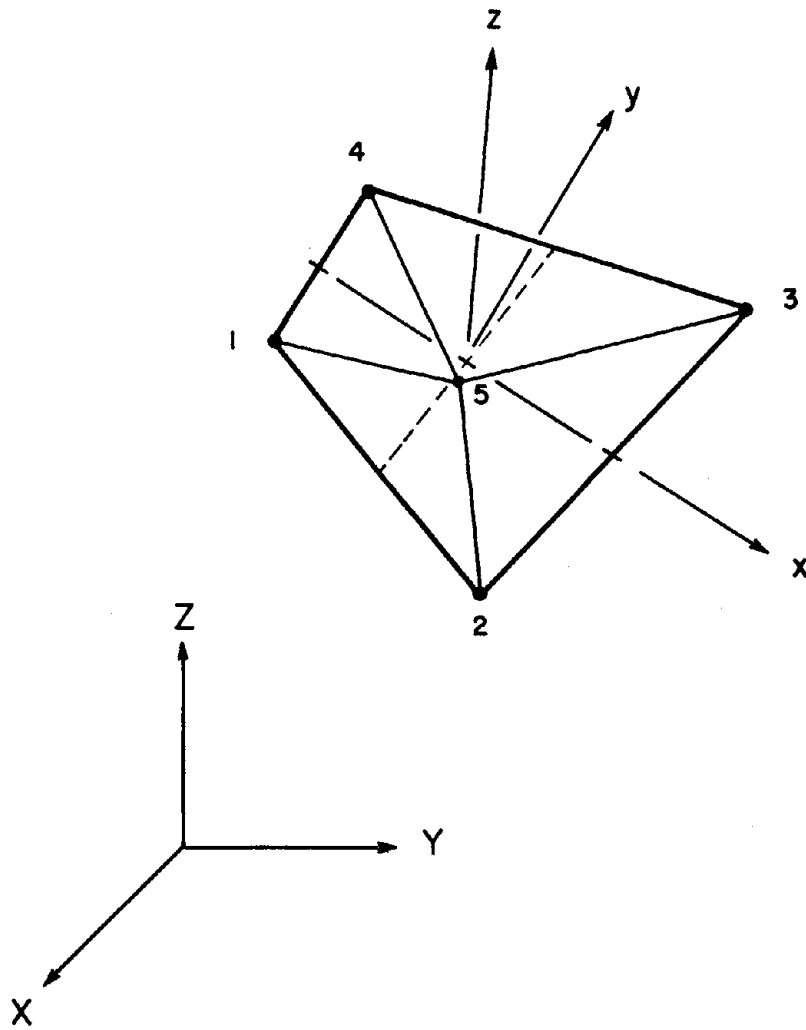


FIGURE 6-1 THIN SHELL ELEMENT

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 than 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 partitioned forms as

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

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

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

in which

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

$$\underline{I} = K_{aa}^{-1} K_{ab} \quad (\text{A.4})$$

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

$$\underline{K}^* \underline{u}_b = \underline{P}^* \quad (\text{A.5})$$

where

$$\underline{K}^* = \underline{K}_{-bb} - \underline{K}_{-ba} \underline{\Gamma} \quad (\text{A.6})$$

$$\underline{P}^* = \underline{P}_{-b} - \underline{K}_{-ba} \underline{C} \quad (\text{A.7})$$

Physically, the term $\underline{K}_{-ba} \underline{\Gamma}$ 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

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

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

$$\underline{\sigma} = \underline{A}^* \underline{u}_a + \underline{\tau}^* \quad (\text{A.9})$$

where

$$\underline{A}^* = \underline{A}_b - \underline{A}_a \underline{\Gamma} \quad (\text{A.10})$$

$$\underline{\tau}^* = \underline{\tau} + \underline{A}_a \underline{C} \quad (\text{A.11})$$

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

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

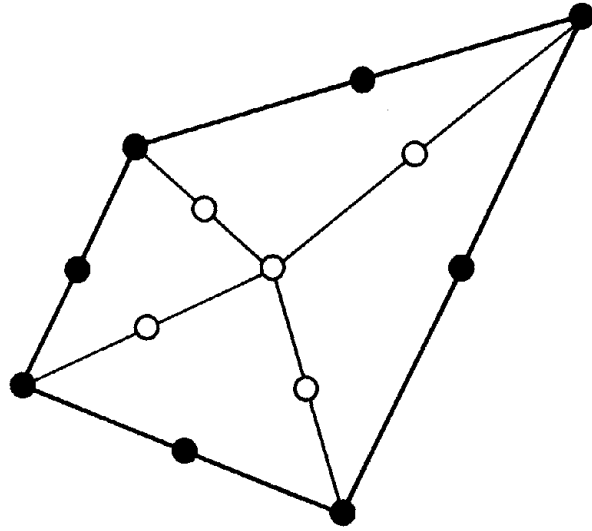


FIGURE A-1 QUADRILATERAL ELEMENT

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

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

3. Substitute into the stress-displacement equations

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

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.

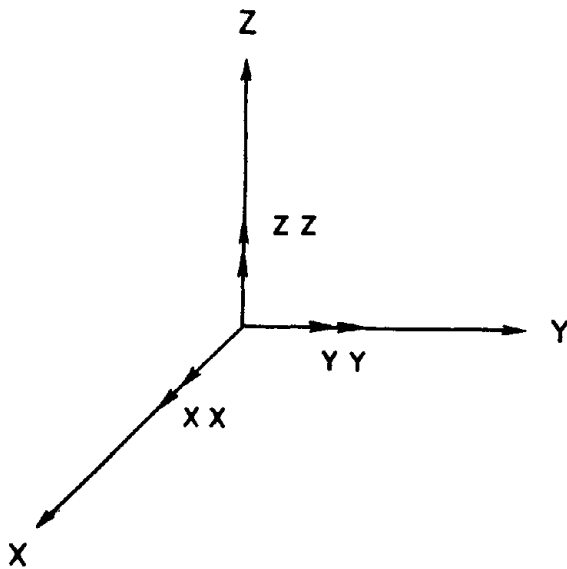
Systems composed of large numbers of joints and members may be analysed. The capacity of the program depends mainly on the total number of joints in the system. There is practically no restriction on the number of elements, number of load cases, or the "bandwidth" of the equations to be solved. Note, 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

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

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.

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.

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

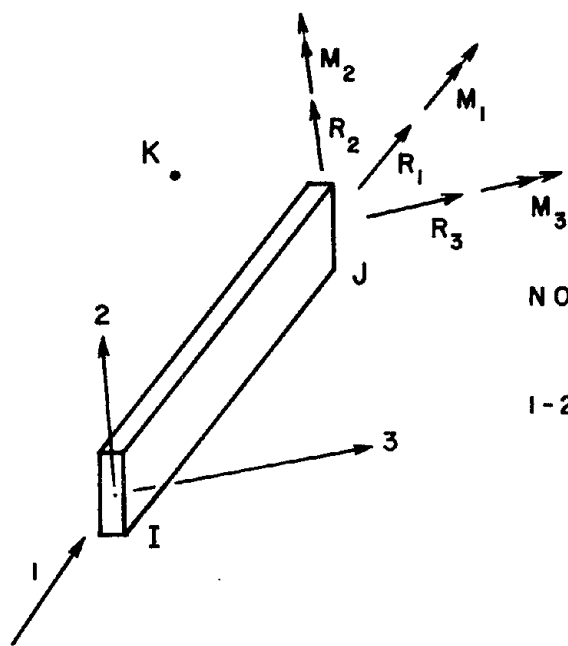
B. Material Property Cards (I5,3F10.0)

Columns	1 - 5	Material identification number
	6 - 15	Young's modulus
	16 - 25	Poisson's ratio

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.



NOTE:
K IS ANY NODAL POINT
WHICH LIES IN THE LOCAL
1-2 PLANE (NOT ON THE 1-AXIS)

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.

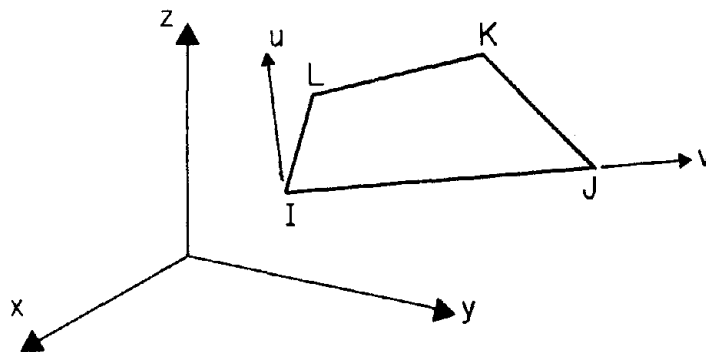
The generator option is the value of k and if left blank is taken to be one. The element data card for the last element in the structure must always be given.

*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 (615)

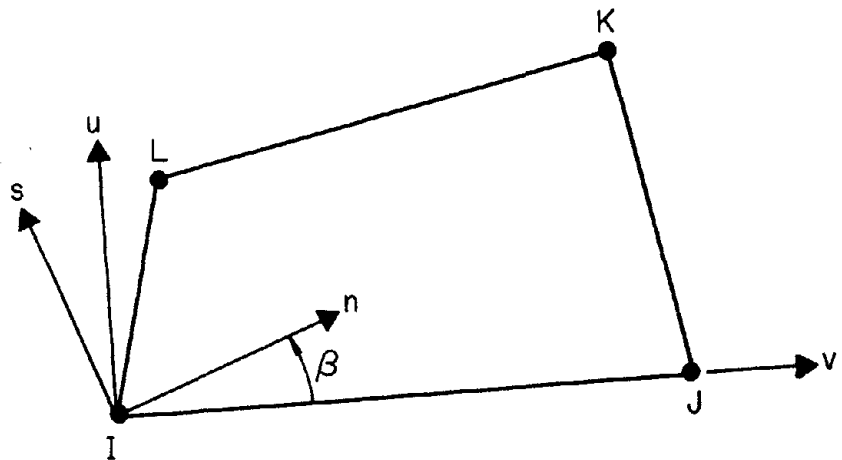
Columns	1 - 5	Number 3
	6 - 10	Number of elements
	11 - 15	Number of different materials
	16 - 20	Maximum number of temperature cards for any one material see Section B below.
	30	Non-zero numerical punch will suppress the introduction of incompatible displacement modes.

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 (215,3F10.0)

- | | | |
|---------|---------|--|
| 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 - E_n
 21 - 30 Modulus of Elasticity - E_s
 31 - 40 Blank
 41 - 50 Strain Ratio - ν_{ns}
 51 - 60 Blank
 61 - 70 Blank
 71 - 80 Shear Modulus - G_{ns}

Card 2: (3F10.0)

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

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 (615,2F10.0,215,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

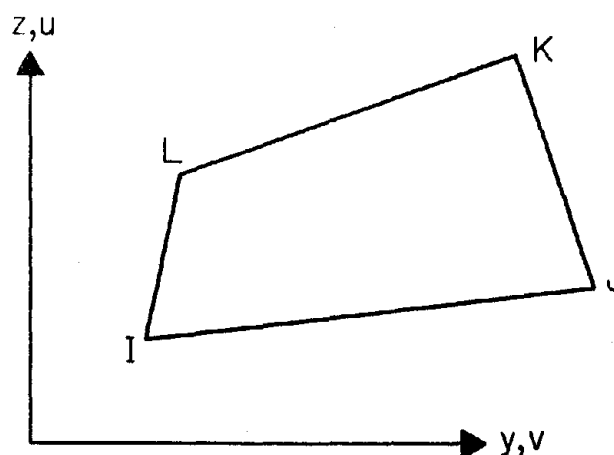
Type 4 - Two Dimensional Finite Elements

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.

A general quadrilateral element is shown below:



A. Control Card (6I5)

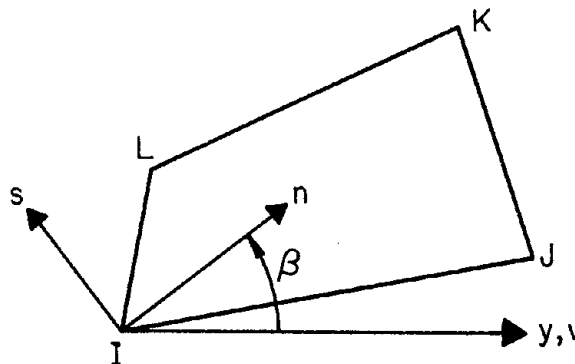
Columns	1 - 5	Number 4
	6 - 10	Number of elements
	11 - 15	Number of different materials
	16 - 20	Maximum number of temperature cards for any one material - see Section B below.
		0 axisymmetric analysis
	25	1 plane strain analysis
		2 plane stress analysis
	30	Non-zero numerical punch will suppress 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	Blank
	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 - E_n
	21 - 30	Modulus of elasticity - E_s
	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 - G_{ns}

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 values of E_t , ν_{nt} , ν_{st} and α_t are automatically set to zero by the program.

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 withing element
	41 - 50	Normal pressure on I-J side of element
	51 - 55	Stress evaluation option "n"
	56 - 70	Element thickness (For plain strain set equal to 1.0 by program)

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:

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

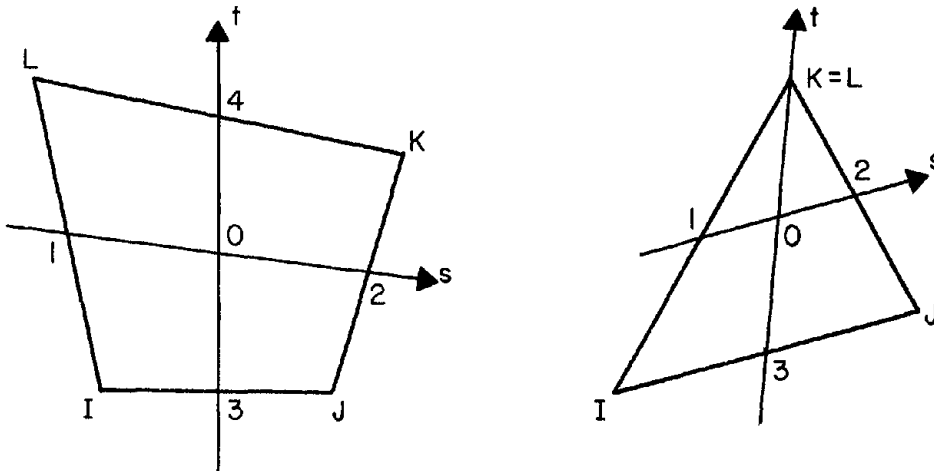
$$J_n = J_{n-1} + k$$

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

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

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.

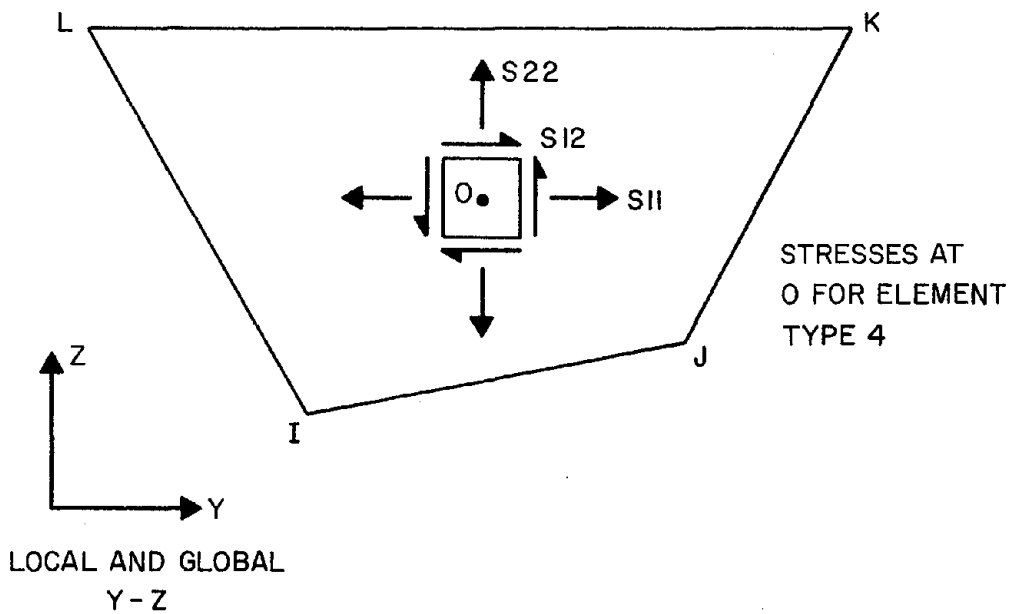
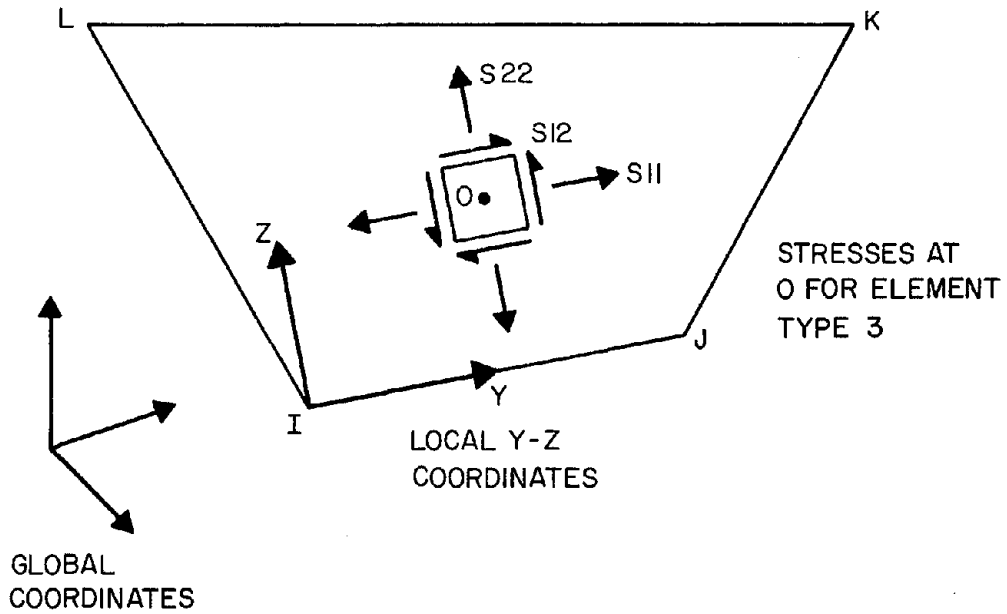
Stress Print Option - The following description of the stress print option applies to both element types 3 and 4. The value of the stress print option "n" can be given as 1, 0, 8, 16 or 20.



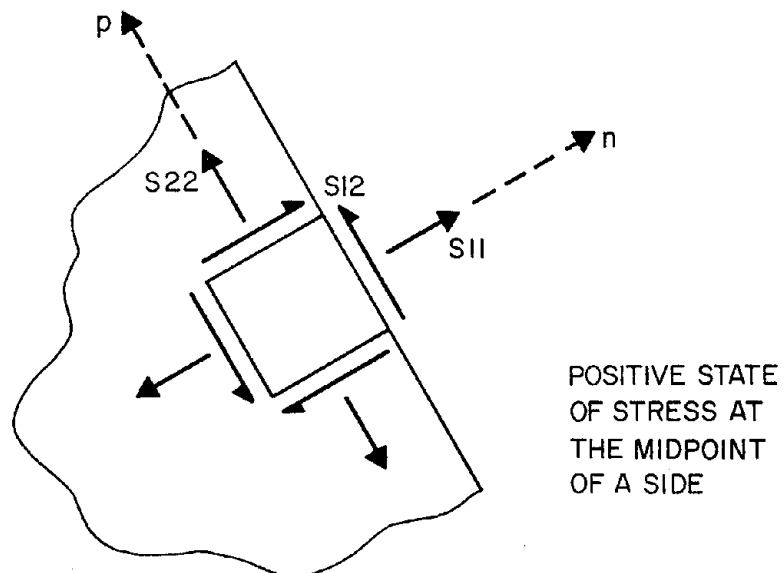
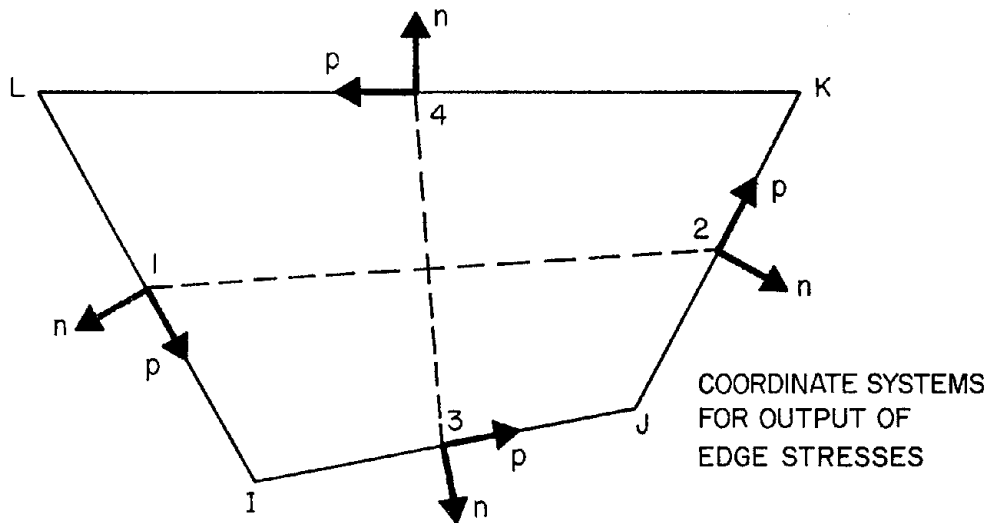
0 = origin of natural s-t coordinates (Figure 5-2). Points 1, 2, 3 and 4 are midpoints of sides. The points at which stresses are output depend on the value of n as described in the following table.

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).

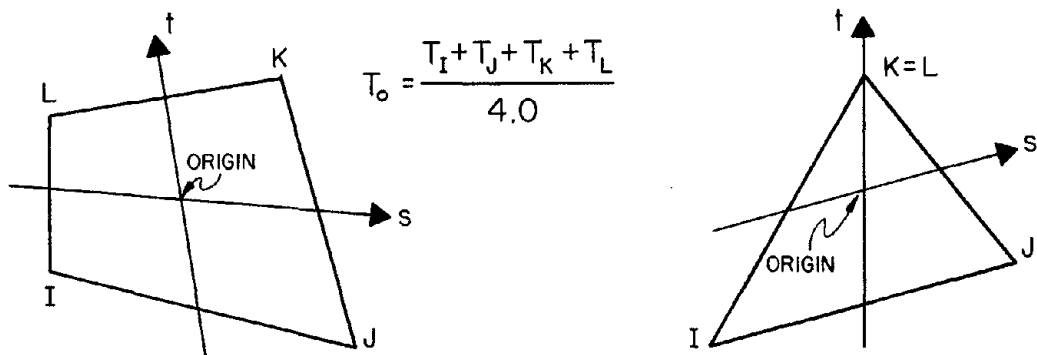


The stresses for an element are output under the following headings: S11, S22, S12, S33, S-MAX, S-MIN, ANGLE. The normal stresses S11 and S22 and the shear stress S12 are as described above. S-MAX and S-MIN are the principal stresses in the plane of the element and S33 is the third principal stress acting on 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.

Thermal Data - 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 pressure 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 pressures.

D. One Blank Card

- E. Element Load Case Multipliers (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	1 - 5	Element number	
	6 - 10	Global node point numbers corresponding to element nodes (See note III)	1 2 3 4 5 6 7 8
	11 - 15		
	16 - 20		
	21 - 25		
	26 - 30		
	31 - 35		
	36 - 40		
	41 - 45		
	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

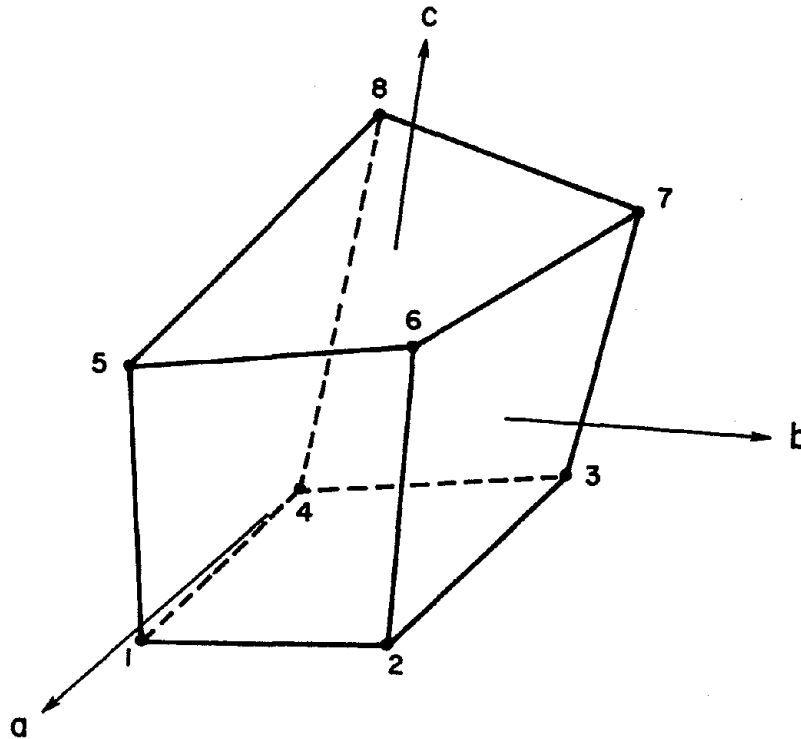
Computation time (for element stiffness) increases with the cube of the integration order. Therefore, the smallest satisfactory order should be used. This is found to be:

- 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

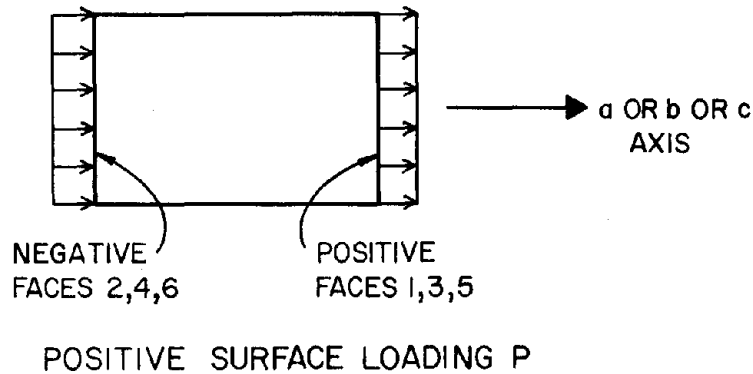
Element faces are numbered as follows:

Face 1 corresponds to + a direction	} Faces 1,3,5 are positive faces	
2 corresponds to - a direction		} Faces 2,4,6 are negative faces
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		

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.

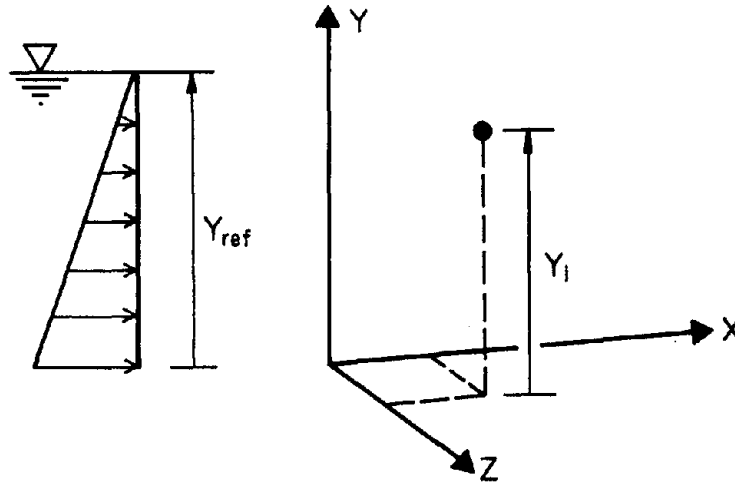


If the uniformly distributed surface loading P is input as a positive quantity then it describes pressure loading on faces 2, 4 or 6 and tensile loading on faces 1, 3 or 5. If P is input as a negative quantity then it describes tensile loading on faces 2, 4 or 6 and pressure on faces 1, 3 or 5.

(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.

$$\Delta T = T_{\text{avg}} - T_0$$

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

T_0 = stress free element temperature specified on the element card.

VII. Element Load Cases

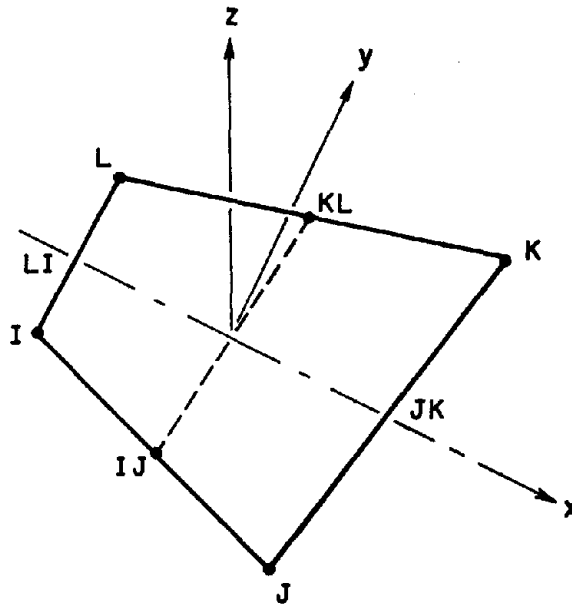
Element load case A consists of all the contributions from distributed loadings, thermal loadings and gravity loading for all the elements taken collectively.

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

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_z

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]

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

C. Element Load Multipliers (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. Element Cards (8I5,F10.0)

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

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

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

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

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

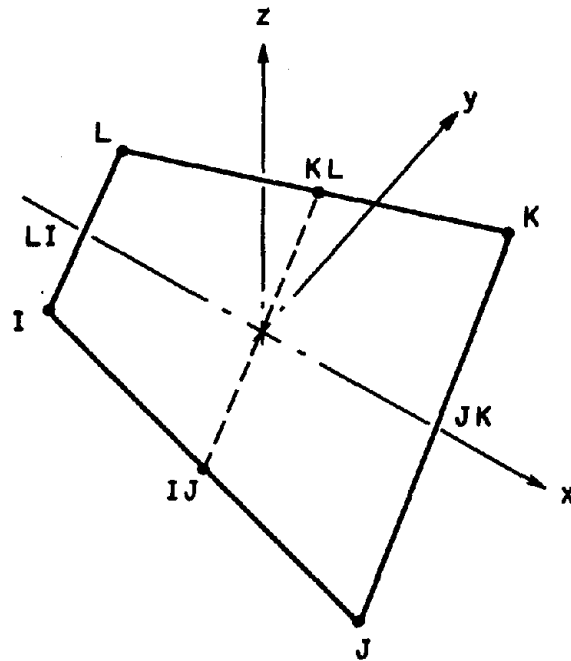
Material identification, element thickness, distributed lateral load, temperature and temperature gradient for generated elements are the same as the first element in the series. The last element card is always needed.

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 (215)

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
 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.

} Leave columns 11 - 25 blank if only node I is needed.

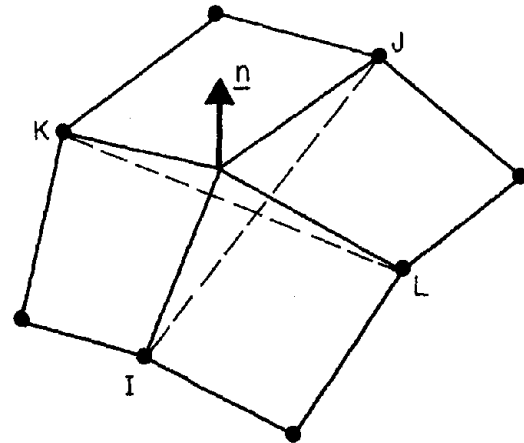
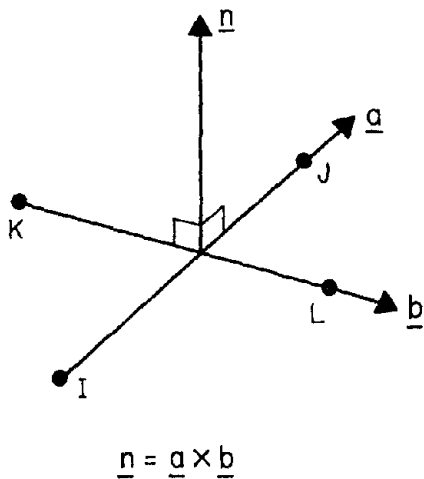
EXPLANATION OF INPUT DATA

(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})



ROTATIONAL CONSTRAINT
IN THIN SHELL ANALYSIS

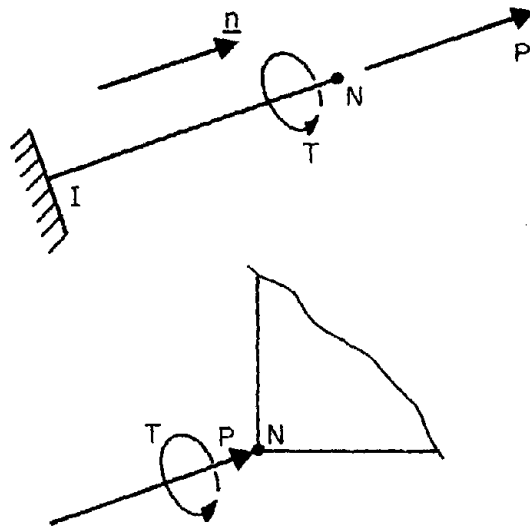
The four points I, J, K and L need not be unique. A useful application for the analysis of shallow thin shells employs the boundary element to approximate rotational constraint about the surface normal as shown above.

\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 acts as an extensional spring at node N with stiffness as specified in columns 61 - 70. Nodal loads applied to joint N are specified on the joint data card for node N (section B-4).

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 provides rotational spring stiffness about the axis of the element to node N. The stiffness is specified in columns 61 - 70 and arbitrary nodal loads at node N are specified on the joint data card for this node.

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 K_n

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 $0.0 \times 1.0 = 0.0$ while that for B is $2.0 \times 1.0 = 2.0$. 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

1 - 5	Element number
6 - 10	Integration order
11 - 15	Material number
16 - 20	Generation parameter (INC)
21 - 22	LSA
23 - 24	LSB
25 - 26	LSC
27 - 28	LSD
	} 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
31 - 40	Element temperature

Card 2: Columns

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

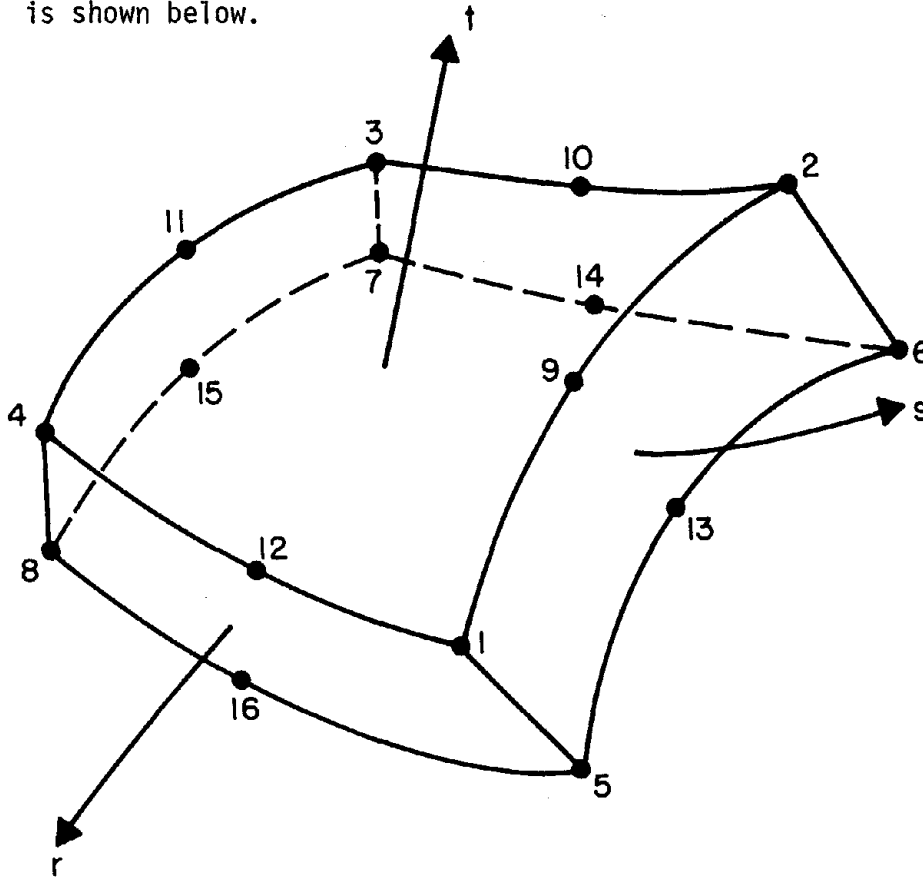
I. Element Generation

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. Element Face are Numbered as Follows:

Face 1 corresponds to $+r$ direction	} 1, 3, 5 are positive faces 2, 4, 6 are negative faces
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	

IV. Integration Order

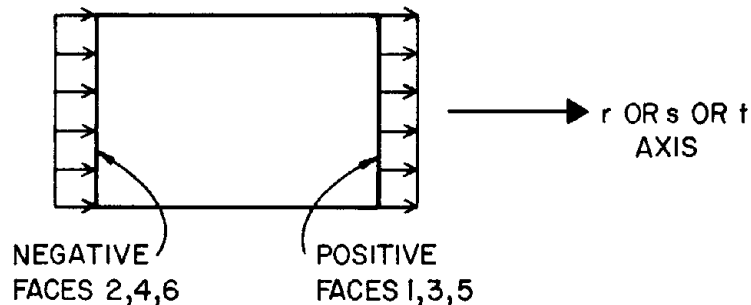
Computation time (for element stiffness) increases with the cube of the integration order. Therefore, the smallest satisfactory order should be used. This is found to be:

- 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.



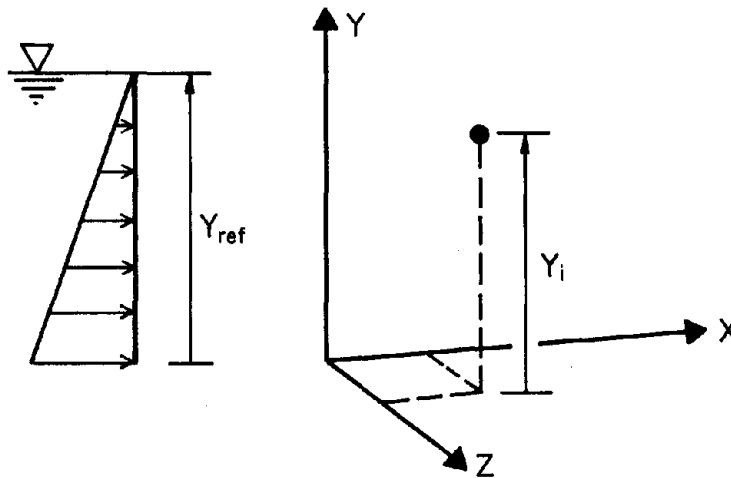
POSITIVE SURFACE LOADING P

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.

$$\Delta T = T_e - T_r$$

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.

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

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. 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:

$$n = 10 * (\text{number of joints}) + M$$

where

M = the maximum value of each of the following:

- (1) Truss elements

$$M = 5 * \text{NMAT} \quad \text{NMAT} = \text{number of material types}$$

- (2) Beam elements

$$M = 3 * \text{NMAT} + 12 * \text{NFIX} + 6 * \text{NPROP}$$

NFIX = number of fixed end force sets

NPROP = number of different beam properties

- (3) Plane stress and plane strain elements.

$$M = 4 * \text{NMAT} + 11 * \text{NMAT} * \text{NTC}$$

NTC = number of material temperatures

- (4) Axisymmetric quadrilateral

$$M = 4 * \text{NMAT}$$

- (5) Three-dimensional solid elements

$$M = 4 * \text{NMAT} + 4 * \text{NLD} + 2475 \quad \text{NLD} = \text{number of element load sets}$$

(6) Plate and shell elements

$$M = 12 * \text{NMAT}$$

(7) Boundary elements

$$M = 0$$

(8) Solid thick shell element

$$M = 4 * \text{NMAT} + 4 * \text{NLD} + 6615$$

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

$$n = 11 * (\text{number of joints})$$

(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 =

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

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 =

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

or for temporary storage during solution of equations

Number of locations on Tape 2 =

$$\left\lceil \frac{N_{\text{band}}}{N_{\text{eb}}} \right\rceil (N_{\text{band}} * N_{\text{eb}})$$

Where N_{band} = half band width of equations

N_{eb} = number of equations in a block

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

N_ℓ = number of load conditions

or, for storage of displacements

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

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


```

SSAP 1      PROGRAM      SAP2 (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE1,
SSAP 2      1 TAPE2,TAPE3,TAPE4,TAPE7,TAPE8)
SSAP 3      C ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** **
SSAP 4      C SAP2 A STATIC ANALYSIS PROGRAM FOR THREE-DIMENSIONAL STRUCTURES
SSAP 5      C REVISIED MARCH 1972
SSAP 6      C ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** ** **
SSAP 7      COMMON /JUNK / HED(12),JUK(210)
SSAP 8      COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
SSAP 9      COMMON / EM / QQQ(2594)
SSAP 10     DIMENSION T(7)
SSAP 11     C
SSAP 12     C PROGRAM CAPACITY CONTROLLED BY THE FOLLOWING TWO STATEMENTS ...
SSAP 13     C COMMON A(7000)
SSAP 14     MTOT=7000
SSAP 15     C
SSAP 16     C PROGRAM CONTROL DATA
SSAP 17     C
SSAP 18     5 CALL SECOND(T(1))
SSAP 19     READ (5,100) HED,NUMNP,NELTYP,LL,NF,NDYN
SSAP 20     IF (NUMNP.EQ.0) STOP
SSAP 21     WRITE(6,200) HED,NUMNP,NELTYP,LL
SSAP 22     C
SSAP 23     C INPUT JOINT DATA--ID ARRAY ON TAPE 8
SSAP 24     C
SSAP 25     N1=1
SSAP 26     N2=N1+6*NUMNP
SSAP 27     N3=N2+NUMNP
SSAP 28     N4=N3+NUMNP
SSAP 29     N5=N4+NUMNP
SSAP 30     N6=N5+NUMNP
SSAP 31     IF(N6.GT.MTOT) CALL ERROR(N6-MTOT)
SSAP 32     CALL INPUTJ(A(N1),A(N2),A(N3),A(N4),A(N5),NUMNP,NEQ)
SSAP 33     C
SSAP 34     C FORM ELEMENT STIFFNESSES--STIFF. ON TAPE 2 --STRESS MATRIX ON TAPE1
SSAP 35     C
SSAP 36     CALL SECOND(T(2))
SSAP 37     MBAND=0
SSAP 38     NUMEL=0
SSAP 39     REWIND 1
SSAP 40     REWIND 2
SSAP 41     DO 900 M=1,NELTYP
SSAP 42     READ (5,1001) NPAR
SSAP 43     WRITE (1) NPAR
SSAP 44     NUMEL=NUMEL+NPAR(2)
SSAP 45     MTYPE=NPAR(1)
SSAP 46     CALL ELTYPE(MTYPE)
SSAP 47     900 CONTINUE
SSAP 48     C
SSAP 49     C INPUT NODAL LOADS AND JOINT MASSES --- WRITE ON TAPE 3
SSAP 50     C
SSAP 51     NEQB=(MTOT-4*LL)/(MBAND+LL+1)/2
SSAP 52     NBLOCK=(NEQ-1)/NEQB +1
SSAP 53     IF (NEQB.GT.NEQ) NEQB=NEQ
SSAP 54     N3=N2+NEQB*LL
SSAP 55     N4=N3+6*LL
SSAP 56     WRITE (6,201) NEQ,MBAND,NEQB,NBLOCK
SSAP 57     CALL SECOND(T(3))
SSAP 58     CALL INL(A(N1),A(N2),A(N3),A(N4),NUMNP,NEQB,LL)
SSAP 59     CALL SECOND(T(4))
SSAP 60     C
SSAP 61     C FORM TOTAL STIFFNESS MATRIX --ON TAPE 4
SSAP 62     C
SSAP 63     NE2B=2*NEQB
SSAP 64     N2=N1+NEQB*MBAND
SSAP 65     N3=N2+NEQB*LL
SSAP 66     N4=N3+4*LL
SSAP 67     NN2=N1+NE2B*MBAND
SSAP 68     NN3=NN2+NE2B*LL
SSAP 69     NN4=NN3+4*LL
SSAP 70     CALL ADSTF(A(N1),A(N2),A(N3),A(N4),NUMEL,NBLOCK,NE2B,LL,MBAND)
SSAP 71     CALL SECOND(T(5))
SSAP 72     C
SSAP 73     C SOLVE FOR DISPLACEMENT UNKNOWNS
SSAP 74     C
SSAP 75     NSB=(MBAND+LL)*NEQB

```

```

SSAP 76     NSBB=NEQB*LL*(2+(MBAND-1)/NEQB)
SSAP 77     IF(NSBB.LT.NSB) NSBB=NSB
SSAP 78     N4=N3+NSBB
SSAP 79     CALL USOL(A(N1),A(N3),A(N4),NEQB,MBAND,LL,NBLOCK,NSB,4,3,7,2,2)
SSAP 80     CALL SECOND(T(6))
SSAP 81     C
SSAP 82     C PRINT DISPLACEMENT
SSAP 83     C
SSAP 84     N2=N1+NUMNP*6
SSAP 85     N3=N2+6*LL
SSAP 86     CALL PRINTD(A(N1),A(N2),A(N3),NEQB,NUMNP,LL,NBLOCK,NEQ,2)
SSAP 87     C
SSAP 88     C COMPUTE STRESSES
SSAP 89     C
SSAP 90     30 N2=N1+4*LL
SSAP 91     N3=N2+NEQB*LL
SSAP 92     LB=(MTOT-N3)/(NEQ +12)
SSAP 93     NOYN=0
SSAP 94     CALL STRESS(A(N1),A(N2),A(N3),NEQB,LB,LL,NEQ,NBLOCK)
SSAP 95     C
SSAP 96     CALL SECOND (T(7))
SSAP 97     DO 40 I=1,6
SSAP 98     40 T(I)=T(I+1)-T(I)
SSAP 99     T(7)=T(1)+T(2)+T(3)+T(4)+T(5)+T(6)
SSAP 100    WRITE (6,203) T
SSAP 101    C
SSAP 102    GO TO 5
SSAP 103    C
SSAP 104    100 FORMAT(12A6/615)
SSAP 105    200 FORMAT(1H1,12A6///
SSAP 106    . 28H NUMBER OF NODAL POINTS = ,15//
SSAP 107    . 28H NUMBER OF ELEMENT TYPES = ,15//
SSAP 108    . 28H NUMBER OF LOAD CASES = ,15)
SSAP 109    201 FORMAT(34H2 TOTAL NUMBER OF EQUATIONS =,15,
SSAP 110    1 /34H BANDWIDTH =,15,
SSAP 111    2 /34H NUMBER OF EQUATIONS IN A BLOCK =,15,
SSAP 112    3 /34H NUMBER OF BLOCKS =,15)
SSAP 113    203 FORMAT ( 12H2OVERALL LOG //
SSAP 114    . 33H NODAL POINT INPUT.....,F8.2//
SSAP 115    . 33H FORM ELEMENT STIFFNESSES.....,F8.2//
SSAP 116    . 33H INPUT NODAL LOADS.....,F8.2//
SSAP 117    . 33H FORM TOTAL STIFFNESS.....,F8.2//
SSAP 118    . 33H EQUATION SOLVING.....,F8.2//
SSAP 119    . 33H ELEMENT STRESSES.....,F8.2//
SSAP 120    . 33H TOTAL SOLUTION TIME.....,F8.2)
SSAP 121    C
SSAP 122    1001 FORMAT (1415)
SSAP 123    END

SSAP 124    SUBROUTINE INPUTJ(ID,X,Y,Z,T,NUMNP,NEQ)
SSAP 125    C-----E. WILSON JULY 1971
SSAP 126    DIMENSION X(1),Y(1),Z(1),ID(NUMNP,6),T(1)
SSAP 127    C
SSAP 128    C----- READ OR GENERATE NODAL POINT DATA-----
SSAP 129    WRITE (6,2000)
SSAP 130    WRITE (6,2001)
SSAP 131    NOLD=0
SSAP 132    10 READ (5,1000) N,((ID(N,I),I=1,6),X(N),Y(N),Z(N),KN,T(N))
SSAP 133    WRITE (6,2002) N,((ID(N,I),I=1,6),X(N),Y(N),Z(N),KN,T(N))
SSAP 134    IF(NOLD.EQ.0) GO TO 50
SSAP 135    C-----CHECK IF GENERATION IS REQUIRED-----
SSAP 136    DO 20 I=1,6
SSAP 137    IF((ID(N,I).EQ.0.AND.ID(NOLD,I).LT.0) ID(N,I)=ID(NOLD,I)
SSAP 138    20 CONTINUE
SSAP 139    IF(KN.EQ.0) GO TO 50
SSAP 140    NUM=(N-NOLD)/KN
SSAP 141    NUMN=NUM-1
SSAP 142    IF(NUMN.LT.1) GO TO 50
SSAP 143    XNUM=NUM

```

NOT REPRODUCIBLE

```

SSAP 144      DX=(X(N)-X(NOLD))/XNUM
SSAP 145      DY=(Y(N)-Y(NOLD))/XNUM
SSAP 146      DZ=(Z(N)-Z(NOLD))/XNUM
SSAP 147      DT=(T(N)-T(NOLD))/XNUM
SSAP 148      K=NOLD
SSAP 149      DO 30 J=1,NUMN
SSAP 150      KK=K
SSAP 151      K=K+KN
SSAP 152      X(K)=X(KK)+DX
SSAP 153      Y(K)=Y(KK)+DY
SSAP 154      Z(K)=Z(KK)+DZ
SSAP 155      T(K)=T(KK)+DT
SSAP 156      DO 30 I=1,6
SSAP 157      ID(K,I)=ID(KK,I)
SSAP 158      IF (ID(K,I).GT.1) ID(K,I)=ID(KK,I)+KN
SSAP 159      30 CONTINUE
SSAP 160      C
SSAP 161      50 NOLD=N
SSAP 162      IF(N.NE.NUMNP) GO TO 10
SSAP 163      C
SSAP 164      C----- PRINT ALL NODAL POINT DATA-----
SSAP 165      C
SSAP 166      WRITE (6,2003)
SSAP 167      WRITE (6,2001)
SSAP 168      WRITE (6,2005) (N,{ID(N,I),I=1,6},X(N),Y(N),Z(N),T(N),N=1,NUMNP)
SSAP 169      C
SSAP 170      C-----NUMBER UNKNOWNNS AND SET MASTER NODES NEGATIVE-----
SSAP 171      C
SSAP 172      NEQ=0
SSAP 173      DO 60 N=1,NUMNP
SSAP 174      DO 60 I=1,6
SSAP 175      ID(N,I)=IABS(ID(N,I))
SSAP 176      IF(ID(N,I)-1) 57,58,59
SSAP 177      57 NEQ=NEQ+1
SSAP 178      ID(N,I)=NEQ
SSAP 179      GO TO 60
SSAP 180      58 ID(N,I)=0
SSAP 181      GO TO 60
SSAP 182      59 ID(N,I)=-ID(N,I)
SSAP 183      60 CONTINUE
SSAP 184      WRITE (6,2004) (N,{ID(N,I),I=1,6},N=1,NUMNP)
SSAP 185      REWIND 8
SSAP 186      WRITE (8) ID
SSAP 187      C
SSAP 188      RETURN
SSAP 189      1000 FORMAT (7I5,3F10.0,I5,F10.0)
SSAP 190      2000 FORMAT (///23H NODAL POINT INPUT DATA )
SSAP 191      2001 FORMAT (5HNODE 3X 24HBOUNDARY CONDITION CODES 11X
SSAP 192      . 23HNODAL POINT COORDINATES / 7H NUMBER 2X 1HX 4X 1HY 4X 1HZ 3X
SSAP 193      . 2HXX 3X 2HY 3X 2HZ12X 1HX 12X 1HY 12X 1HZ 12X 1HT )
SSAP 194      2002 FORMAT (I5,6I5,3F13.3,I5,F13.3)
SSAP 195      2003 FORMAT (///21H GENERATED NODAL DATA)
SSAP 196      2004 FORMAT (///17H EQUATION NUMBERS/
SSAP 197      1 35H N X Y Z XX YY ZZ /(7I5))
SSAP 198      2005 FORMAT (I5,6I5,4F13.3)
SSAP 199      END

```

```

SSAP 200      SUBROUTINE ELTYPE(MTYPE)
SSAP 201      GO TO (1,2,3,4,5,6,7,8) MTYPE
SSAP 202      C
SSAP 203      C      THREE DIMENSIONAL TRUSS ELEMENTS
SSAP 204      C
SSAP 205      1 CALL TRUSS
SSAP 206      GO TO 900
SSAP 207      C
SSAP 208      C      THREDF DIMENSIONAL BEAM ELEMENTS
SSAP 209      C
SSAP 210      2 CALL BEAM
SSAP 211      GO TO 900

```

```

SSAP 212      C
SSAP 213      C      PLANE STRESS ELEMENTS
SSAP 214      C
SSAP 215      3 CALL PLANE
SSAP 216      GO TO 900
SSAP 217      C
SSAP 218      C      AXISYMMETRIC SOLID ELEMENTS
SSAP 219      C
SSAP 220      4 CALL PLANE
SSAP 221      GO TO 900
SSAP 222      C
SSAP 223      C      THREE DIMENSIONAL SOLID ELEMENTS
SSAP 224      C
SSAP 225      5 CALL THREEED
SSAP 226      GO TO 900
SSAP 227      C
SSAP 228      C      PLATE BENDING ELEMENTS
SSAP 229      C
SSAP 230      6 CALL SHELL
SSAP 231      GO TO 900
SSAP 232      C
SSAP 233      C      7 CALL BOUND
SSAP 234      GO TO 900
SSAP 235      C
SSAP 236      C      THICK SHELL ELEMENTS
SSAP 237      C
SSAP 238      C      8 CALL THKSHL
SSAP 239      C
SSAP 240      C      900 RETURN
SSAP 241      END
SSAP 242

SSAP 243      SUBROUTINE INL(ID,B,TR,TMASS,NUMNP,NEQB,LL)
SSAP 244      C
SSAP 245      C      INPUT NODAL LOADS AND MASSES
SSAP 246      C
SSAP 247      DIMENSION ID(NUMNP,6),B(NEQB,LL),TR(6,LL),TMASS(NEQB)
SSAP 248      COMMON / JUNK / R(6),TXM(6)
SSAP 249      C
SSAP 250      NT=3
SSAP 251      REWIND NT
SSAP 252      KSHF=0
SSAP 253      WRITE (6,2002)
SSAP 254      DO 750 I=1,NEQB
SSAP 255      TMASS(I)=0.
SSAP 256      DO 750 K=1,LL
SSAP 257      750 B(I,K)=0.0
SSAP 258      C
SSAP 259      DO 900 NN=1,NUMNP
SSAP 260      C
SSAP 261      DO 100 I=1,6
SSAP 262      TXM(I)=0.
SSAP 263      DO 100 J=1,LL
SSAP 264      100 TR(I,J)=0.0
SSAP 265      C
SSAP 266      IF(NN.EQ.1) GO TO 300
SSAP 267      150 IF(NN.NE.NN) GO TO 400
SSAP 268      DO 200 I=1,6
SSAP 269      IF (L) 180,180,190
SSAP 270      180 TXM(I)=R(I)
SSAP 271      GO TO 200
SSAP 272      190 TR(I,L)=R(I)
SSAP 273      200 CONTINUE
SSAP 274      300 READ (5,1001) N,L,R
SSAP 275      IF (N.EQ.0) GO TO 150
SSAP 276      WRITE(6,2001) N,L,R
SSAP 277      GO TO 150
SSAP 278      C
SSAP 279      400 DO 800 J=1,6

```

```

SSAP 280      II=ID(NN,J)-KSHF
SSAP 281      IF (II) 800,800,500
SSAP 282      500 DO 600 K=1,LL
SSAP 283      500 B(II,K)=TR(J,K)
SSAP 284      TMASS(II)=TXM(J)
SSAP 285      610 IF(II.NE.NEQB) GO TO 800
SSAP 286      WRITE (NT) B,TMASS
SSAP 287      KSHF=KSHF+NEQB
SSAP 288      DO 700 I=1,NEQB
SSAP 289      TMASS(II)=0.
SSAP 290      DO 700 K=1,LL
SSAP 291      700 B(I,K)=0.0
SSAP 292      800 CONTINUE
SSAP 293      900 CONTINUE
SSAP 294      WRITE (NT) B,TMASS
SSAP 295      C
SSAP 296      RETURN
SSAP 297      1001 FORMAT (2I5,7F10.4)
SSAP 298      2001 FORMAT (2I5,7F10.3)
SSAP 299      2002 FORMAT (23H2.....NODAL POINT LOADS // 10H NODE LOAD 23X
SSAP 300      . 14HAPPLIED LOADS / 10H NO. CASE 6X 2HRX 8X
SSAP 301      . 2HRX 8X 2HRZ 8X 2HMX 8X 2HMY 8X 2HMZ )
SSAP 302      END

```

```

SSAP 303      SUBROUTINE ERROR(N)
SSAP 304      WRITE (6,2000) N
SSAP 305      2000 FORMAT (// 20H STORAGE EXCEEDED BY I6)
SSAP 306      STOP
SSAP 307      END

```

```

SSAP 308      SUBROUTINE ADDSTFIA,B,STR,TMASS,NUMEL,NBLOCK,NE2B,LL,MBAND)
SSAP 309      C
SSAP 310      DIMENSION A(NE2B,MBAND),B(NE2B,LL),STR(4,LL),TMASS(NE2B),SS(1)
SSAP 311      COMMON /EM/ LRD,ND,L4(2592)
SSAP 312      EQUIVALENCE (SS,ND)
SSAP 313      NEQB=NE2B/2.
SSAP 314      K=NEQB+1
SSAP 315      X=NBLOCK
SSAP 316      MB=SQRT(X)
SSAP 317      MB=MB/2+1
SSAP 318      NEBB=MB*NE2B
SSAP 319      MM=1
SSAP 320      C
SSAP 321      NSHIFT=0
SSAP 322      REWIND 3
SSAP 323      REWIND 4
SSAP 324      C
SSAP 325      READ ELEMENT LOAD MULTIPLIERS
SSAP 326      C
SSAP 327      WRITE (6,2000)
SSAP 328      DO 50 L=1,LL
SSAP 329      READ (5,1002) (STR(I,L),I=1,4)
SSAP 330      50 WRITE (6,2002) L,(STR(I,L),I=1,4)
SSAP 331      WRITE (6) STR
SSAP 332      C
SSAP 333      FORM EQUATIONS IN BLOCKS ( 2 BLOCKS AT A TIME)
SSAP 334      C
SSAP 335      DO 1000 M=1,NBLOCK ,2
SSAP 336      DO 100 I=1,NE2B
SSAP 337      DO 100 J=1,MBAND
SSAP 338      100 A(I,J)=0.
SSAP 339      READ (3) ((B(I,L),I=1,NEQB),L=1,LL),(TMASS(I),I=1,NEQB)
SSAP 340      IF (M.EQ.NBLOCK) GO TO 200

```

```

SSAP 341      READ (3) ((B(I,L),I=K,NE2B),L=1,LL),(TMASS(I),I=K,NE2B)
SSAP 342      200 CONTINUE
SSAP 343      C
SSAP 344      REWIND 7
SSAP 345      REWIND 2
SSAP 346      NA=7
SSAP 347      NUME=NUM7
SSAP 348      IF (MM.NE.1) GO TO 75
SSAP 349      NA=2
SSAP 350      NUME=NUMEL
SSAP 351      NUM7 = 0
SSAP 352      C
SSAP 353      75 DO 700 N=1,NUME
SSAP 354      READ (NA) LRD,(SS(I),I=1,LRD)
SSAP 355      DO 600 I=1,ND
SSAP 356      LMN=L-LM(I)
SSAP 357      II=LM(I)-NSHIFT
SSAP 358      IF (II.LE.0.OR.II.GT.NE2B) GO TO 600
SSAP 359      DO 300 L=1,LL
SSAP 360      DO 300 J=1,4
SSAP 361      KK=ND*(NC+J)+1
SSAP 362      300 B(II,L)=B(II,L)+SS(I+KK)*STR(J,L)
SSAP 363      DO 500 J=1,ND
SSAP 364      JJ=LM(J)+LMN
SSAP 365      IF(JJ) 500,500,390
SSAP 366      390 KK=ND*J+1
SSAP 367      400 A(II,JJ)=A(II,JJ)+SS(I+KK)
SSAP 368      500 CONTINUE
SSAP 369      600 CONTINUE
SSAP 370      C
SSAP 371      DETERMINE IF STIFFNESS IS TO BE PLACED ON TAPE 7
SSAP 372      C
SSAP 373      IF (MM.GT.1) GO TO 700
SSAP 374      DO 650 I=1,ND
SSAP 375      II=LM(I) -NSHIFT
SSAP 376      IF(II.GT.NE2B.AND.II.LE.NEBB) GO TO 660
SSAP 377      650 CONTINUE
SSAP 378      GO TO 700
SSAP 379      660 WRITE (7) LRD,(SS(I),I=1,LRD)
SSAP 380      NUM7=NUM7+1
SSAP 381      C
SSAP 382      700 CONTINUE
SSAP 383      WRITE(4) ((A(I,J),I=1,NEQB),J=1,MBAND),((B(I,L),I=1,NEQB),L=1,LL)
SSAP 384      IF(M.EQ.NBLOCK) GO TO 1000
SSAP 385      WRITE(4) ((A(I,J),I=K,NE2B),J=1,MBAND),((B(I,L),I=K,NE2B),L=1,LL)
SSAP 386      IF (MM.EQ.MB) MM=0
SSAP 387      MM=MM+1
SSAP 388      1000 NSHIFT=NSHIFT+NE2B
SSAP 389      C
SSAP 390      RETURN
SSAP 391      1002 FORMAT (4F10.0)
SSAP 392      2000 FORMAT (10H2STRUCTURE 12X 25HELEMENT LOAD MULTIPLIERS /
SSAP 393      . 10H LOAD CASE 9X 1HA 9X 1HB 9X 1HC 9X 1HD/)
SSAP 394      2002 FORMAT (I6,7X,4F10.3)
SSAP 395      END

```

```

SSAP 396      SUBROUTINE PRINTD(ID,D,B,NEQB,NUMNP,LL,NBLOCK,NEQ,NT)
SSAP 397      C
SSAP 398      DIMENSION ID(NUMNP,6),B(NEQB,LL),D(6,LL)
SSAP 399      C
SSAP 400      REWIND NT
SSAP 401      REWIND 9
SSAP 402      READ (8) ID
SSAP 403      M=NEQ
SSAP 404      NN=NEQB*NBLOCK
SSAP 405      WRITE (6,2003)
SSAP 406      N=NUMNP
SSAP 407      C
SSAP 408      DO 500 KK=1,NUMNP

```

```

SSAP 409 C
SSAP 410 I=6
SSAP 411 DO 250 II=1,6
SSAP 412 DO 100 L=1,LL
SSAP 413 100 D(I,L)=0.
SSAP 414 IF(M.GT.NN) GO TO 150
SSAP 415 IF (M.EQ.0) GO TO 150
SSAP 416 READ (NT) B
SSAP 417 NN=NN-NEQB
SSAP 418 150 IF(I.D(N,I).LT.1) GO TO 250
SSAP 419 K=M-NN
SSAP 420 M=M-1
SSAP 421 C
SSAP 422 DO 200 L=1,LL
SSAP 423 200 D(I,L)=B(K,L)
SSAP 424 250 I=I-1
SSAP 425 C
SSAP 426 WRITE (6,2004) N,(L,(D(I,L),I=1,6),L=1,LL)
SSAP 427 C
SSAP 428 500 N=N-1
SSAP 429 C
SSAP 430 RETURN
SSAP 431 C
SSAP 432 2003 FORMAT (40H1.....NODE DISPLACEMENTS AND ROTATIONS//
SSAP 433 . 5H NODE 5H LOAD 11X 1HX 11X 1HY 11X 1HZ 9X 2HXX
SSAP 434 . 9X 2HYY 9X 2HZZ)
SSAP 435 2004 FORMAT (1H0,14,15,1P3E12.3,3E11.2/(110,3E12.3,3E11.2))
SSAP 436 END

```

```

SSAP 437 SUBROUTINE USOL (A, B, MAXB, NEQB, MB, LL, NBLOCK, NSB, NORG, NBKS, NT1,
SSAP 438 NT2, NRST)
SSAP 439 DIMENSION A(NSB), B(NSB), MAXB(NEQB)
SSAP 440 C-----
SSAP 441 NC=MB+LL
SSAP 442 NBR=(MB-1)/NEQB+1
SSAP 443 INC=NEQB-1
SSAP 444 NMB=NEQB*MB
SSAP 445 N2=NT2
SSAP 446 NI=NT1
SSAP 447 REWIND NORG
SSAP 448 REWIND NBKS
SSAP 449 C
SSAP 450 C---- REDUCE EQUATIONS BLOCK-BY-BLOCK -----
SSAP 451 C
SSAP 452 DO 900 N=1, NBLOCK
SSAP 453 IF (N.GT.1.AND.NBR.EQ.1) GO TO 110
SSAP 454 IF (NBR.EQ.1) GO TO 105
SSAP 455 REWIND N1
SSAP 456 REWIND N2
SSAP 457 105 NI=NI
SSAP 458 IF(N.EQ.1) NI=NORG
SSAP 459 READ (NI) A
SSAP 460 110 DO 300 I=1, NEQB
SSAP 461 D=A(I)
SSAP 462 IF(D) 115, 300, 120
SSAP 463 115 M=NEQB*(N-1)+1
SSAP 464 WRITE (6,116) M,D
SSAP 465 116 FORMAT (33H0SET OF EQUATIONS MAY BE SINGULAR /
SSAP 466 . 26H DIAGCNAL TERM OF EQUATION 18, 8H EQUALS 1PE12.4)
SSAP 467 C
SSAP 468 120 II=I
SSAP 469 DO 125 J=2, NC
SSAP 470 II=II+NEQB
SSAP 471 125 A(II)=A(II)/D
SSAP 472 C
SSAP 473 DO 130 J=I, NMB, NEQB
SSAP 474 IF (A(J).NE.0.) MAXB(II)=J
SSAP 475 130 CONTINUE
SSAP 476 C

```

```

SSAP 477 JL=J+1
SSAP 478 IF (JL.GT.NEQB) GO TO 300
SSAP 479 II=J
SSAP 480 DO 200 J=JL, NEQB
SSAP 481 II=II+NEQB
SSAP 482 IF(II.GT.NMB) GO TO 200
SSAP 483 C=A(II)
SSAP 484 IF (C.EQ.0.0) GO TO 200
SSAP 485 C=C*A(II)
SSAP 486 C
SSAP 487 KK=J-II
SSAP 488 MAX=MAXB(II)
SSAP 489 DO 150 JJ=II, MAX, NEQB
SSAP 490 150 A(JJ+KK)=A(JJ+KK)-C*A(JJ)
SSAP 491 C
SSAP 492 KK=J+NMB
SSAP 493 JJ=J+NMB
SSAP 494 DO 175 L=1, LL
SSAP 495 A(KK)=A(KK)-C*A(JJ)
SSAP 496 KK=KK+NEQB
SSAP 497 175 JJ=JJ+NEQB
SSAP 498 200 CONTINUE
SSAP 499 300 CONTINUE
SSAP 500 WRITE (NBKS) A, MAXB
SSAP 501 C
SSAP 502 C---- SUBSTITUTE INTO REMAINING EQUATIONS -----
SSAP 503 C
SSAP 504 DO 800 NN=1, NBR
SSAP 505 IF(N+NN.GT.NBLOCK) GO TO 800
SSAP 506 NI=N1
SSAP 507 IF(N.EQ.1) NI=NORG
SSAP 508 IF(NN.EQ.NBR) NI=NORG
SSAP 509 READ (NI) B
SSAP 510 IL=1+NN*NEQB*NEQB
SSAP 511 DO 700 I=1, NEQB
SSAP 512 II=IL
SSAP 513 DO 690 K=1, NEQB
SSAP 514 IF (II.GT.NMB) GO TO 690
SSAP 515 C=A(II)
SSAP 516 IF (C.EQ.0.0) GO TO 690
SSAP 517 C=C*A(K)
SSAP 518 MAX=MAXB(K)
SSAP 519 C
SSAP 520 KK=I-II
SSAP 521 DO 640 JJ=II, MAX, NEQB
SSAP 522 640 B(JJ+KK)=B(JJ+KK)-C*A(JJ)
SSAP 523 C
SSAP 524 KK=I+NMB
SSAP 525 JJ=K+NMB
SSAP 526 DO 650 L=1, LL
SSAP 527 B(KK)=B(KK)-C*A(JJ)
SSAP 528 KK=KK+NEQB
SSAP 529 650 JJ=JJ+NEQB
SSAP 530 C
SSAP 531 690 II=II+NEQB
SSAP 532 700 IL=IL+NEQB
SSAP 533 C
SSAP 534 IF(NBR.NE.1) GO TO 750
SSAP 535 DO 740 I=1, NSB
SSAP 536 740 A(II)=B(II)
SSAP 537 GO TO 800
SSAP 538 750 WRITE (N2) B
SSAP 539 800 CONTINUE
SSAP 540 C
SSAP 541 M=N1
SSAP 542 N1=N2
SSAP 543 900 N2=M
SSAP 544 C
SSAP 545 C---- BACKSUBSTITUTION - RESULTS ON TAPE NRST -----
SSAP 546 C
SSAP 547 LS=LL*NEQB
SSAP 548 NEB=NEQB*(NBR+1)
SSAP 549 NUM=NBR*NEQB
SSAP 550 MAX=NEB*LL
SSAP 551 DO 905 I=1, MAX

```

```

SSAP 552 905 B(I)=0.
SSAP 553 REWIND NRST
SSAP 554 C-----
SSAP 555 DO 1000 N=1,NBLOCK
SSAP 556 BACKSPACE NBKS
SSAP 557 READ (NBKS) A,MAXB
SSAP 558 BACKSPACE NBKS
SSAP 559 DO 910 L=1,LL
SSAP 560 K=L*NEB
SSAP 561 DO 910 J=1,NUM
SSAP 562 I=K-NEQB
SSAP 563 B(K)=B(I)
SSAP 564 910 K=K-1
SSAP 565 C
SSAP 566 I=NMB
SSAP 567 DO 920 L=1,LL
SSAP 568 K=(L-1)*NEB
SSAP 569 DO 920 J=1,NEQB
SSAP 570 I=I+1
SSAP 571 K=K+1
SSAP 572 920 B(K)=A(I)
SSAP 573 C
SSAP 574 DO 955 I=1,NEQB
SSAP 575 J=NEQB+1-I
SSAP 576 MAX=MAXB(J)
SSAP 577 IF (A(J).EQ.0.) GO TO 955
SSAP 578 DO 950 L=1,LL
SSAP 579 KK=J+(L-1)*NEB
SSAP 580 JJ=KK+1
SSAP 581 IL=J+NEQB
SSAP 582 C=B(KK)
SSAP 583 DO 940 II=IL,MAX,NEQB
SSAP 584 C=C-A(II)*B(JJ)
SSAP 585 940 JJ=JJ+1
SSAP 586 950 B(KK)=C
SSAP 587 955 CONTINUE
SSAP 588 C
SSAP 589 I=0
SSAP 590 DO 960 L=1,LL
SSAP 591 K=(L-1)*NEB
SSAP 592 DO 960 J=1,NEQB
SSAP 593 K=K+1
SSAP 594 I=I+1
SSAP 595 960 A(II)=B(K)
SSAP 596 C
SSAP 597 WRITE (NRST) (A(II),I=1,LS)
SSAP 598 1000 CONTINUE
SSAP 599 C-----
SSAP 600 RETURN
SSAP 601 END

```

```

SSAP 602 SUBROUTINE STRESS(STR,B,D,NEQB,LB,LL,NEQ,NBLOCK)
SSAP 603 C
SSAP 604 DIMENSION D(NEQ,LB),B(NEQB,LL),STR(4,LL)
SSAP 605 COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,MEQ
SSAP 606 COMMON /JUNK/ LT,LH
SSAP 607 C
SSAP 608 READ (8) STR
SSAP 609 NT=(LL-1)/LB +1
SSAP 610 LH=0
SSAP 611 REWIND 3
SSAP 612 C
SSAP 613 DO 1000 II=1,NT
SSAP 614 C
SSAP 615 LT =LH+1
SSAP 616 LLT=1-LL
SSAP 617 LH=LT+LB-1
SSAP 618 IF(LH.GT.LL) LH=LL
SSAP 619 C

```

```

SSAP 620 C MOVE DISPLACEMENTS INTO CORE FOR LB LOAD CONDITIONS
SSAP 621 C
SSAP 622 REWIND 2
SSAP 623 IF(NOYN.EQ.3) READ (2)
SSAP 624 NQ=NEQB*NBLOCK
SSAP 625 DO 200 NN=1,NBLOCK
SSAP 626 READ (2) B
SSAP 627 N=NEQB
SSAP 628 IF (NN.EQ.1) N=NEQ-NQ+NEQB
SSAP 629 NQ=NQ-NEQB
SSAP 630 DO 200 J=1,N
SSAP 631 I=NQ+J
SSAP 632 DO 200 L=LT,LH
SSAP 633 K=L+LLT
SSAP 634 200 D(I,K)=B(J,L)
SSAP 635 LK=LH-LT+1
SSAP 636 C
SSAP 637 C CALCULATE STRESSES FOR ALL ELEMENTS FOR LB LOAD CONDITIONS
SSAP 638 C
SSAP 639 REWIND 1
SSAP 640 DO 1000 M=1,NELTYP
SSAP 641 READ (1) NPAR
SSAP 642 MTYPE=NPAR(1)
SSAP 643 NPAR(1)=0
SSAP 644 CALL ELTYP(MTYPE)
SSAP 645 1000 CONTINUE
SSAP 646 C
SSAP 647 RETURN
SSAP 648 END

```

```

SSAP 649 SUBROUTINE CALBAN(MBAND,NDIF,LM,XM,S,P,ND,NOM)
SSAP 650 C-----CALCULATES BAND WIDTH AND WRITES STIFFNESS MATRIX ON TAPE 2
SSAP 651 DIMENSION LM(1),XM(1),S(NDM,NDM),P(NDM,4)
SSAP 652 MIN=100000
SSAP 653 MAX=0
SSAP 654 DO 800 L=1,ND
SSAP 655 IF (LM(L).EQ.0) GO TO 800
SSAP 656 IF (LM(L).GT.MAX) MAX=LM(L)
SSAP 657 IF (LM(L).LT.MIN) MIN=LM(L)
SSAP 658 800 CONTINUE
SSAP 659 NDIF=MAX-MIN+1
SSAP 660 IF (NDIF.GT.MBAND) MBAND=NDIF
SSAP 661 LRD=1+ND*(ND+6)
SSAP 662 WRITE (2) LRD,ND,{LM(I),I=1,ND},{S(I,J),I=1,ND},J=1,ND),
SSAP 663 1 ({P(I,J),I=1,ND},J=1,4),(XM(I),I=1,ND)
SSAP 664 RETURN
SSAP 665 END

```

```

SSAP 666 SUBROUTINE STRSC(STR,D,NEQ,NTAG)
SSAP 667 DIMENSION STR(4,1),D(NEQ,1)
SSAP 668 COMMON /JUNK/ LT,LH,L,SG(20),SIG(7),EXTRA(150)
SSAP 669 COMMON /EM/ NS,ND,LM(48),B(48,48),TI(48,4)
SSAP 670 C
SSAP 671 IF (NTAG.EQ.0) GO TO 800
SSAP 672 LL=L-LT+1
SSAP 673 DO 300 I=1,NS
SSAP 674 SG(I)=0.0
SSAP 675 DO 300 J=1,4
SSAP 676 300 SG(I)=SG(I)+TI(I,J)*STR(J,L)
SSAP 677 DO 500 J=1,ND
SSAP 678 JJ=LM(J)
SSAP 679 IF(JJ.EQ.0) GO TO 500
SSAP 680 DO 400 I=1,NS

```

```

SSAP 681 400 SG(I)=SG(I)+B(I,J)*D(JJ,LL)
SSAP 682 C
SSAP 683 500 CONTINUE
SSAP 684 GO TO 900
SSAP 685 800 READ (1) ND,NS, (LM(I),I=1,ND), (( B(I,J),I=1,NS),J=1,ND),
SSAP 686 1 ((TI(I,J),I=1,NS),J=1,4)
SSAP 687 900 RETURN
SSAP 688 END

```

```

SSAP 735 FUNCTION DOT(A,B)
SSAP 736 DIMENSION A(4),B(4)
SSAP 737 DOT=A(1)*B(1)+A(2)*B(2)+A(3)*B(3)
SSAP 738 RETURN
SSAP 739 END

```

```

SSAP 689 SUBROUTINE PCSINV(A,NMAX,NDD)
SSAP 690 C
SSAP 691 DIMENSION A(NDD,NDD)
SSAP 692 C
SSAP 693 DO 200 N=1,NMAX
SSAP 694 C
SSAP 695 D=A(N,N)
SSAP 696 DO 100 J=1,NMAX
SSAP 697 100 A(N,J)=-A(N,J)/D
SSAP 698 C
SSAP 699 DO 150 I=1,NMAX
SSAP 700 IF(N-I) 110,150,110
SSAP 701 110 DO 140 J=1,NMAX
SSAP 702 IF(N-J) 120,140,120
SSAP 703 120 A(I,J)=A(I,J)+A(I,N)*A(N,J)
SSAP 704 140 CONTINUE
SSAP 705 150 A(I,N)=A(I,N)/D
SSAP 706 C
SSAP 707 A(N,N)=1.0/D
SSAP 708 C
SSAP 709 200 CONTINUE
SSAP 710 C
SSAP 711 RETURN
SSAP 712 END

```

```

SSAP 713 SUBROUTINE VECTOR(V,XI,YI,ZI,XJ,YJ,ZJ)
SSAP 714 DIMENSION V(4)
SSAP 715 X=XJ-XI
SSAP 716 Y=YJ-YI
SSAP 717 Z=ZJ-ZI
SSAP 718 V(4)=SQRT(X*X+Y*Y+Z*Z)
SSAP 719 V(3)=Z/V(4)
SSAP 720 V(2)=Y/V(4)
SSAP 721 V(1)=X/V(4)
SSAP 722 RETURN
SSAP 723 END

```

```

SSAP 724 SUBROUTINE CROSS(A,B,C)
SSAP 725 DIMENSION A(4),B(4),C(4)
SSAP 726 X=A(2)*B(3)-A(3)*B(2)
SSAP 727 Y=A(3)*B(1)-A(1)*B(3)
SSAP 728 Z=A(1)*B(2)-A(2)*B(1)
SSAP 729 C(4)=SQRT(X*X+Y*Y+Z*Z)
SSAP 730 C(3)=Z/C(4)
SSAP 731 C(2)=Y/C(4)
SSAP 732 C(1)=X/C(4)
SSAP 733 RETURN
SSAP 734 END

```

```

TRUS 1 SUBROUTINE TRUSS
TRUS 2 C
TRUS 3 COMMON A(1)
TRUS 4 COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
TRUS 5 COMMON /JUNK / LT,LH,L,SIG(20)
TRUS 6 C
TRUS 7 IF(NPAR(1),EQ,0) GO TO 500
TRUS 8 N6=N5+NUMNP
TRUS 9 N7=N6+NPAR(3)
TRUS 10 N8=N7+NPAR(3)
TRUS 11 N9=N8+NPAR(3)
TRUS 12 N10=N9+NPAR(3)
TRUS 13 MM=N10+NPAR(3)-MTOT
TRUS 14 IF(MM.GT.0) CALL ERROR(MM)
TRUS 15 CALL RUSS(A(N1),A(N2),A(N3),A(N4),A(N5),A(N6),A(N7),A(N8),A(N9),
TRUS 16 A(N10),NUMNP)
TRUS 17 RETURN
TRUS 18 C
TRUS 19 500 WRITE (6,2002)
TRUS 20 NUME=NPAR(2)
TRUS 21 DO 800 HM=1,NUME
TRUS 22 CALL STRSC (A(N1),A(N3),NEQ,0)
TRUS 23 WRITE (6,2001)
TRUS 24 DO 800 L=LT,LH
TRUS 25 CALL STRSC (A(N1),A(N3),NEQ,1)
TRUS 26 WRITE (6,3002) MM,L,SIG(1),SIG(2)
TRUS 27 800 CONTINUE
TRUS 28 RETURN
TRUS 29 C
TRUS 30 2001 FORMAT (/)
TRUS 31 2002 FORMAT (//23H TRUSS MEMBER ACTIONS //
TRUS 32 46H0 MEMBER LOAD STRESS FORCE )
TRUS 33 3002 FORMAT (2I8,F15.5,F15.3)
TRUS 34 END

```

```

TRUS 35 SUBROUTINE RUSS (ID,X,Y,Z,T,E,THERM,DEN,AREA,WT,NUMNP)
TRUS 36 C
TRUS 37 DIMENSION X(1),Y(1),Z(1),ID(NUMNP,1),E(1),THERM(1),DEN(1),AREA(1)
TRUS 38 ,T(1),WT(1)
TRUS 39 COMMON /ELPAR/ NPAR(14),NNMNN,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
TRUS 40 COMMON /EM/LM(24),ND,NS,S(24,24),P(24,4),XM(24),ST(12,24),TT(12,4)
TRUS 41 COMMON /JUNK/ EMUL(4,4),I,J,K,L,M,N,II,JJ,KK,MTYPE,TEMP,DX,DY,DZ,
TRUS 42 XL2,XL,XX,YY,F,FT,FX,FY,FZ,MIN,MAX,NDIF,KKK,TEM,MTYP
TRUS 43 C
TRUS 44 C CONTROL INFORMATION AND MEMBER PROPERTIES
TRUS 45 C
TRUS 46 NUME=NPAR(2)
TRUS 47 NUMMAT=NPAR(3)
TRUS 48 WRITE (6,2000) NUME,NUMMAT
TRUS 49 WRITE (6,2001)
TRUS 50 DO 10 I=1,NUMMAT
TRUS 51 READ (5,1001) N,E(IN),THERM(IN),DEN(IN),AREA(IN),WT(IN)
TRUS 52 10 WRITE (6,2002) N,E(IN),THERM(IN),DEN(IN),AREA(IN),WT(IN)
TRUS 53 C
TRUS 54 C ELEMENT LOAD MULTIPLIERS
TRUS 55 C
TRUS 56 READ (5,1003) EMUL
TRUS 57 WRITE (6,2003) EMUL
TRUS 58 C
TRUS 59 C ELEMENT INFORMATION
TRUS 60 WRITE (6,2005)
TRUS 61 C
TRUS 62 N=1
TRUS 63 100 READ (5,1004) M,II,JJ,MTYP,TEM,KK
TRUS 64 IF(KK.EQ.0) KK=1
TRUS 65 120 IF(M.NE.N) GO TO 200
TRUS 66 I=II
TRUS 67 J=JJ
TRUS 68 MTYPE=MTYP

```

```

TRUS 69 REFT=TEM
TRUS 70 KKK=KK
TRUS 71 C
TRUS 72 C 1. FORM ELEMENT STIFFNESS AND STRESS MATRICES
TRUS 73 C
TRUS 74 200 DX=X(1)-X(J)
TRUS 75 DY=Y(1)-Y(J)
TRUS 76 DZ=Z(1)-Z(J)
TRUS 77 XL2=DX*DX+DY*DY+DZ*DZ
TRUS 78 XL=SQR(XL2)
TRUS 79 XX=E(MTYPE)*AREA(MTYPE)*XL
TRUS 80 ST(1,1)=DX/XL2
TRUS 81 ST(1,2)=DY/XL2
TRUS 82 ST(1,3)=DZ/XL2
TRUS 83 ST(1,4)=-ST(1,1)
TRUS 84 ST(1,5)=-ST(1,2)
TRUS 85 ST(1,6)=-ST(1,3)
TRUS 86 C
TRUS 87 DO 300 L=1,6
TRUS 88 YY=ST(1,L)*KK
TRUS 89 DO 250 K=L,6
TRUS 90 S(K,L)=ST(1,K)*YY
TRUS 91 250 S(L,K)=S(K,L)
TRUS 92 ST(L,1)=E(MTYPE)*ST(1,L)
TRUS 93 300 ST(2,L)=AREA(MTYPE)*ST(1,L)
TRUS 94 C
TRUS 95 C 2. INERTIA AND THERMAL LOADS
TRUS 96 C
TRUS 97 F=WT(MTYPE)*XL/2.
TRUS 98 TEMP=(T(1)+T(J))*0.5 - REFT
TRUS 99 FT=TEMP*THERM(MTYPE)*E(MTYPE)*AREA(MTYPE)
TRUS 100 FX=DX*FT/XL
TRUS 101 FY=DY*FT/XL
TRUS 102 FZ=DZ*FT/XL
TRUS 103 C
TRUS 104 DO 350 L=1,4
TRUS 105 TT(2,L)=EMUL(L,4)*FT
TRUS 106 TT(1,L)=TT(2,L)/AREA(MTYPE)
TRUS 107 P(1,L)=EMUL(L,1)*F-EMUL(L,4)*FX
TRUS 108 P(2,L)=EMUL(L,2)*F-EMUL(L,4)*FY
TRUS 109 P(3,L)=EMUL(L,3)*F-EMUL(L,4)*FZ
TRUS 110 P(4,L)=EMUL(L,1)*F+EMUL(L,4)*FX
TRUS 111 P(5,L)=EMUL(L,2)*F+EMUL(L,4)*FY
TRUS 112 350 P(6,L)=EMUL(L,3)*F+EMUL(L,4)*FZ
TRUS 113 F=DEN(MTYPE)*AREA(MTYPE)*XL/2.
TRUS 114 DO 375 L=1,6
TRUS 115 375 XM(L)=F
TRUS 116 C
TRUS 117 C 3. FORM LOCATION MATRIX AND COMPUTE BAND WIDTH
TRUS 118 C
TRUS 119 DO 400 L=1,3
TRUS 120 LM(L)=ID(I,L)
TRUS 121 400 LM(L+3)=ID(J,L)
TRUS 122 C
TRUS 123 ND=6
TRUS 124 NS=2
TRUS 125 NDM=24
TRUS 126 CALL CALBAN (MBAND,NDIF,LM,XM,S,P,ND,NDM)
TRUS 127 WRITE (1) ND,NS,(LM(L),L=1,ND),(ST(L,K),L=1,NS),K=1,ND),
TRUS 128 I((TT(L,K),L=1,NS),K=1,4)
TRUS 129 C
TRUS 130 C 4. CHECK FOR MORE ELEMENTS
TRUS 131 C
TRUS 132 WRITE (6,2004) N,I,J,MTYPE,REFT,NDIF
TRUS 133 IF (N.EQ.NUME) RETURN
TRUS 134 N=N+1
TRUS 135 I=I+KKK
TRUS 136 J=J+KKK
TRUS 137 IF(N.GT.M) GO TO 100
TRUS 138 GO TO 120
TRUS 139 C
TRUS 140 1001 FORMAT (15,5F10.0)
TRUS 141 1003 FORMAT (4F10.0)
TRUS 142 1004 FORMAT (4I5,1F10.0,15)
TRUS 143 2000 FORMAT (//25H1NUMBER OF TRUSS MEMBERS= 15/

```

```
TRUS 144      1 25H NUMBER OF DIFF. MEMBERS= 15)
TRUS 145      2001 FORMAT (///1X,4HTYPE,14X,1HE,10X,5HALPHA,12X,3HDEN,11X,4HAREA
TRUS 146      1 11X,4HWT/L )
TRUS 147      2002 FORMAT (I5,5E15.7)
TRUS 148      2003 FORMAT(///25H ELEMENT LOAD MULTIPLIERS / 20X,1HA,14X,1HB,14X,1HC,
TRUS 149      1 14X,1HD,/6H X-DIR4E15.6/ 6H Y-DIR4E15.6/ 6H Z-DIR4E15.6/
TRUS 150      2 6H TEMP4E15.6)
TRUS 151      2004 FORMAT (4I6,F10.2,I7)
TRUS 152      2005 FORMAT (///42HI      N      I      J TYPE      TEMP      BAND )
TRUS 153      END
```

```

BEAM 1 SUBROUTINE BEAM
BEAM 2 C
BEAM 3 COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
BEAM 4 COMMON / JUNK / LT,LH,L,SIG(20)
BEAM 5 COMMON A(1)
BEAM 6 C
BEAM 7 IF(NPAR(1).EQ.0) GO TO 500
BEAM 8 N6=N5+NPAR(5) + NUMNP
BEAM 9 N7=N6+NPAR(5)
BEAM 10 N8=N7+NPAR(5)
BEAM 11 N9=N8+12*NPAR(4)
BEAM 12 N10=N9+6*NPAR(3)
BEAM 13 IF(N10.GT.MTOT) CALL ERROR(N10-MTOT)
BEAM 14 CALL TEAM(NPAR(2),NPAR(3),NPAR(4),NPAR(5),A(N1),A(N2),A(N3),
BEAM 15 I A(N4),A(N5),A(N6),A(N7),A(N8),A(N9),NUMNP,MBAND)
BEAM 16 RETURN
BEAM 17 C
BEAM 18 500 WRITE (6,2002)
BEAM 19 NUME=NPAR(2)
BEAM 20 DO 800 MM=1,NUME
BEAM 21 CALL STRSC (A(N1),A(N3),NEQ,0)
BEAM 22 WRITE (6,2001)
BEAM 23 DO 800 L=LT,LH
BEAM 24 CALL STRSC (A(N1),A(N3),NEQ,1)
BEAM 25 WRITE(6,3002) MM,L,(SIG(I),I=1,12)
BEAM 26 800 CONTINUE
BEAM 27 RETURN
BEAM 28 2001 FORMAT (//)
BEAM 29 2002 FORMAT(/29H0.....BEAM FORCES AND MOMENTS//
BEAM 30 . 10H0BEAM LOAD 5X 5HAXIAL 2(7X,5HSHEAR),5X 7HTORSION
BEAM 31 . 2(5X,7HBENDING)/ 10H NO. NO. 8X 2HR1 10X 2HR2 10X
BEAM 32 . 2HR3 10X 2HM1 10X 2HM2 10X 2HM3)
BEAM 33 3002 FORMAT (15,14,1PE11.3,5E12.3/8X,6E12.3/)
BEAM 34 END

```

```

BEAM 35 SUBROUTINE TEAM(NBEAM,NUMETP,NUMFIX,NUMMAT, ID,X,Y,Z,E,G,RO,
BEAM 36 ,SFT, COPROP,NUMNP,MBAND)
BEAM 37 C
BEAM 38 C FORMS 3-D BEAM STIFFNESS AND STRESS ARRAYS
BEAM 39 C
BEAM 40 COMMON/EM/LM(24),ND,NS,ASA(24,24),RF(24,4),XM(24),SA(12,24),
BEAM 41 SF(12,4)
BEAM 42 DIMENSION X(1),Y(1),Z(1),ID(NUMNP,1),E(1),G(1),SFT(NUMFIX,1)
BEAM 43 ,COPROP(NUMETP,1),RO(1),EMUL(3,4)
BEAM 44 COMMON /NEWB/ LC(4),T(3,3),JK(6),MELTYP,DL,MATTYP
BEAM 45 DIMENSION ILC(4),TI(3,3),TJ(3,3),STIF(722),TS(2,2),LS(4)
BEAM 46 EQUIVALENCE (STIF,LM)
BEAM 47 C
BEAM 48 C INITIALIZATICN
BEAM 49 C
BEAM 50 WRITE (6,2005) NBEAM,NUMETP,NUMFIX,NUMMAT
BEAM 51 N=0
BEAM 52 DO 5 I=1,1058
BEAM 53 5 STIF(I)=0.
BEAM 54 C
BEAM 55 C READ AND PRINT MATERIAL PROPERTY DATA
BEAM 56 C
BEAM 57 WRITE (6,2001)
BEAM 58 DO 10 I=1,NUMMAT
BEAM 59 READ (5,1001) N,E(N),G(N),RO(N)
BEAM 60 WRITE (6,2002) N,E(N),G(N),RO(N)
BEAM 61 10 G(N)=0.5*E(N)/(1.+G(N))
BEAM 62 C
BEAM 63 C READ AND PRINT GEOMETRIC PROPERTIES OF COMMON ELEMENTS.
BEAM 64 C
BEAM 65 WRITE (6,2003)
BEAM 66 DO 30 I=1,NUMETP
BEAM 67 READ (5,1002) N,(COPROP(N,J),J=1,6)
BEAM 68 IF((COPROP(N,1).NE.0.0).AND.(COPROP(N,4).NE.0.0).AND.

```

```

BEAM 69 1 (COPROP(N,5).NE.0.0).AND.(COPROP(N,6).NE.0.0) GO TO 20
BEAM 70 WRITE (6,2013)
BEAM 71 CALL EXIT
BEAM 72 20 WRITE (6,2004) N,(COPROP(N,J),J=1,6)
BEAM 73 30 CONTINUE
BEAM 74 C
BEAM 75 C ELEMENT LOAD MULTIPLIERS
BEAM 76 C
BEAM 77 READ (5,1006) ((EMUL(I,J),J=1,4),I=1,3)
BEAM 78 WRITE (6,2006) ((EMUL(I,J),J=1,4),I=1,3)
BEAM 79 C
BEAM 80 C READ AND PRINT FIXED END FORCES IN LOCAL COORDINATES
BEAM 81 C
BEAM 82 IF(NUMFIX .EQ. 0) GO TO 56
BEAM 83 WRITE (6,2010)
BEAM 84 DO 55 I=1,NUMFIX
BEAM 85 READ (5,1005) N,(SFT(N,J),J=1,12)
BEAM 86 55 WRITE (6,2011) N,(SFT(N,J),J=1,12)
BEAM 87 56 CONTINUE
BEAM 88 C
BEAM 89 C READ AND PRINT ELEMENT DATA. GENERATE MISSING INPUT.
BEAM 90 C
BEAM 91 WRITE (6,4000)
BEAM 92 L=0
BEAM 93 60 KKK=0
BEAM 94 READ (5,3000) INEL,INI,INJ,INK,IMAT,IMEL,ILC,INELKI,INELKJ,INC
BEAM 95 IF (INEL.NE.1) GO TO 15
BEAM 96 NI=INI
BEAM 97 NJ=INJ
BEAM 98 NK=INK
BEAM 99 15 IF (INC.EQ.0) INC=1
BEAM 100 65 L=L+1
BEAM 101 KKK=KKK+1
BEAM 102 ML=INEL-L
BEAM 103 IF (ML) 66,67,68
BEAM 104 66 WRITE (6,4003) INEL
BEAM 105 CALL EXIT
BEAM 106 67 NEL=INEL
BEAM 107 NI =INI
BEAM 108 NJ =INJ
BEAM 109 NK=INK
BEAM 110 MATTYP=IMAT
BEAM 111 MELTYP=IMEL
BEAM 112 DO 90 I=1,4
BEAM 113 90 LC(I)=ILC(I)
BEAM 114 NLOAD=LC(1)+LC(2)+LC(3)+LC(4)
BEAM 115 NEKODI=INELKI
BEAM 116 NEKODJ=INELKJ
BEAM 117 DO 91 I=1,3
BEAM 118 91 T(2,I)=TI(2,I)
BEAM 119 GO TO 69
BEAM 120 68 NEL=INEL-ML
BEAM 121 NI =IN+KKK*INCR
BEAM 122 NJ =JN+KKK*INCR
BEAM 123 69 CONTINUE
BEAM 124 WRITE (6,4001) NEL,NI,NJ,NK,MATTYP,MELTYP,LC,NEKODI,NEKODJ
BEAM 125 C
BEAM 126 74 DX=X(NJ)-X(NI)
BEAM 127 DY=Y(NJ)-Y(NI)
BEAM 128 DZ=Z(NJ)-Z(NI)
BEAM 129 DL=SQRT(DX*DX+DY*DY+DZ*DZ)
BEAM 130 IF(DL) 75,75,76
BEAM 131 75 WRITE (6,4005) NEL
BEAM 132 CALL EXIT
BEAM 133 C
BEAM 134 C FORM GLOBAL TO LOCAL COORDINATE TRANSFORMATION.
BEAM 135 C
BEAM 136 76 T(1,1)=DX/DL
BEAM 137 T(1,2)=DY/DL
BEAM 138 T(1,3)=DZ/DL
BEAM 139 C
BEAM 140 C COMPUTE DIRECTION COSINES OF LOCAL Y-AXIS
BEAM 141 C
BEAM 142 A1=X(NJ)-X(NI)
BEAM 143 A2=Y(NJ)-Y(NI)

```

```

BEAM 144      A3=Z(NJ)-Z(NI)
BEAM 145      B1=X(NK)-X(NI)
BEAM 146      B2=Y(NK)-Y(NI)
BEAM 147      B3=Z(NK)-Z(NI)
BEAM 148      AA=A1*A1+A2*A2+A3*A3
BEAM 149      AB=A1*B1+A2*B2+A3*B3
BEAM 150      U1=AA*B1-AB*A1
BEAM 151      U2=AA*B2-AB*A2
BEAM 152      U3=AA*B3-AB*A3
BEAM 153      UU=U1*U1+U2*U2+U3*U3
BEAM 154      UU=SQRT(UU)
BEAM 155      IF (UU.GT.0.) GO TO 40
BEAM 156      WRITE (6,4002) INEL
BEAM 157      STOP
BEAM 158      40 T(2,1)=U1/UU
BEAM 159      T(2,2)=U2/UU
BEAM 160      T(2,3)=U3/UU
BEAM 161      T(3,1)=T(1,2)*T(2,3)-T(1,3)*T(2,2)
BEAM 162      T(3,2)=T(1,3)*T(2,1)-T(1,1)*T(2,3)
BEAM 163      T(3,3)=T(1,1)*T(2,2)-T(1,2)*T(2,1)
BEAM 164      C
BEAM 165      C CHECK IF NEW STIFFNESS NEEDED
BEAM 166      C
BEAM 167      IF (NEL.GE.1) GO TO 80
BEAM 168      IF (ABS(CS-DL).GT.DL/100.) GO TO 80
BEAM 169      IF ((MT.NE.MATTYP).OR.(ME.NE.MELTYP)) GO TO 80
BEAM 170      IF ((JK(1).NE.NEKODI).OR.(JK(2).NE.NEKODJ)) GO TO 80
BEAM 171      DO 81 I=1,4
BEAM 172      IF (LS(I).NE.LC(I)) GO TO 80
BEAM 173      81 CONTINUE
BEAM 174      DO 82 I=1,2
BEAM 175      DO 82 J=1,2
BEAM 176      IF (ABS(TS(I,J))-T(I,J)).GT.ABS(T(I,J)/100.) GO TO 80
BEAM 177      82 CONTINUE
BEAM 178      GO TO 185
BEAM 179      C
BEAM 180      80 DS=DL
BEAM 181      MT=MATTYP
BEAM 182      ME=MELTYP
BEAM 183      DO 77 I=1,2
BEAM 184      DO 77 J=1,2
BEAM 185      77 TS(I,J)=T(I,J)
BEAM 186      DO 78 I=1,4
BEAM 187      78 LS(I)=LC(I)
BEAM 188      JK(1)=NEKODI
BEAM 189      JK(2)=NEKODJ
BEAM 190      C
BEAM 191      C FORM NEW STIFFNESS
BEAM 192      C
BEAM 193      CALL NEWBM (E,G,RO,COPROP,SFT,NUMFIX,NUMETP)
BEAM 194      C
BEAM 195      C ADD GRAVITY LOADING ... POINT LOADS ONLY COMPUTED
BEAM 196      C
BEAM 197      DO 180 I=1,3
BEAM 198      DO 180 J=1,4
BEAM 199      RF(I,J)=RF(I,J)+EMUL(I,J)*XM(I)
BEAM 200      180 RF(I+6,J)=RF(I+6,J)+EMUL(I,J)*XM(I+6)
BEAM 201      C
BEAM 202      C FORM ELEMENT LOCATION MATRIX
BEAM 203      C
BEAM 204      185 CONTINUE
BEAM 205      DO 170 M=1,6
BEAM 206      LM(M)=ID(NI,M)
BEAM 207      L4(M+12)=0
BEAM 208      LM(M+18)=0
BEAM 209      170 LM(M+6)=ID(NJ,M)
BEAM 210      C
BEAM 211      NS=12
BEAM 212      ND=12
BEAM 213      C
BEAM 214      C TRANSFORM TO MASTER DEGREES OF FREEDOM
BEAM 215      C
BEAM 216      CALL SLAVE (X,Y,Z,LD,NUMNP,NI,NJ)
BEAM 217      C
BEAM 218      C WRITE ELEMENT INFORMATION ON TAPE

```

```

BEAM 219      C
BEAM 220      NDM=24
BEAM 221      CALL CALBAN (MBAND,NDIF,LM,XM,ASA,RF,ND,NDM)
BEAM 222      WRITE (1) ND,NS,(LM(1),I=1,ND),(ISA(I,J),I=1,NS),J=1,ND)
BEAM 223      1 ((SF(I,J),I=1,NS),J=1,4)
BEAM 224      C
BEAM 225      C CHECK FOR LAST ELEMENT
BEAM 226      C
BEAM 227      IF (NBEAM-NEL) 66,500,260
BEAM 228      260 CONTINUE
BEAM 229      IF (ML.GT.0) GO TO 65
BEAM 230      IN =INI
BEAM 231      JN =INJ
BEAM 232      INCR=INC
BEAM 233      GO TO 60
BEAM 234      500 RETURN
BEAM 235      C
BEAM 236      1001 FORMAT(15,3F10.0)
BEAM 237      1002 FORMAT(15,6F10.0)
BEAM 238      1005 FORMAT(15,6F10.0/F15.0,5F10.0)
BEAM 239      1006 FORMAT(4F10.0)
BEAM 240      2001 FORMAT(55H MATERIAL YOUNG S POISSON S MASS
BEAM 241      1 /55H MODULUS RATIO DENSITY )
BEAM 242      2002 FORMAT(1H ,15,3X,F12.0,F14.5,F14.5)
BEAM 243      2003 FORMAT(1H/
BEAM 244      . 26H BEAM GEOMETRIC PROPERTIES//
BEAM 245      1 48H ELEMENT AREA AREA AREA
BEAM 246      2 60H INERTIA INERTIA INERTIA
BEAM 247      3 / 48H TYPE X Y Z
BEAM 248      4 30H X Y Z )
BEAM 249      2004 FORMAT(1H ,15,2X,6F12.3)
BEAM 250      2005 FORMAT(39H1.....THREE DIMENSIONAL BEAM ELEMENTS//
BEAM 251      . 36H NUMBER OF BEAMS =,15/
BEAM 252      . 36H NUMBER OF GEOMETRIC PROPERTY SETS=,15/
BEAM 253      . 36H NUMBER OF FIXED END FORCE SETS =,15/
BEAM 254      . 36H NUMBER OF MATERIALS =,15)
BEAM 255      2006 FORMAT(///25H ELEMENT LOAD MULTIPLIERS / 20X,1HA,14X,1HB,14X,1HC,
BEAM 256      1 14X,1HD,/6H X-DIR4E15.6/ 6H Y-DIR4E15.6/ 6H Z-DIR4E15.6/ )
BEAM 257      2010 FORMAT(1H,1H ,
BEAM 258      1 30X40H FIXED END FORCES IN LOCAL COORDINATES
BEAM 259      2//53H TYPE NODE FORCE X FORCE Y FORCE Z
BEAM 260      3 35H MOMENT X MOMENT Y MOMENT Z )
BEAM 261      2011 FORMAT(1H ,13,6X,1HI,3X,6F12.3/1H ,9X,1HJ,3X,6F12.3/)
BEAM 262      2013 FORMAT(1H0/
BEAM 263      1 60H SECTION PROPERTIES OTHER THAN SHEAR AREAS MAY NOT BE SPECIF
BEAM 264      2 34HIED AS ZERO. EXECUTION TERMINATED.)
BEAM 265      3000 FORMAT (1015,2I6,I8)
BEAM 266      4000 FORMAT(1H1/
BEAM 267      . 5H0BEAM 5X 5HNODES 5X 5H MATL 5H GEOM 5X 10HELEM LOADS 4X 10X
BEAM 268      . 12H END CODES / 5H NO 4X 1HI 4X 1HJ 4X 1HK 5H NO 5H NO
BEAM 269      . 4X 1HA 4X 1HB 4X 1HC 4X 1HD 9X 1HI 9X 1HJ)
BEAM 270      4001 FORMAT (1015,2I10)
BEAM 271      4002 FORM4T (9H0BEAM NO ,15, 26H K NODE ON BEAM X-AXIS ,
BEAM 272      . 26H.....EXECUTION TERMINATED )
BEAM 273      4003 FORMAT(36H0ELEMENT CARD ERROR, ELEMENT NUMBER= I6)
BEAM 274      4004 FORMAT(1H ,31HMODAL POINT NUMBERS FOR ELEMENT,15,36HARE IDENTICAL.
BEAM 275      1 EXECUTION TERMINATED.)
BEAM 276      4005 FORMAT(8H0ELEMENT,15,39H HAS ZERO LENGTH. EXECUTION TERMINATED.)
BEAM 277      END
BEAM 278      C
BEAM 279      C SUBROUTINE NEWBM(E,G,RO,COPROP,SFT,NUMFIX,NUMETP)
BEAM 280      C
BEAM 281      C FORM NEW BEAM STIFFNESS
BEAM 282      C
BEAM 283      DIMENSION E(1),G(1),RO(1),COPROP(NUMETP,1),SFT(NUMFIX,1)
BEAM 284      COMMON/EM/LM(24),ND,NS,ASA(24,24),RF(24,4),XM(24),SA(12,24),
BEAM 285      SF(12,4)
BEAM 286      COMMON /NEWB/ LC(4),T(3,3),JK(6),MELTYP,DL,MATTYP
BEAM 287      DIMENSION R(12),S(12,12),C(12)

```

```

BEAM 287 C
BEAM 288
BEAM 289 5 DD 5 I=1,144
BEAM 290 S(I)=0.
BEAM 291 AX=COPROP(MELTYP,1)
BEAM 292 AY=COPROP(MELTYP,2)
BEAM 293 AZ=COPROP(MELTYP,3)
BEAM 294 AAX=COPROP(MELTYP,4)
BEAM 295 AAY=COPROP(MELTYP,5)
BEAM 296 AAZ=COPROP(MELTYP,6)
BEAM 297 SHFY=0.0
BEAM 298 SHFZ=0.0
BEAM 299 ZY=E(MATTYP)/(DL*DL)
BEAM 300 EIZ=ZY*AAZ
BEAM 301 EIZ=ZY*AAZ
BEAM 302 IF(AY.NE.0.0) SHFY=6.*EIZ/(G(MATTYP)*AY)
BEAM 303 IF(AZ.NE.0.0) SHFZ=6.*EIZ/(G(MATTYP)*AZ)
BEAM 304 COMMY=EIZ/(1.+2.*SHFY)
BEAM 305 COMMZ=EIZ/(1.+2.*SHFZ)
BEAM 306 C
BEAM 307 C FIXED END FORCES IN LOCAL COORDS
BEAM 308 C
BEAM 309 DD 73 N=1,4
BEAM 310 M=LC(N)
BEAM 311 IF (M.GT.0) GO TO 71
BEAM 312 DO 70 I=1,12
BEAM 313 70 SF(I,N)=0.
BEAM 314 GO TO 73
BEAM 315 71 DO 72 I=1,12
BEAM 316 72 SF(I,N)=SFT(M,I)
BEAM 317 73 CONTINUE
BEAM 318 C
BEAM 319 C FORM ELEMENT STIFFNESS IN LOCAL COORDINATES
BEAM 320 C
BEAM 321 S(1,1)= E(MATTYP)* AX/DL
BEAM 322 S(4,4)= G(MATTYP)*AAX/DL
BEAM 323 S(2,2)= COMMY*12./DL
BEAM 324 S(3,3)= COMMZ*12./DL
BEAM 325 S(5,5)= COMMY* 4.*DL*(1.+0.5*SHFZ)
BEAM 326 S(6,6)= COMMZ* 4.*DL*(1.+0.5*SHFZ)
BEAM 327 S(2,6)= COMMY* 6.
BEAM 328 S(3,5)=-COMMZ* 6.
BEAM 329 DO 102 I=1,6
BEAM 330 J=I+6
BEAM 331 S(J,J)=S(I,I)
BEAM 332 DO 104 I=1,4
BEAM 333 J=I+6
BEAM 334 104 S(I,J)=-S(I,I)
BEAM 335 S(6,12)= S(6,6)*(1.-SHFY)/(2.+SHFY)
BEAM 336 S(5,11)= S(5,5)*(1.-SHFZ)/(2.+SHFZ)
BEAM 337 S(2,12)= S(2,6)
BEAM 338 S(6, 8)=-S(2,6)
BEAM 339 S(8,12)=-S(2,6)
BEAM 340 S(3,11)= S(3,5)
BEAM 341 S(9, 9)=-S(3,5)
BEAM 342 S(9,11)=-S(3,5)
BEAM 343 DO 106 I=2,12
BEAM 344 K=I-1
BEAM 345 DO 106 J=1,K
BEAM 346 106 S(I,J)=S(J,I)
BEAM 347 C
BEAM 348 C MODIFY ELEMENT STIFFNESS AND ELEMENT FIXED END FORCES FOR KNOWN
BEAM 349 C ZERO MEMBER END FORCES.
BEAM 350 C
BEAM 351 IF ((JK(1)+JK(2)).EQ.0) GO TO 145
BEAM 352 DD 140 K=1,2
BEAM 353 KK=JK(K)
BEAM 354 KD=100000
BEAM 355 I1=6*(K-1)+1
BEAM 356 I2=I1+5
BEAM 357 DO 140 I=11,12
BEAM 358 IF (KK.LT.KD) GO TO 140
BEAM 359 S(I,I)=1
BEAM 360 DD 125 N=1,12
BEAM 361 125 R(N)=S(I,N)
DD 130 M=1,12

```

```

BEAM 362 C(M)=S(M,1)/S11
BEAM 363 DD 130 N=1,12
BEAM 364 130 S(M,N)=S(M,N)-C(M)*R(N)
BEAM 365 DO 135 N=1,4
BEAM 366 SFI=SF(I,N)
BEAM 367 DO 135 M=1,12
BEAM 368 135 SF(M,N)=SF(M,N)-C(M)*SFI
BEAM 369 136 KK=KK-KC
BEAM 370 140 KD=KD/10
BEAM 371 145 CONTINUE
BEAM 372 C
BEAM 373 C OBTAIN SA(12,12) RELATING ELEMENT END FORCES (LOCAL) AND
BEAM 374 C JOINT DISPLACEMENTS (GLOBAL).
BEAM 375 C
BEAM 376 DO 31 I=1,288
BEAM 377 31 SA(I)=0.
BEAM 378 DO 150 LA=1,10,3
BEAM 379 LB=LA+2
BEAM 380 DO 150 MA=1,10,3
BEAM 381 MB=MA-1
BEAM 382 DO 150 I=LA,LB
BEAM 383 DO 150 JM=1,3
BEAM 384 J=JM+MB
BEAM 385 XX=0.
BEAM 386 DO 151 K=1,3
BEAM 387 151 XX=XX+S(I,K+MB)*T(K,JM)
BEAM 388 150 SA(I,J)=XX
BEAM 389 C
BEAM 390 C ELEM STIFF ASA(12,12) AND FIXED END FORCES RF(12) IN GLOBAL COORDS
BEAM 391 C
BEAM 392 DO 32 I=1,576
BEAM 393 32 ASA(I)=0.
BEAM 394 DO 160 LA=1,10,3
BEAM 395 LB=LA-1
BEAM 396 DO 160 MA=1,10,3
BEAM 397 MB=MA+2
BEAM 398 DO 160 IL=1,3
BEAM 399 I=IL+LB
BEAM 400 DO 160 J=MA,MB
BEAM 401 XX=0.
BEAM 402 DO 161 K=1,3
BEAM 403 161 XX=XX+T(K,IL)*SA(K+LB,J)
BEAM 404 160 ASA(I,J)=XX
BEAM 405 C
BEAM 406 DO 165 LA=1,10,3
BEAM 407 LB=LA-1
BEAM 408 DO 165 IL=1,3
BEAM 409 I=IL+LB
BEAM 410 DO 165 N=1,4
BEAM 411 XX=0.
BEAM 412 DO 162 K=1,3
BEAM 413 162 XX=XX-T(K,IL)*SF(K+LB,N)
BEAM 414 165 RF(I,N)=XX
BEAM 415 C
BEAM 416 C FORM MASS MATRIX
BEAM 417 C
BEAM 418 XXM=RO(MATTYP)*AX*DL/2.
BEAM 419 DO 180 M=1,3
BEAM 420 XM(M)=XXM
BEAM 421 XM(M+3)=0.
BEAM 422 XM(M+9)=0.
BEAM 423 180 XM(M+6)=XXM
BEAM 424 RETURN
BEAM 425 END

```

NOT REPRODUCIBLE

```

BEAM 426      SUBROUTINE SLAVE (X,Y,Z, ID, NUMNP, NI, NJ)
BEAM 427      C
BEAM 428      C PERFORMS SLAVE...MASTER DISPLACEMENT TRANSFORMATION
BEAM 429      C ( FOR NODES CONNECTED TO BEAM ELEMENTS ONLY)
BEAM 430      C
BEAM 431      DIMENSION X(1),Y(1),Z(1),ID(NUMNP,1)
BEAM 432      COMMON /EM/ LM(24),ND,NS,S(24,24),R(96),XM(24),SA(12,24),TT(12,4)
BEAM 433      C
BEAM 434      C DETERMINE REQUIRED TRANSLATION DEGREES OF FREEDOM
BEAM 435      C
BEAM 436      DO 54 NF=1,12,6
BEAM 437      NOD=NI
BEAM 438      IF (NF.EQ.7) NOD=NJ
BEAM 439      DO 30 K=1,3
BEAM 440      I=K+NF-1
BEAM 441      IF (LM(I).GE.0) GO TO 30
BEAM 442      M=-LM(I)
BEAM 443      LM(I)=ID(M,K)
BEAM 444      IF(K-2) 35,45,55
BEAM 445      35 D1=-(Y(NOD)-Y(M))
BEAM 446      D2= Z(NOC)-Z(M)
BEAM 447      LM(ND+1)=ID(M,6)
BEAM 448      LM(ND+2)=ID(M,5)
BEAM 449      GO TO 50
BEAM 450      45 D1=-(Z(NOC)-Z(M))
BEAM 451      D2= X(NOD)-X(M)
BEAM 452      LM(ND+1)=ID(M,4)
BEAM 453      LM(ND+2)=ID(M,6)
BEAM 454      GO TO 50
BEAM 455      55 D1=-(X(NOD)-X(M))
BEAM 456      D2= Y(NOC)-Y(M)
BEAM 457      LM(ND+1)=ID(M,5)
BEAM 458      LM(ND+2)=ID(M,4)
BEAM 459      50 CONTINUE
BEAM 460      C
BEAM 461      C TRANSFORMATION...ARRAYS INCREASE IN SIZE
BEAM 462      C
BEAM 463      DO 60 II=1,ND
BEAM 464      S(ND+1,II)=S(II,II)*D1
BEAM 465      S(ND+2,II)=S(II,II)*D2
BEAM 466      XM(ND+1)=XM(II)*D1*D1
BEAM 467      XM(ND+2)=XM(II)*D2*D2
BEAM 468      S(II,ND+1)=S(II,II)*D1
BEAM 469      S(II,ND+2)=S(II,II)*D2
BEAM 470      60 CONTINUE
BEAM 471      C
BEAM 472      DO 70 II=1,NS
BEAM 473      SA(II,ND+1)=SA(II,II)*D1
BEAM 474      70 SA(II,ND+2)=SA(II,II)*D2
BEAM 475      C
BEAM 476      S(ND+1,ND+1)=S(II,II)*D1**2
BEAM 477      S(ND+2,ND+2)=S(II,II)*D2**2
BEAM 478      S(ND+1,ND+2)=S(II,II)*D1*D2
BEAM 479      S(ND+2,ND+1)=S(ND+1,ND+2)
BEAM 480      ND=ND+2
BEAM 481      30 CONTINUE
BEAM 482      C
BEAM 483      C SET ROTATIONS
BEAM 484      C
BEAM 485      DO 54 J=1,3
BEAM 486      K=NF+J+2
BEAM 487      IF (LM(K).GE.0) GO TO 54
BEAM 488      M=-LM(K)
BEAM 489      LM(K)=ID(M,J+3)
BEAM 490      54 CONTINUE
BEAM 491      C
BEAM 492      RETURN
BEAM 493      END

```

```

PLAX 1 SUBROUTINE PLANE
PLAX 2 COMMON A(1)
PLAX 3 COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
PLAX 4 COMMON /EM/ NS,ND,LM(48),B(48,48),TI(48,4)
PLAX 5 COMMON /JUNK/ LT,LH,L,SG(20),SIG(7),EXTRA(150)
PLAX 6 DIMENSION STRLAB(5)
PLAX 7 DATA STRLAB/3HCEN,3HL-I,3HJ-K,3HI-J,3HK-L/
PLAX 8 IF(INPAR(1).EQ.0) GO TO 200
PLAX 9 IF(INPAR(1).EQ.3) NPAR(5)=2
PLAX 10 IF(INPAR(5).EQ.0) WRITE (6,2000)
PLAX 11 IF(INPAR(5).EQ.1) WRITE (6,2001)
PLAX 12 IF(INPAR(5).EQ.2) WRITE (6,2002)
PLAX 13 IF (NPAR(1).EQ.3) WRITE (6,2003)
PLAX 14 IF (NPAR(6).NE.0) WRITE (6,2004)
PLAX 15 IF(INPAR(4).EQ.0) NPAR(4)=1
PLAX 16 N6=N5+NUMNP
PLAX 17 N7=N6+NPAR(3)
PLAX 18 N8=N7+NPAR(3)
PLAX 19 N9=N8+NPAR(3)
PLAX 20 N10=N9+NPAR(3)
PLAX 21 MM=N10+11*NPAR(4)*NPAR(3)-MTOT
PLAX 22 IF(MM.GT.0) CALL ERROR(MM)
PLAX 23 CALL PLNAX(A(N1),A(N2),A(N3),A(N4),A(N5),A(N6),A(N7),A(N8),A(N9),
PLAX 24 IA(N10), NPAR(4),NUMNP)
PLAX 25 RETURN
PLAX 26 C
PLAX 27 200 NUME=NPAR(2)
PLAX 28 DO 800 MM=1,NUME
PLAX 29 CALL STRSC(A(N1),A(N3),NEQ,0)
PLAX 30 IF(NS.EQ.1) GO TO 800
PLAX 31 WRITE (6,3000) MM
PLAX 32 DO 700 L=LT,LH
PLAX 33 CALL STRSC(A(N1),A(N3),NEQ,1)
PLAX 34 ITAG=0
PLAX 35 510 DO 600 KK=1,NS,4
PLAX 36 ITAG=ITAG+1
PLAX 37 DO 520 I=1,4
PLAX 38 II=KK-1+I
PLAX 39 520 SIG(I)=SG(II)
PLAX 40 CC=(SIG(1)+SIG(2))/2.0
PLAX 41 BB=(SIG(1)-SIG(2))/2.
PLAX 42 CR=SQR(BB**2+SIG(4)**2)
PLAX 43 SIG(5)=CC+CR
PLAX 44 SIG(6)=CC-CR
PLAX 45 SIG(7)=0.0
PLAX 46 IF ((BB.EQ.0.0).AND.(SIG(4).EQ.0.0)) GO TO 600
PLAX 47 SIG(7)=28.648*ATAN2(SIG(4),BB)
PLAX 48 600 WRITE (6,3001) L,STRLAB(ITAG),SIG(I),I=1,7)
PLAX 49 700 CONTINUE
PLAX 50 800 CONTINUE
PLAX 51 RETURN
PLAX 52 2000 FORMAT (22H1AXISYMMETRIC ANALYSIS I)
PLAX 53 2001 FORMAT (22H1PLANE STRAIN ANALYSIS I)
PLAX 54 2002 FORMAT (22H1PLANE STRESS ANALYSIS I)
PLAX 55 2003 FORMAT (18H MEMBRANE ELEMENTS I)
PLAX 56 2004 FORMAT (30H INCOMPATIBLE MODES SUPPRESSED I)
PLAX 57 3000 FORMAT(1X,14FELEMENT NUMBER,15,5X,84HCENTER STRESSES IN LOCAL Y-Z
PLAX 58 . COORDS,BOUNDARY STRESSES NORMAL AND PARALLEL TO SIDES),
PLAX 59 . /1X,5H LOAD,17X,3HS11,12X,3HS22,12X,3HS33,12X,3HS12,10X,
PLAX 60 . 5HS-MAX,10X,5HS-MIN,5X,5HANGLE)
PLAX 61 3001 FORMAT(1X,15,2X,A3,1P6E15.6, 0P 1F10.3)
PLAX 62 END

PLAX 63 SUBROUTINE PLNAX(ID,X,Y,Z,T,NTC,WT,RO,WANG,E,NUMTC,NUMNP)
PLAX 64 DIMENSION X(1),Y(1),Z(1),ID(NUMNP,1),NTC(1),WT(1),RO(1),WANG(1),
PLAX 65 . E(NUMTC,11,1),T(1)
PLAX 66 COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
PLAX 67 COMMON /EM/ LM(12),S(12,12),P(12,4),XM(12),B(20,12),BB(20,12),
PLAX 68 1 TI(20,4),IX(4),IE(5),C(4,4),EMUL(4,5),RR(4),ZZ(4),HI(6),HS(6),

```

```

PLAX 69 2 HT(6),HR(6),HZ(6),FAC,XMM,PRESS,NS,EE(10),TTI(4),PP(12,4),THICK
PLAX 70 3 ,TMP(4),TP(12),ALP(4)
PLAX 71 COMMON /JUNK/ MAT,NT,TEMP,REFT,BETA,U(4),V(4),W(4),G(4)
PLAX 72 C
PLAX 73 NUME=NPAR(2)
PLAX 74 NUMMAT=NPAR(3)
PLAX 75 WRITE (6,2000) NUME,NUMMAT
PLAX 76 C
PLAX 77 C READ AND PRINT OF MATERIAL PROPERTIES
PLAX 78 DO 60 M=1,NUMMAT
PLAX 79 READ (5,1010) MAT,NTC(MAT),NT(MAT),RO(MAT),WANG(MAT)
PLAX 80 IF (NTC(MAT).EQ.0) NTC(MAT)=1
PLAX 81 WRITE (6,2020) MAT,NTC(MAT),NT(MAT),RO(MAT),WANG(MAT)
PLAX 82 NT=NTC(MAT)
PLAX 83 READ (5,1005) ((E(I,J,MAT),J=1,11),I=1,NT)
PLAX 84 WRITE (6,2010) ((E(I,J,MAT),J=1,11),I=1,NT)
PLAX 85 IF ( NPAR(5) .NE. 2 ) GO TO 60
PLAX 86 DO 50 I=1,NT
PLAX 87 E(I,4,MAT)=1.0
PLAX 88 E I, 6,MAT)=0.0
PLAX 89 E I, 7,MAT)=0.0
PLAX 90 50 E(I,11,MAT)=0.0
PLAX 91 60 CONTINUE
PLAX 92 READ (5,1002) ((EMUL(I,J),J=1,5),I=1,4)
PLAX 93 WRITE (6,2004) ((EMUL(I,J),J=1,5),I=1,4)
PLAX 94 C
PLAX 95 C READ AND PRINT OF ELEMENT PROPERTIES
PLAX 96 C
PLAX 97 WRITE (6,2002)
PLAX 98 N=0
PLAX 99 130 READ(5,1003) M,(IE(I),I=1,5),REFT,PRESS,NS,KG,THICK
PLAX 100 MAT=IE(5)
PLAX 101 IF(KG.EQ.0) KG=1
PLAX 102 IF (NPAR(5).EQ.1) THICK=1.0
PLAX 103 IF(NS.EQ.0) NS=4
PLAX 104 IF(NS.LT.4) NS=1
PLAX 105 IF( (IE(3) .EQ. IE(4)) .AND. (NS.EQ. 20) ) NS=16
PLAX 106 140 N=N+1
PLAX 107 IF(M.EQ.N) GO TO 145
PLAX 108 DO 142 I=1,4
PLAX 109 142 IX(I)=IX(I)+KG
PLAX 110 GO TO 149
PLAX 111 145 DO 148 I=1,4
PLAX 112 148 IX(I)=IE(I)
PLAX 113 C
PLAX 114 C FORM CONSTITUTIVE LAW AND COMPUTE THERMAL STRESSES
PLAX 115 C
PLAX 116 149 NT=NTC(MAT)
PLAX 117 WRITE (6,2003) N,IX,MAT,REFT,PRESS,NS,KG,THICK
PLAX 118 I=IX(1)
PLAX 119 J=IX(2)
PLAX 120 K=IX(3)
PLAX 121 L=IX(4)
PLAX 122 TEMP = (T(I)+T(J)+T(K)+T(L))/4.0
PLAX 123 BETA=WANG(MAT)
PLAX 124 XMM=RO(MAT)
PLAX 125 WGT=WT(MAT)
PLAX 126 CALL ELAW (NUMTC,EE,E,D,TTI,ALP)
PLAX 127 C
PLAX 128 C CALCULATE ELEMENT STIFFNESS MATRIX
PLAX 129 C
PLAX 130 IF(NPAR(1).EQ.3) GO TO 160
PLAX 131 ND=8
PLAX 132 DO 155 I=1,4
PLAX 133 II=IX(I)
PLAX 134 RR(I)=Y(II)
PLAX 135 ZZ(I)=Z(II)
PLAX 136 THP(I)=T(II)
PLAX 137 LM(I)=ID(II,2)
PLAX 138 155 LM(I+4)=ID(II,3)
PLAX 139 CALL QUAD
PLAX 140 DO 158 I=1,4
PLAX 141 DO 157 L=1,4
PLAX 142 P(I,L)=P(I,L)+XMM(I)*WGT*EMUL(L,4)
PLAX 143 157 P(I+4,L)=P(I+4,L)+XMM(I)*WGT*EMUL(L,5)

```

```

PLAX 144      XM(I)=XM(I)*XXM
PLAX 145      158 XM(I+4)=XM(I)
PLAX 146      GO TO 300
PLAX 147      C
PLAX 148      160 ND=12
PLAX 149      CALL VECTOR(V,X(I),Y(I),Z(I),X(J),Y(J),Z(J))
PLAX 150      CALL VECTOR(G,X(I),Y(I),Z(I),X(L),Y(L),Z(L))
PLAX 151      CALL CROSS(V,G,W)
PLAX 152      CALL CROSS(W,V,U)
PLAX 153      CALL VECTOR(W,X(I),Y(I),Z(I),X(K),Y(K),Z(K))
PLAX 154      RR(1)=0.0
PLAX 155      ZZ(1)=0.0
PLAX 156      RR(2)=V(4)
PLAX 157      ZZ(2)=0.0
PLAX 158      RR(3)=W(4)*DDOT(W,V)
PLAX 159      ZZ(3)=W(4)*DDOT(W,U)
PLAX 160      RR(4)=G(4)*DDOT(G,V)
PLAX 161      ZZ(4)=G(4)*DDOT(G,U)
PLAX 162      C
PLAX 163      DO 170 I=1,4
PLAX 164      II=IX(I)
PLAX 165      TMP(I)=T(II)
PLAX 166      LM(I)=ID(II,1)
PLAX 167      LM(I+4)=ID(II,2)
PLAX 168      170 LM(I+8)=ID(II,3)
PLAX 169      CALL QUAD
PLAX 170      C
PLAX 171      DO 190 I=1,3
PLAX 172      DO 190 K=1,4
PLAX 173      KK=4*(I-1)+K
PLAX 174      DO 180 L=1,4
PLAX 175      180 PP(KK,LL)=V(I)*P(K,L)+U(I)*P(K+4,L)
PLAX 176      DO 190 J=1,3
PLAX 177      DO 190 L=1,4
PLAX 178      LL=4*(J-1)+L
PLAX 179      190 BB(KK,LL)=V(I)*S(K,L)*V(J)+S(K,L+4)*U(J)
PLAX 180      1+U(I)*S(K+4,L)*V(J)+S(K+4,L+4)*U(J)
PLAX 181      C
PLAX 182      DO 196 I=1,12
PLAX 183      DO 194 L=1,4
PLAX 184      194 P(I,L)=PP(I,L)
PLAX 185      DO 196 J=1,12
PLAX 186      S(I,J)=BB(I,J)
PLAX 187      196 S(J,I)=S(I,J)
PLAX 188      C
PLAX 189      DO 210 K=1,NS
PLAX 190      DO 200 L=1,4
PLAX 191      DO 200 J=1,3
PLAX 192      LL=4*(J-1)+L
PLAX 193      200 BB(K,LL)=B(K,L)*V(J)+B(K,L+4)*U(J)
PLAX 194      DO 210 J=1,12
PLAX 195      210 B(K,J)=BB(K,J)
PLAX 196      C
PLAX 197      DO 220 I=1,4
PLAX 198      DO 215 L=1,4
PLAX 199      P(I,L)=P(I,L)+XM(I)*WGT*EMUL(L,3)
PLAX 200      P(I+4,L)=P(I+4,L)+XM(I)*WGT*EMUL(L,4)
PLAX 201      215 P(I+8,L)=P(I+8,L)+XM(I)*WGT*EMUL(L,5)
PLAX 202      XM(I)=XM(I)*XXM
PLAX 203      XM(I+4)=XM(I)
PLAX 204      220 XM(I+8)=XM(I)
PLAX 205      C
PLAX 206      C      CALCULATION OF BAND WIDTH AND WRITES ELEMENT MATRICES ON TAPES
PLAX 207      C
PLAX 208      300 CALL CALBAN(MBAND,NDIF,LM,XM,S,P,ND,12)
PLAX 209      WRITE (1) ND,NS,(LM(I),I=1,ND),((B(I),J),I=1,NS),J=1,ND),
PLAX 210      1 ((T(I),J),I=1,NS),J=1,4)
PLAX 211      IF(N.EQ.NUMF) RETURN
PLAX 212      IF(N.EQ.M) GO TO 130
PLAX 213      GO TO 140
PLAX 214      C
PLAX 215      1002 FORMAT (5F10.0)
PLAX 216      1003 FORMAT (6I5,2F10.0,2I5,F10.0)
PLAX 217      1005 FORMAT (8F10.0/3F10.0)
PLAX 218      1010 FORMAT (2I5,3F10.0)
PLAX 219      2000 FORMAT (34HNUMBER OF ELEMENTS = I/
PLAX 220      1 34HNUMBER OF MATERIALS = I5)
PLAX 221      2002 FORMAT (//50H EL.NO. I J K L TYPE TEMPERATURE
PLAX 222      1 ,7X,8HPRESSURE ,14H NO. STRESSES 20H KG THICKNESS )
PLAX 223      2003 FORMAT (110,5I5,2E15.6,2I10,E15.5)
PLAX 224      2004 FORMAT(23H2ELEMENT LOAD FRACTIONS /70H LOAD CASE TEMPERATURE PRES
PLAX 225      1SURE X-DIRECTION Y-DIRECTION Z-DIRECTION /9X IHA 5F12.3/
PLAX 226      2 9X IHB 5F12.3/ 9X IHC 5F12.3/ 9X IHD 5F12.3)
PLAX 227      2010 FORMAT (15H0 TEMPERATURE 11X 4HE(N),11X,4HE(S),11X,4HE(T),
PLAX 228      1 9X,6HNU(NS),9X,6HNU(NT), 9X,6HNU(ST),10X,5HMG(NS)/26X,4HA(N),
PLAX 229      2 11X,4HA(S),11X,4HA(T),
PLAX 230      3 (/F15.2,3F15.2,3F15.4,F15.0/22X,3F15.9))
PLAX 231      2020 FORMAT (///16H MATERIAL NUMBER I3//
PLAX 232      1 30H NUMBER OF TEMPERATURE CARDS =I3,5X,15H WGT. DENSITY =E12.4,
PLAX 233      2 15H MASS DENSITY =E12.4,13H ANGLE BETA =F6.1)
PLAX 234      END
PLAX 235      SUBROUTINE QUAD
PLAX 236      COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
PLAX 237      COMMON /EM/ LM(12),S(12,12),P(12,4),XM(12),B(20,12),BB(20,12),
PLAX 238      1 T(120,4),IX(4),TE(5),D(4,4),EMUL(4,5),RR(4),ZZ(4),H(6),H5(6),
PLAX 239      2 HT(6),HR(6),HZ(6),FAC,XXM,PRESS,NS,EE(10),TT(4),PP(12,4),THICK
PLAX 240      3 ,TMP(4),TP(12),ALP(4)
PLAX 241      COMMON /JUNK/ MAT,NT,TEMP,REFT,BETA
PLAX 242      DIMENSION SS(2),TT(2),HH(2),SSS(5),TTT(5),IVECT(4),JVECT(4),V(4)
PLAX 243      DATA SSS/0.,-1.,1.,0.,0./, TTT/0.,0.,0.,-1.,1./
PLAX 244      DATA SS/-0.57735026918963,0.57735026918963/
PLAX 245      DATA TT/0.57735026918963,0.57735026918963/
PLAX 246      DATA HH/1.0,1.0/,IVECT/4,2,1,3/,JVECT/1,3,2,4/
PLAX 247      C
PLAX 248      DO 170 J=1,12
PLAX 249      XM(J)=0.0
PLAX 250      TP(J)=0.0
PLAX 251      DO 160 I=1,20
PLAX 252      BB(I,J)=0.0
PLAX 253      160 B(I,J)=0.0
PLAX 254      DO 170 I=1,12
PLAX 255      170 S(I,J)=0.0
PLAX 256      C
PLAX 257      DO 500 II=1,2
PLAX 258      DO 500 JJ=1,2
PLAX 259      CALL FORMB(SS(II),SS(JJ),B)
PLAX 260      TEMP=0.
PLAX 261      DO 200 I=1,4
PLAX 262      200 TEMP=TEMP+H(I)*TMP(I)
PLAX 263      FAC=FAC*HH(JJ)*HH(II)
PLAX 264      FTP=TEMP-REFT
PLAX 265      DO 400 J=1,12
PLAX 266      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))*FAC
PLAX 267      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))*FAC
PLAX 268      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))*FAC
PLAX 269      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))*FAC
PLAX 270      TP(J)=TP(J)+FTP*(D1*ALP(1)+D2*ALP(2)+D3*ALP(3)+D4*ALP(4))
PLAX 271      DO 400 I=J,12
PLAX 272      S(I,J)=S(I,J)+B(1,I)*D1+B(2,I)*D2+B(3,I)*D3+B(4,I)*D4
PLAX 273      400 S(J,I)=S(I,J)
PLAX 274      DO 450 I=1,4
PLAX 275      450 XM(I)=XM(I)+FAC*H(I)
PLAX 276      500 CONTINUE
PLAX 277      C
PLAX 278      C      FORM STRESS DISPLACEMENT MATRIX
PLAX 279      C
PLAX 280      LL=NS/4
PLAX 281      DO 530 L=1,LL
PLAX 282      CALL FORMB(SSS(L),TTT(L),BB)
PLAX 283      C
PLAX 284      TEMP=0.
PLAX 285      DO 515 K=1,4
PLAX 286      515 TEMP=TEMP+H(K)*TMP(K)

```

```

PLAX 287      FAC=TEMP-REFT
PLAX 288      DO 530 I=1,4
PLAX 289      I=I+4*(L-1)
PLAX 290      TI(I,4)=-TTI(II)*FAC
PLAX 291      DO 530 J=1,12
PLAX 292      B(I,J)=0.0
PLAX 293      DO 530 K=1,4
PLAX 294      530 B(I,J)=B(I,J)+D(II,K)*BB(K,J)
PLAX 295      C
PLAX 296      C   ELIMINATE EXTRA DEGREES OF FREEDOM
PLAX 297      C
PLAX 298      IF ( IX(3) .EQ. IX(4) ) GO TO 560
PLAX 299      IF(NPAR(6).NE.0) GO TO 560
PLAX 300      DO 550 NN=1,4
PLAX 301      L=12-NN
PLAX 302      K=L+1
PLAX 303      C=TP(K)/S(K,K)
PLAX 304      DO 535 J=1,NS
PLAX 305      535 TI(J,4)=TI(J,4)+C*B(J,K)
PLAX 306      DO 550 I=1,L
PLAX 307      C=S(I,K)/S(K,K)
PLAX 308      TP(I)=TP(I)-C*TP(K)
PLAX 309      DO 540 J=1,NS
PLAX 310      540 B(J,I)=B(J,I)-C*B(J,K)
PLAX 311      DO 550 J=1,L
PLAX 312      550 S(I,J)=S(I,J)-C*S(K,J)
PLAX 313      C
PLAX 314      C   ROTATE STRESS-DISPLACEMENT TRANSFORMATION TO GIVE STRESSES
PLAX 315      C   NORMAL AND PARALLEL TO SIDES - SIMILARLY ROTATE INITIAL STRESSES
PLAX 316      C
PLAX 317      560 NSET=LL-1
PLAX 318      IF ( NSFT .LE. 0 ) GO TO 730
PLAX 319      DO 720 L=1,NSET
PLAX 320      IV=IVECT(L)
PLAX 321      JV=JVECT(L)
PLAX 322      CALL VECTOR(IV,RR(IV),ZZ(IV),0.0,RR(JV),ZZ(JV),0.0)
PLAX 323      S2=V(1)*V(1)
PLAX 324      C2=V(2)*V(2)
PLAX 325      SC=-V(1)*V(2)
PLAX 326      I1=4*L+1
PLAX 327      I2=I1+1
PLAX 328      I4=I1+3
PLAX 329      T1=TI(I1,4)
PLAX 330      T2=TI(I2,4)
PLAX 331      T4=TI(I4,4)
PLAX 332      T5=2.0*SC*T4
PLAX 333      TI(I1,4)=C2*T1+S2*T2+T5
PLAX 334      TI(I2,4)=S2*T1+C2*T2-T5
PLAX 335      TI(I4,4)=SC*(T2-T1)+(C2-S2)*T4
PLAX 336      DO 710 J=1,8
PLAX 337      B1=B(I1,J)
PLAX 338      B2=B(I2,J)
PLAX 339      B4=B(I4,J)
PLAX 340      B5=2.0*SC*B4
PLAX 341      B(I1,J)=C2*B1+S2*B2+B5
PLAX 342      B(I2,J)=S2*B1+C2*B2-B5
PLAX 343      710 B(I4,J)=SC*(B2-B1)+(C2-S2)*B4
PLAX 344      720 CONTINUE
PLAX 345      730 CONTINUE
PLAX 346      C
PLAX 347      DO 660 L=1,4
PLAX 348      DO 600 I=1,NS
PLAX 349      600 TI(I,L)=TI(I,4)*EMUL(L,1)
PLAX 350      DO 660 I=1,8
PLAX 351      660 P(I,L)=TP(I)*EMUL(L,1)
PLAX 352      C
PLAX 353      C   CALCULATE PRESSURE LOADS ON I-J FACE
PLAX 354      C
PLAX 355      DR=RR(2)-RR(1)
PLAX 356      DZ=ZZ(1)-ZZ(2)
PLAX 357      RI=PRESS*(2.*RR(1)+RR(2))/6.
PLAX 358      RJ=PRESS*(2.*RR(2)+RR(1))/6.
PLAX 359      IF(NPAR(5).EQ.0) GO TO 670
PLAX 360      RI=PRESS*THICK/2.
PLAX 361      RJ=RI

```

```

PLAX 362      670 DO 700 L=1,4
PLAX 363      P(L,L)=P(L,L)+DZ*RI*EMUL(L,2)
PLAX 364      P(5,L)=P(5,L)+DR*RJ*EMUL(L,2)
PLAX 365      P(2,L)=P(2,L)+DZ*RI*EMUL(L,2)
PLAX 366      700 P(6,L)=P(6,L)+DR*RJ*EMUL(L,2)
PLAX 367      RETURN
PLAX 368      END

```

```

PLAX 369      SUBROUTINE FORMB(S,T,B)
PLAX 370      COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,HTOT,NEQ
PLAX 371      COMMON /EM/ LM(12),U(12,12),P(12,4),XM(12),Q(20,12),BB(20,12),
PLAX 372      1 TI(20,4),IX(4),IB(5),DI(4,4),EMUL(4,5),RR(4),ZZ(4),H(6),HS(6),
PLAX 373      2 HT(6),HR(6),HZ(6),FAC,XMM,PRESS,NS,EE(10),TTI(4),PP(12,4),THICK
PLAX 374      3 ,TMP(4),QP(12),ALP(4)
PLAX 375      DIMENSION B(20,12)
PLAX 376      DIMENSION II(6),JJ(6)
PLAX 377      DATA II/1,2,3,4,9,10/,JJ/5,6,7,8,11,12/
PLAX 378      C
PLAX 379      SM=1.0-S
PLAX 380      SP=1.0+S
PLAX 381      TM=1.0-T
PLAX 382      TP=1.0+T
PLAX 383      C
PLAX 384      H(1)=SM*TM/4.
PLAX 385      H(2)=SP*TM/4.
PLAX 386      H(3)=SP*TP/4.
PLAX 387      H(4)=SM*TP/4.
PLAX 388      H(5)=(1.0-S)*S
PLAX 389      H(6)=(1.0-T)*T
PLAX 390      C
PLAX 391      HS(1)=-TM/4.
PLAX 392      HS(2)=-HS(1)
PLAX 393      HS(3)=TP/4.
PLAX 394      HS(4)=-HS(3)
PLAX 395      HS(5)=-2.*S
PLAX 396      HS(6)=0.0
PLAX 397      C
PLAX 398      HT(1)=-SM/4.
PLAX 399      HT(2)=-SP/4.
PLAX 400      HT(3)=-HT(2)
PLAX 401      HT(4)=-HT(1)
PLAX 402      HT(5)=0.0
PLAX 403      HT(6)=-2.*T
PLAX 404      C
PLAX 405      PZT=HT(1)*ZZ(1)+HT(2)*ZZ(2)+HT(3)*ZZ(3)+HT(4)*ZZ(4)
PLAX 406      PZS=HS(1)*ZZ(1)+HS(2)*ZZ(2)+HS(3)*ZZ(3)+HS(4)*ZZ(4)
PLAX 407      PRS=HS(1)*RR(1)+HS(2)*RR(2)+HS(3)*RR(3)+HS(4)*RR(4)
PLAX 408      PRT=HT(1)*RR(1)+HT(2)*RR(2)+HT(3)*RR(3)+HT(4)*RR(4)
PLAX 409      XJ=PRS*PZT-PRT*PZS
PLAX 410      C
PLAX 411      PSR=PZT/XJ
PLAX 412      PTR=-PZS/XJ
PLAX 413      PSZ=-PRT/XJ
PLAX 414      PTZ=PRS/XJ
PLAX 415      C
PLAX 416      DO 100 I=1,6
PLAX 417      HR(I)=PSR*HS(I)+PTR*HT(I)
PLAX 418      HZ(I)=PSZ*HS(I)+PTZ*HT(I)
PLAX 419      100 R=H(1)*RR(1)+H(2)*RR(2)+H(3)*RR(3)+H(4)*RR(4)
PLAX 420      IF(NPAR(5).NE.0) R=THICK
PLAX 421      C
PLAX 422      C   FORM STRAIN DISPLACEMENT MATRIX
PLAX 423      C
PLAX 424      DO 200 K=1,6
PLAX 425      I=II(K)
PLAX 426      J=JJ(K)
PLAX 427      B(I,I)=FR(K)
PLAX 428      B(2,J)=HZ(K)
PLAX 429      IF(NPAR(5).EQ.0) B(3,I)=H(K)/R

```

```

PLAX 430      B(4,I)=HZ(K)
PLAX 431      200 B(4,J)=HR(K)
PLAX 432      C
PLAX 433      FAC=XJ*RR
PLAX 434      RETURN
PLAX 435      END

PLAX 436      SUBROUTINE ELAW (NUMTC,EE,E,C,P,ALP)
PLAX 437      COMMON/JUNK/MAT,NT,TEMP,REFT,BETA,TAU(4),D(4,4),CC(4,4),XX(4)
PLAX 438      DIMENSION E(NUMTC,1,1),EE(10),C(4,4),P(4),ALP(4)
PLAX 439      C
PLAX 440      C      STRESS-STRAIN LAW IN N-S-T SYSTEM
PLAX 441      IF (NT.NE.1) GO TO 220
PLAX 442      DO 210 KK=1,10
PLAX 443      210 EE(KK)=E(1,KK+1,MAT)
PLAX 444      GO TO 260
PLAX 445      220 DO 230 I=2,NT
PLAX 446      T1=E(I-1,1,MAT)
PLAX 447      T2=E(I,1,MAT)
PLAX 448      IF(T2.GE.TEMP) GO TO 240
PLAX 449      230 CONTINUE
PLAX 450      240 CONTINUE
PLAX 451      RI=(T2-TEMP)/(T2-T1)
PLAX 452      RJ=(TEMP-T1)/(T2-T1)
PLAX 453      DO 250 KK=1,10
PLAX 454      250 EE(KK)=E(I-1,KK+1,MAT)*RI+E(I,KK+1,MAT)*RJ
PLAX 455      260 CONTINUE
PLAX 456      DO 265 II=1,4
PLAX 457      DO 265 KK=1,4
PLAX 458      C(II,KK)=0.
PLAX 459      265 D(II,KK)=0.
PLAX 460      C(1,1)=1.0/EE(1)
PLAX 461      C(1,2)=-EE(4)*C(1,1)
PLAX 462      C(1,3)=-EE(5)*C(1,1)
PLAX 463      C(2,1)=C(1,2)
PLAX 464      C(2,2)=1.0/EE(2)
PLAX 465      C(2,3)=-EE(6)*C(2,2)
PLAX 466      C(3,1)=C(1,3)
PLAX 467      C(3,2)=C(2,3)
PLAX 468      C(3,3)=1.0/EE(3)
PLAX 469      C(4,4)=EE(7)
PLAX 470      CALL PCSINV(C,3,4)
PLAX 471      C
PLAX 472      DO 270 M=1,3
PLAX 473      ALP(M)=EE(M+7)
PLAX 474      270 P(M)=C(M,1)*EE(8)+C(M,2)*EE(9)+C(M,3)*EE(10)
PLAX 475      ALP(4)=0.
PLAX 476      P(4)=0.0
PLAX 477      C
PLAX 478      C      ROTATE MATERIAL PROPERTIES TO R-Z-T SYSTEM
PLAX 479      C
PLAX 480      IF(BETA.EQ.0.0) GO TO 500
PLAX 481      ANG=BETA/57.2957795
PLAX 482      SS=SIN(ANG)
PLAX 483      CC=COS(ANG)
PLAX 484      S2=SS*SS
PLAX 485      C2=CC*CC
PLAX 486      SC=SS*CC
PLAX 487      O(1,1)=C2
PLAX 488      O(1,2)=S2
PLAX 489      O(1,4)=SC
PLAX 490      O(2,1)=S2
PLAX 491      O(2,2)=C2
PLAX 492      O(2,4)=-SC
PLAX 493      O(3,3)=1.0
PLAX 494      O(4,1)=-2.*SC
PLAX 495      O(4,2)=O(4,1)
PLAX 496      O(4,4)=C2-S2
PLAX 497      DO 287 JJ=1,4

```

```

PLAX 498      O1=C(1,1)*D(1,1,1,1)+C(1,2)*D(2,1,1,1)+C(1,3)*D(3,1,1,1)+C(1,4)*D(4,1,1,1)
PLAX 499      O2=C(2,1)*D(1,1,2,1)+C(2,2)*D(2,1,2,1)+C(2,3)*D(3,1,2,1)+C(2,4)*D(4,1,2,1)
PLAX 500      O3=C(3,1)*D(1,1,3,1)+C(3,2)*D(2,1,3,1)+C(3,3)*D(3,1,3,1)+C(3,4)*D(4,1,3,1)
PLAX 501      O4=C(4,1)*D(1,1,4,1)+C(4,2)*D(2,1,4,1)+C(4,3)*D(3,1,4,1)+C(4,4)*D(4,1,4,1)
PLAX 502      DO 287 II=JJ,4
PLAX 503      CC(II,II)=D(1,II)*O1+D(2,II)*O2+D(3,II)*O3+D(4,II)*O4
PLAX 504      287 CC(JJ,II)=CC(II,II)
PLAX 505      DO 290 II=1,4
PLAX 506      TAU(II)=0.0
PLAX 507      XX(II)=0.
PLAX 508      DO 290 KK=1,4
PLAX 509      XX(II)=XX(II)+D(KK,II)*ALP(KK)
PLAX 510      290 TAU(II)=TAU(II)+D(KK,II)*P(KK)
PLAX 511      C
PLAX 512      DO 300 I=1,4
PLAX 513      ALP(I)=XX(I)
PLAX 514      P(I)=TAU(I)
PLAX 515      DO 300 J=1,4
PLAX 516      300 C(I,J)=CC(I,J)
PLAX 517      C
PLAX 518      500 RETURN
PLAX 519      END

```

p-16

```

SOLB 1 SUBROUTINE THREEE
SOLB 2 C
SOLB 3 COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
SOLB 4 COMMON / EM / NS,VD,LM(48),B(48,48),TT(48,4)
SOLB 5 EQUIVALENCE (IS1,TT(4)), (IS2,TT(6))
SOLB 6 COMMON / JUNK / LT,LH,L,SIG(24)
SOLB 7 COMMON A(1)
SOLB 8 DIMENSION SPR(6)
SOLB 9 C
SOLB 10 IF(NPAR(1).EQ.0) GO TO 500
SOLB 11 N6=N5+NUMNP
SOLB 12 N7=N6+NP(3)
SOLB 13 N8=N7+NP(3)
SOLB 14 N9=N8+NP(3)
SOLB 15 N10=N9+NP(3)
SOLB 16 N11=N10+NP(4)
SOLB 17 N12=N11+NP(4)
SOLB 18 N13=N12+NP(4)
SOLB 19 N14=N13+NP(4)
SOLB 20 N15=N14+33*33
SOLB 21 N16=N15+12*33
SOLB 22 IF(N16.GT.MTOT) CALL ERROR (N16-MTOT)
SOLB 23 CALL BRICK8 (A(N14),A(N15),
SOLB 24 . NPAR(2),NPAR(3),NPAR(4),A(N1),A(N2),A(N3),A(N4),
SOLB 25 . A(N5),A(N6),A(N7),A(N8),A(N9),A(N10),A(N11),
SOLB 26 . A(N12),A(N13),NUMNP)
SOLB 27 RETURN
SOLB 28 C
SOLB 29 500 WRITE (6,2005)
SOLB 30 NUME=NP(2)
SOLB 31 DO 800 MM=1,NUME
SOLB 32 CALL STRC (A(N1),A(N3),NEQ,0)
SOLB 33 WRITE (6,2000)
SOLB 34 DO 800 L=LT,LH
SOLB 35 CALL STRC (A(N1),A(N3),NEQ,1)
SOLB 36 CALL PRST (NS,IS1,IS2,SIG,SPR)
SOLB 37 WRITE (6,3005) MM,L,IS1,(SIG(I),I=1,6),(SPR(I),I=1,3)
SOLB 38 IF(NS.EQ.12) WRITE (6,3015) IS2,(SIG(I),I=7,12),(SPR(I),I=4,6)
SOLB 39 800 CONTINUE
SOLB 40 C
SOLB 41 RETURN
SOLB 42 C
SOLB 43 2000 FORMAT (/)
SOLB 44 2005 FORMAT (36H .....8-NODE SOLID ELEMENT STRESSES //
SOLB 45 . 24H ELEMENT LOAD NO. FACE 5X,
SOLB 46 . 104H SIG-XX SIG-YY SIG-ZZ SIG-XY SIG-YZ
SOLB 47 . SIG-ZX SIG-MAX SIG-MIN S2/ANGLE)
SOLB 48 3005 FORMAT (16,I9,I8,2X,1P9E12.2)
SOLB 49 3015 FORMAT (15X, I8,2X,1P9E12.2)
SOLB 50 END

```

```

SOLB 51 SUBROUTINE BRICK8 (S,STR,NBRK8,NMAT,NLD,IX,Y,Z,T,EE,ENU,RHO,
SOLB 52 . ALPT,KTYPE,PR,YREF,NFACE,NUMNP)
SOLB 53 C
SOLB 54 STIFFNESS SUBROUTINE FOR 24 D.F. ISOPARAMETRIC HEXAHEDRON
SOLB 55 C LINEAR ELASTIC ISOTROPIC MATERIAL
SOLB 56 C *NINT*NINT*NINT* GAUSSIAN INTEGRATION RULE USED (NINT=1,2,3,4)
SOLB 57 C
SOLB 58 DIMENSION KTYPE(1),PR(1),YREF(1),NFACE(1)
SOLB 59 DIMENSION T(1)
SOLB 60 DIMENSION X(1),Y(1),Z(1),ID(NUMNP,6)
SOLB 61 COMMON/EM/LM(24),ND,NS, SS(24,24),RF(24,4),XM(24),SA(12,24),
SOLB 62 . SF(12,4)
SOLB 63 EQUIVALENCE (IS1,SF(4)), (IS2,SF(6))
SOLB 64 DIMENSION EE(1),ENU(1),RHO(1),ALPT(1)
SOLB 65 COMMON /GASS/ XK(4,4),WGT(4,4),IPERM(3)
SOLB 66 COMMON /JUNK/ E1,E2,E3,DET,MLC(4),KLDI(4),MULT(4),NP(8),INP(8),
SOLB 67 . A(3,3),P(11,3),E(3,3),XX(8,3),Q(11),DL(8),
SOLB 68 . TT(24),XLF(4),YLF(4),ZLF(4),TLF(4),PLF(4),

```

```

SOLB 69 . REFT,INEL,ININT,IPAT,IINC,TTEMP,NEL,ML,NINT,MAT,
SOLB 70 . INC,TAG,TEMP,SKIP,I,J,K,L,FAC,CC1,CC2,CC3,CC4,G,
SOLB 71 . DEN,FACT,GT,GG,C1,C2,C3,C,K1,K2
SOLB 72 COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
SOLB 73 DIMENSION S(33,33),STR(12,33)
SOLB 74 DIMENSION E(3,3)
SOLB 75 DIMENSION IS(2),ISP(2)
SOLB 76 C
SOLB 77 DATA XK / 0., 0., 0., 0., 0.,
SOLB 78 1 -.5773502691896, .5773502691896, 0., 0.,
SOLB 79 2 -.7745966692415, .0000000000000, .7745966692415,
SOLB 80 3 -.8611363115941,-.3399810435849, .3399810435849, .8611363115941/
SOLB 81 DATA WGT / 2.000, 0., 0., 0.,
SOLB 82 1 1.0000000000000,1.0000000000000, 0., 0.,
SOLB 83 2 .5555555555556, .8888888888889, .5555555555556,
SOLB 84 3 .3478548451375, .6521451548625, .6521451548625, .3478548451375/
SOLB 85 CATA IPERM / 2,3,1 /
SOLB 86 DIMENSION STPTS(7,3)
SOLB 87 DATA STPTS / 0., 1., -1., 0., 0., 0., 0.,
SOLB 88 . 0., 0., 0., 1., -1., 0., 0.,
SOLB 89 . 0., 0., 0., 0., 0., 1., -1. /
SOLB 90 C
SOLB 91 C
SOLB 92 C
SOLB 93 C ZERO EM
SOLB 94 C
SOLB 95 WRITE (6,3000) NBRK8,NMAT,NLD
SOLB 96 DO 9 I=1,1058
SOLB 97 9 LM(I)=0
SOLB 98 C
SOLB 99 C MATERIAL PROPERTIES
SOLB 100 C
SOLB 101 WRITE (6,1300)
SOLB 102 DO 1 I=1,NMAT
SOLB 103 READ (5,1001) N,EE(N),ENU(N),RHO(N),ALPT(N)
SOLB 104 1 WRITE (6,2001) N,EE(N),ENU(N),RHO(N),ALPT(N)
SOLB 105 C
SOLB 106 C ELEMENT DISTRIBUTED LOAD CARDS
SOLB 107 C
SOLB 108 IF(NLD) 23,23,15
SOLB 109 15 WRITE (6,1302)
SOLB 110 DO 16 I=1,NLD
SOLB 111 READ (5,1002) N,KTYPE(N),PR(N),YREF(N),NFACE(N)
SOLB 112 16 WRITE (6,2002) N,KTYPE(N),PR(N),YREF(N),NFACE(N)
SOLB 113 C
SOLB 114 23 READ (5,1003) GRAV,PLF,TLF,XLF,YLF,ZLF
SOLB 115 WRITE (6,2003) PLF,TLF,XLF,YLF,ZLF
SOLB 116 GRAV=1.0
SOLB 117 WRITE (6,1301)
SOLB 118 NEL=0
SOLB 119 30 READ (5,1000) INEL,INP,ININT,IPAT,IINC,MLD,ISP,TTEMP
SOLB 120 DO 39 I=1,4
SOLB 121 39 MULTI(I)=1
SOLB 122 IF(IINC.EQ.0) IINC=1
SOLB 123 IF(IPAT.EQ.0) IPAT=1
SOLB 124 40 NEL=NEL+1
SOLB 125 ML=INEL-NEL
SOLB 126 IF(ML) 50,55,60
SOLB 127 50 WRITE (6,4003) INEL
SOLB 128 STOP
SOLB 129 55 DO 56 I=1,8
SOLB 130 56 NP(I)=INP(I)
SOLB 131 NINT=ININT
SOLB 132 MAT=IPAT
SOLB 133 INC=IINC
SOLB 134 TAG=I*
SOLB 135 REPT=TTEMP
SOLB 136 IS(1)=ISP(1)
SOLB 137 IS(2)=ISP(2)
SOLB 138 SKIP=999999.
SOLB 139 IF(NINT) 33,33,57
SOLB 140 33 NINT=IABS(NINT)
SOLB 141 SKIP=1.
SOLB 142 IF(NINT.EQ.0) SKIP=0.
SOLB 143 57 CONTINUE

```

D-17

```

SOL8 144      DO 59 I=1,4
SOL8 145      KLD(I)=IABS(MLD(I))
SOL8 146      IF(MLD(I)) 58,58,59
SOL8 147      58 MULTI(I)=0
SOL8 148      59 CONTINUE
SOL8 149      GO TO 62
SOL8 150      C
SOL8 151      60 DO 61 I=1,8
SOL8 152      61 NP(I)=NP(I)+INC
SOL8 153      TAG=IH
SOL8 154      DO 64 I=1,4
SOL8 155      64 KLD(I)=KLD(I)*MULT(I)
SOL8 156      C
SOL8 157      62 TEMP=0.
SOL8 158      DO 10 I=1,8
SOL8 159      K=NP(I)
SOL8 160      TEMP=TEMP+T(K)
SOL8 161      XX(I,1)=X(K)
SOL8 162      XX(I,2)=Y(K)
SOL8 163      10 XX(I,3)=Z(K)
SOL8 164      TEMP=TEMP*.125
SOL8 165      K=MAT
SOL8 166      FAC = EE(K)/((1.-2.*ENU(K))*(1.+ENU(K)))
SOL8 167      FACT=FAC*ALPT(K)*(TEMP-REFT)*(1.+ENU(K))
SOL8 168      IF(SKIP) 70,70,63
SOL8 169      63 SKIP=SKIP-1.
SOL8 170      CC1=1.-ENU(K)
SOL8 171      CC2=ENU(K)
SOL8 172      CC3=.5-ENU(K)
SOL8 173      C
SOL8 174      L3=33*33
SOL8 175      DO 100 I=1,L3
SOL8 176      100 S(I)=0.
SOL8 177      DO 110 I=1,24
SOL8 178      110 TT(I)=0.
SOL8 179      DO 120 I=1,8
SOL8 180      120 DL(I)=0.
SOL8 181      C
SOL8 182      C LOOP OVER NINT**3 INTEGRATION POINTS
SOL8 183      C
SOL8 184      DO 300 LX = 1,NINT
SOL8 185      E1=KK(LX,NINT)
SOL8 186      DO 300 LY = 1,NINT
SOL8 187      E2=KK(LY,NINT)
SOL8 188      DO 300 LZ = 1,NINT
SOL8 189      E3=KK(LZ,NINT)
SOL8 190      C
SOL8 191      CALL DERIV(1,SA)
SOL8 192      C
SOL8 193      GT= WGT(LX,NINT)*WGT(LY,NINT)*WGT(LZ,NINT)*DET
SOL8 194      GG=GT*RHO(MAT)
SOL8 195      G=GT*FAC
SOL8 196      C1=G*CC1
SOL8 197      C2=G*CC2
SOL8 198      C3=G*CC3
SOL8 199      C
SOL8 200      L=0
SOL8 201      DO 310 I=1,8
SOL8 202      DL(I)=DL(I) + GG*Q(I)
SOL8 203      DO 310 K=1,3
SOL8 204      L=L+1
SOL8 205      310 TT(L)=TT(L) + GT*SA(I,K)
SOL8 206      C
SOL8 207      C ADD CONTRIBUTION TO STIFFNESS MATRIX
SOL8 208      C
SOL8 209      DO 300 I=1,11
SOL8 210      K3 = 3*I
SOL8 211      K2 = K3 - 1
SOL8 212      K1 = K2 - 1
SOL8 213      UJ=SA(I,1)
SOL8 214      VJ=SA(I,2)
SOL8 215      WJ=SA(I,3)
SOL8 216      DO 300 J=1,11
SOL8 217      L3 = 3*J
SOL8 218      L2 = L3 - 1

```

```

SOL8 219      L1 = L2 - 1
SOL8 220      UJ=SA(J,1)
SOL8 221      VJ=SA(J,2)
SOL8 222      WJ=SA(J,3)
SOL8 223      UJ=UJ*UJ
SOL8 224      VJ=VJ*VJ
SOL8 225      WJ=WJ*WJ
SOL8 226      UV=UJ*VJ
SOL8 227      VU=VJ*UJ
SOL8 228      UW=UJ*WJ
SOL8 229      WU=WJ*UJ
SOL8 230      VW=VJ*WJ
SOL8 231      WV=WJ*VJ
SOL8 232      S(K1,L1) = S(K1,L1) + C1*UU + C3*(VV+WV)
SOL8 233      S(K2,L2) = S(K2,L2) + C1*VV + C3*(WW+UV)
SOL8 234      S(K3,L3) = S(K3,L3) + C1*WW + C3*(UU+VW)
SOL8 235      S(K1,L2) = S(K1,L2) + C2*UV + C3*VU
SOL8 236      S(K1,L3) = S(K1,L3) + C2*UW + C3*WU
SOL8 237      S(K2,L3) = S(K2,L3) + C2*VW + C3*WV
SOL8 238      IF (.EQ.J) GO TO 300
SOL8 239      S(K2,L1) = S(K2,L1) + C2*VU + C3*UV
SOL8 240      S(K3,L1) = S(K3,L1) + C2*WU + C3*UW
SOL8 241      S(K3,L2) = S(K3,L2) + C2*WV + C3*VW
SOL8 242      300 CONTINUE
SOL8 243      C
SOL8 244      C FORM STRAIN MATRIX
SOL8 245      C
SOL8 246      NSS=2
SOL8 247      IF (IS(2).EQ.0) NSS=1
SOL8 248      DO 305 I=1,12
SOL8 249      DO 305 J=1,33
SOL8 250      305 STR(I,J)=0.
SOL8 251      DO 405 L=1,NSS
SOL8 252      LL=IS(L)+1
SOL8 253      E1=STPTS(LL,1)
SOL8 254      E2=STPTS(LL,2)
SOL8 255      E3=STPTS(LL,3)
SOL8 256      CALL DERIV(2,SA)
SOL8 257      L3=6*L-6
SOL8 258      DO 402 K=1,11
SOL8 259      K3=3*K
SOL8 260      K2=K3-1
SOL8 261      K1=K2-1
SOL8 262      STR(L3+1,K1) = SAIK,1)
SOL8 263      STR(L3+2,K2) = SAIK,2)
SOL8 264      STR(L3+3,K3) = SAIK,3)
SOL8 265      STR(L3+4,K1) = SAIK,2)
SOL8 266      STR(L3+4,K2) = SAIK,1)
SOL8 267      STR(L3+5,K2) = SAIK,3)
SOL8 268      STR(L3+5,K3) = SAIK,2)
SOL8 269      STR(L3+6,K1) = SAIK,3)
SOL8 270      402 STR(L3+6,K3) = SAIK,1)
SOL8 271      405 CONTINUE
SOL8 272      NS=6*NSS
SOL8 273      C
SOL8 274      C STATIC CONDENSATION
SOL8 275      C
SOL8 276      DO 710 M=1,9
SOL8 277      MN=34-M
SOL8 278      MD=MN-1
SOL8 279      C STIFFNESS MATRIX - S
SOL8 280      SP=S(MN,MN)
SOL8 281      DO 650 I=1,MO
SOL8 282      650 S(MN,I)=S(I,MN)/SP
SOL8 283      DO 700 K=1,MO
SOL8 284      SP=S(MN,K)
SOL8 285      DO 700 J=1,K
SOL8 286      700 S(J,K)=S(J,K) - SP*S(I,MN)
SOL8 287      C DERIVATIVE MATRIX - STR
SOL8 288      DO 710 J=1,NS
SOL8 289      SP=STR(J,MN)
SOL8 290      IF (SP.EQ.0.) GO TO 710
SOL8 291      DO 705 K=1,MO
SOL8 292      705 STR(J,K)=STR(J,K) - SP*S(MN,K)
SOL8 293      710 CONTINUE

```

```

SOL8 294 C
SOL8 295 DO 760 I=1,24
SOL8 296 DO 760 J=1,24
SOL8 297 SS(I,J)=S(I,J)
SOL8 298 760 SS(J,I)=SS(I,J)
SOL8 299 C
SOL8 300 C STRAIN TO STRESS MATRIX
SOL8 301 C
SOL8 302 E(1,1)=CC1*FAC
SOL8 303 E(2,2)=E(1,1)
SOL8 304 E(3,3)=E(1,1)
SOL8 305 E(1,2)=CC2*FAC
SOL8 306 E(1,3)=E(1,2)
SOL8 307 E(2,3)=E(1,2)
SOL8 308 E(2,1)=E(1,2)
SOL8 309 E(3,1)=E(1,2)
SOL8 310 E(3,2)=E(1,2)
SOL8 311 C
SOL8 312 DO 900 I=1,NSS
SOL8 313 II=I*6-6
SOL8 314 DO 850 J=1,3
SOL8 315 DO 850 K=1,24
SOL8 316 SP=0.
SOL8 317 DO 840 L=1,3
SOL8 318 840 SP = SP + E(J,L)*STR(II+L,K)
SOL8 319 SA(II+J,K)=SP
SOL8 320 JJ=II+3+J
SOL8 321 850 SA(JJ,K)=CC3*FAC*STR(JJ,K)
SOL8 322 C
SOL8 323 C
SOL8 324 DO 860 J=1,3
SOL8 325 JJ=J+3
SOL8 326 DO 860 K=1,4
SOL8 327 SF(II+J,K)=-FACT*TLF(K)
SOL8 328 860 SF(II+JJ,K)=0.
SOL8 329 C
SOL8 330 IF (IS(II).LE.0) GO TO 900
SOL8 331 LL=IS(II)+1
SOL8 332 E1=STPTS(LL,1)
SOL8 333 E2=STPTS(LL,2)
SOL8 334 E3=STPTS(LL,3)
SOL8 335 CALL DERIV (4,SA)
SOL8 336 CALL LOSTR (IS,A,B,SA,SF,I)
SOL8 337 C
SOL8 338 900 CONTINUE
SOL8 339 C
SOL8 340 70 CONTINUE
SOL8 341 C
SOL8 342 C DISTRIBUTEC LOAD
SOL8 343 C
SOL8 344 DO 410 J=1,24
SOL8 345 DO 410 I=1,4
SOL8 346 410 RF(I,J)=0.
SOL8 347 CALL LOAD (KTYPE,PR,YREF,NFACE)
SOL8 348 C
SOL8 349 C SELF WGT.
SOL8 350 C
SOL8 351 DO 460 II=1,8
SOL8 352 K=3*II
SOL8 353 J=K-1
SOL8 354 I=J-1
SOL8 355 DO 460 L=1,4
SOL8 356 RF(I,L) = RF(I,L)*PLF(L) + DL(II)*XLFL(L)
SOL8 357 RF(J,L) = RF(J,L)*PLF(L) + DL(II)*YLFL(L)
SOL8 358 460 RF(K,L) = RF(K,L)*PLF(L) + DL(II)*ZLFL(L)
SOL8 359 C
SOL8 360 C THERMAL LOADS
SOL8 361 C
SOL8 362 DO 470 I=1,24
SOL8 363 GT=TT(I)*FACT
SOL8 364 DO 470 J=1,4
SOL8 365 470 RF(I,J)=RF(I,J) + GT*TLF(J)
SOL8 366 C
SOL8 367 C MASS ARRAY
SOL8 368 C
SOL8 369 L=0
SOL8 370 DO 465 I=1,8
SOL8 371 DO 465 J=1,3
SOL8 372 L=L+1
SOL8 373 465 XM(L)=DL(I)/GRAV
SOL8 374 C
SOL8 375 IJ=0
SOL8 376 DO 550 I=1,8
SOL8 377 II=NP(I)
SOL8 378 DO 550 J=1,3
SOL8 379 IJ=IJ+1
SOL8 380 550 LM(IJ)=10(II,J)
SOL8 381 ND=24
SOL8 382 C
SOL8 383 IS1=IS(1)
SOL8 384 IS2=IS(2)
SOL8 385 NDM=24
SOL8 386 CALL CALBAN (MBAND,NDIF,LM,XM,SS,RF,ND,NDM)
SOL8 387 WRITE (1) ND,NS,(LM(I),I=1,ND),((SA(I,J),I=1,NS),J=1,ND),
SOL8 388 1 ((SF(I,J),I=1,NS),J=1,4)
SOL8 389 WRITE (6,2000) NEL,NP,NINT,MAT,TAG,KLD,REFT,IS,NDIF
SOL8 390 C
SOL8 391 C CHECK IF LAST ELEMENT
SOL8 392 C
SOL8 393 IF(NBRK8-NEL) 50,600,590
SOL8 394 590 IF(ML) 30,30,40
SOL8 395 C
SOL8 396 C
SOL8 397 600 RETURN
SOL8 398 C
SOL8 399 C
SOL8 400 1000 FORMAT (12I5,4I2,2I1,F10.2)
SOL8 401 1001 FORMAT (15,4F10.0)
SOL8 402 1002 FORMAT (2I5,2F10.2,15)
SOL8 403 1003 FORMAT(10X,F10.2/(4F10.2))
SOL8 404 2000 FORMAT (I6,X,8I5,I9,I12,8X,A1,3X,4I5,F9.1,5X,2I3,18)
SOL8 405 2001 FORMAT (X,I5,4E15.4)
SOL8 406 2002 FORMAT (I5,I9,2F13.3,I12)
SOL8 407 2003 FORMAT (////)
SOL8 408 . 30H LOAD FACTORS FOR 4 ELEMENT LOAD CASES //
SOL8 409 . 46X 1THELEMENT LOAD CASE /
SOL8 410 . 36X 1HA 9X 1HB 9X 1HC 9X 1HD /
SOL8 411 . 30H PRESSURE LOAD FACTORS . . 4F10.3/
SOL8 412 . 30H THERMAL LOAD FACTORS . . 4F10.3//
SOL8 413 . 30H PERCENT GRAVITY IN +X DIRN. 4F10.3/
SOL8 414 . 30H PERCENT GRAVITY IN +Y DIRN. 4F10.3/
SOL8 415 . 30H PERCENT GRAVITY IN +Z DIRN. 4F10.3/ )
SOL8 416 1300 FORMAT (9HOMATERIAL 10X 1HE 12X 2HNU 10X 3HRHO 11X THALPHA-T /
SOL8 417 . 8H NUMBER //)
SOL8 418 1301 FORMAT (30H1....8 NODE SOLID ELEMENT DATA ///
SOL8 419 . 8H ELEMENT 10X 15HCONNECTED NODES 17X 28HINTEGRATION MATERIAL I
SOL8 420 .INPUT 7X 13HELEMENT LOADS 5X 7HELEMENT ,5X,6HSTRESS /
SOL8 421 . 8H NUMBER 3X,36H1 2 3 4 5 6 7 8 6X,5HORDER
SOL8 422 . 7X,3HND, 6X 3HTAG 7X 16H1 2 3 4 4X 5HTEMP. ,6X,6HPPOINTS
SOL8 423 . 5X,4HBAND //)
SOL8 424 1302 FORMAT (//////26H ELEMENT DISTRIBUTED LOADS //
SOL8 425 . 52H NUMBER KTYPE PR YREF FACE I
SOL8 426 3000 FORMAT ( 31H1.....8 - NODE SOLID ELEMENTS ///
SOL8 427 . 24H NUMBER OF ELEMENTS.... ,15 //
SOL8 428 . 24H NUMBER OF MATERIALS... ,15 //
SOL8 429 . 24H NUMBER OF LOAD TYPES.. ,15 ///)
SOL8 430 4003 FORMAT (36HOELEMENT CARD ERROR, ELEMENT NUMBER= 16)
SOL8 431 C
SOL8 432 END

```

```

SOL8 433 SUBROUTINE DERIV(KK,DI)
SOL8 434 C
SOL8 435 DIMENSION D(12,1)
SOL8 436 COMMON /GASS/ XK(4,4),WGT(4,4),IPERM(3)
SOL8 437 COMMON /JUNK/ R,S,T,DET,MLD(4),KLD(4),MULT(4),NP(8),INP(8),
SOL8 438 A(3,3),P(3,11),B(3,3),XX(8,3),Q(11),DL(8)
SOL8 439 C
SOL8 440 RP=(1.+R)*.125
SOL8 441 RM=(1.-R)*.125
SOL8 442 SP=1.+S
SOL8 443 SM=1.-S
SOL8 444 TP=1.+T
SOL8 445 TM=1.-T
SOL8 446 IF (KK.EQ.2.OR.KK.EQ.4) GO TO 100
SOL8 447 C
SOL8 448 C SHAPE FUNCTIONS
SOL8 449 C
SOL8 450 Q(1) = RP*SM*TM
SOL8 451 Q(2) = RP*SP*TM
SOL8 452 Q(3) = RM*SP*TM
SOL8 453 Q(4) = RM*SM*TM
SOL8 454 Q(5) = RP*SM*TP
SOL8 455 Q(6) = RP*SP*TP
SOL8 456 Q(7) = RM*SP*TP
SOL8 457 Q(8) = RM*SM*TP
SOL8 458 C
SOL8 459 C DERIVATIVES OF SHAPE FUNCTIONS
SOL8 460 C
SOL8 461
SOL8 462 100 P(1,1) = SM*TM*.125
SOL8 463 P(1,2) = SP*TM*.125
SOL8 464 P(1,3) = -P(1,2)
SOL8 465 P(1,4) = -P(1,1)
SOL8 466 P(1,5) = SM*TP*.125
SOL8 467 P(1,6) = SP*TP*.125
SOL8 468 P(1,7) = -P(1,6)
SOL8 469 P(1,8) = -P(1,5)
SOL8 470 P(1,9) = -R
SOL8 471 P(1,10) = 0.
SOL8 472 P(1,11) = 0.
SOL8 473 C
SOL8 474 P(2,1) = -RP*TM
SOL8 475 P(2,2) = -P(2,1)
SOL8 476 P(2,3) = RM*TM
SOL8 477 P(2,4) = -P(2,3)
SOL8 478 P(2,5) = -RP*TP
SOL8 479 P(2,6) = -P(2,5)
SOL8 480 P(2,7) = RM*TP
SOL8 481 P(2,8) = -P(2,7)
SOL8 482 P(2,9) = 0.
SOL8 483 P(2,10) = -S
SOL8 484 P(2,11) = 0.
SOL8 485 C
SOL8 486 P(3,1) = -RP*SM
SOL8 487 P(3,2) = -RP*SP
SOL8 488 P(3,3) = -RM*SP
SOL8 489 P(3,4) = -RM*SM
SOL8 490 P(3,5) = -P(3,1)
SOL8 491 P(3,6) = -P(3,2)
SOL8 492 P(3,7) = -P(3,3)
SOL8 493 P(3,8) = -P(3,4)
SOL8 494 P(3,9) = 0.
SOL8 495 P(3,10) = 0.
SOL8 496 P(3,11) = -T
SOL8 497 C
SOL8 498 C JACOBIAN MATRIX A
SOL8 499 C
SOL8 500 DO 200 I=1,3
SOL8 501 DO 200 J=1,3
SOL8 502 C=0.
SOL8 503 DO 150 L=1,8
SOL8 504 150 C = C + P(I,L)*XX(L,J)
SOL8 505 200 A(I,J) = C
SOL8 506 C
SOL8 507 C INVERT JACOBIAN

```

```

SOL8 508 C
SOL8 509 IF(KK.EQ.3) GO TO 500
SOL8 510 DO 250 I=1,3
SOL8 511 J = IPERM(I)
SOL8 512 K = IPERM(J)
SOL8 513 B(I,I) = A(J,J)*A(K,K) - A(K,J)*A(I,K)
SOL8 514 B(I,J) = A(K,J)*A(I,K) - A(I,J)*A(K,K)
SOL8 515 250 B(J,I) = A(J,K)*A(K,I) - A(J,I)*A(K,K)
SOL8 516 IF (KK.EQ.4) GO TO 500
SOL8 517 DET = A(1,1)*B(1,1) + A(1,2)*B(2,1) + A(1,3)*B(3,1)
SOL8 518 C
SOL8 519 C MATRIX OF X-Y-Z DERIVATIVES
SOL8 520 DO 400 I=1,3
SOL8 521 DO 400 J=1,11
SOL8 522 C = 0.
SOL8 523 DO 350 K=1,3
SOL8 524 350 C = C + B(I,K)*P(K,J)
SOL8 525 400 D(J,I)=C/DET
SOL8 526 C
SOL8 527 500 RETURN
SOL8 528 C
SOL8 529 END

```

```

SOL8 530 SUBROUTINE LOAD (KTYPEE,PRR,YREFF,NFACE)
SOL8 531 C
SOL8 532 COMMON/EM/LM(24),ND,NS, ES(24,24),RF(24,4),XM(24),SA(12,24),
SOL8 533 SF(12,4)
SOL8 534 COMMON /JUNK/ ETA(3),DET,MLD(4),KLD(4),MULT(4),NP(8),INP(8),
SOL8 535 A(3,3),P(11,3),B(3,3),XX(8,3),Q(11),OL(8)
SOL8 536 DIMENSION KTYPEE(1),PRR(1),YREFF(1),NFACE(1)
SOL8 537 COMMON /GASS / DUM(12),XK(4),DDUM(12),WGT(4),IPERM(3)
SOL8 538 DIMENSION KCRD(6),FVAL(6),KFACE(6,4)
SOL8 539 C
SOL8 540 DATA KFACE / 1, 4, 2, 1, 6, 2,
SOL8 541 2, 3, 3, 4, 7, 3,
SOL8 542 2, 6, 7, 7, 8, 8, 4,
SOL8 543 3, 5, 8, 6, 5, 5, 1/
SOL8 544 DATA KCRD / 1,1,2,2,3,3/
SOL8 545 DATA FVAL / 1.,-1.,1.,-1.,1.,-1./
SOL8 546 C
SOL8 547 C
SOL8 548 DO 700 KK=1,4
SOL8 549 NNN=KLD(KK)
SOL8 550 IF(NNN) 700,700,10
SOL8 551 10 KTYPE=KTYPEE(NNN)
SOL8 552 PRR=PRR(NNN)
SOL8 553 YREF=YREFF(NNN)
SOL8 554 KF=NFACE(NNN)
SOL8 555 C
SOL8 556 C INTEGRATE OVER THE SURFACE
SOL8 557 C
SOL8 558 ML = KCRD(KF)
SOL8 559 MN = IPERM(ML)
SOL8 560 MM = IPERM(MN)
SOL8 561 ETA(ML) = FVAL(KF)
SOL8 562 DO 300 LX = 1,4
SOL8 563 ETA(MM) = XK(LX)
SOL8 564 DO 300 LY = 1,4
SOL8 565 ETA(MN) = XK(LY)
SOL8 566 CALL DERIV(3,SA)
SOL8 567 C
SOL8 568 C COMPUTE DIRECTION COSINES OF NORMAL TO SURFACE
SOL8 569 C
SOL8 570 A1 = (A(MM,2)*A(MN,3)-A(MM,3)*A(MN,2))
SOL8 571 A2 = (A(MM,3)*A(MN,1)-A(MM,1)*A(MN,3))
SOL8 572 A3 = (A(MM,1)*A(MN,2)-A(MM,2)*A(MN,1))
SOL8 573 AA=SQRT(A1**2+A2**2+A3**2)
SOL8 574 A1 = A1/AA
SOL8 575 A2 = A2/AA

```

```

SOL8 576      A3 = A3/AA
SOL8 577      C
SOL8 578      COMPUTE FIRST FUND. FORM (SIN / )
SOL8 579      C
SOL8 580      AA = 0.
SOL8 581      BB = 0.
SOL8 582      CC = 0.
SOL8 583      DD 200 I = 1,3
SOL8 584      AA=AA+A(MM,I)**2
SOL8 585      CC=CC+A(MM,I)**2
SOL8 586      200 BB = BB + A(MM,I)*A(MM,I)
SOL8 587      C=C+SQRT (AA*CC - BB*BB)
SOL8 588      C
SOL8 589      COMPUTE PRESSURE,LOAD COMPONENTS, STDR IN R
SOL8 590      C
SOL8 591      IF (KTYPE,EQ.2) GO TO 170
SOL8 592      FORCE = PR
SOL8 593      GO TO 185
SOL8 594      170 YY = 0.
SOL8 595      DD 180 I = 1,8
SOL8 596      180 YY = YY + Q(I)*XX(I,2)
SOL8 597      YY = YY - YREF
SOL8 598      FORCE = -PR*YY
SOL8 599      IF (YY.GT.0.) FORCE = 0.
SOL8 600      1E5 CONTINUE
SOL8 601      TS=FORCE*WGT(LX)*WGT(LY)*C
SOL8 602      C
SOL8 603      DD 190 I = 1,4
SOL8 604      N = KFACE(KF,I)
SOL8 605      QQ=TS*Q(IN)
SOL8 606      K=3*N
SOL8 607      RF(K-2,KK) = RF(K-2,KK) + QQ*A1
SOL8 608      RF(K-1,KK) = RF(K-1,KK) + QQ*A2
SOL8 609      RF(K ,KK) = RF(K ,KK) + QQ*A3
SOL8 610      190 CONTINUE
SOL8 611      C
SOL8 612      300 CONTINUE
SOL8 613      C
SOL8 614      700 CONTINUE
SOL8 615      C
SOL8 616      RETURN
SOL8 617      END

```

```

SOL8 618      SUBROUTINE LOSTR (IS,A,B,SA,SF,L)
SOL8 619      DIMENSION IS(2),A(3,3),B(3,3),SA(12,24),SF(12,4),IRF(6,2),TC(6,24)
SOL8 620      1,TR(6,6)
SOL8 621      DATA IRF /1,1,2,2,3,3,
SOL8 622      1      2,2,3,3,1,1/
SOL8 623      C
SOL8 624      LL=IS(L)
SOL8 625      I=IRF(LL,1)
SOL8 626      TT=B(I,I)*B(1,I)+B(2,I)*B(2,I)+B(3,I)*B(3,I)
SOL8 627      TT=SQRT(TT)
SOL8 628      TC(3,1)=B(1,I)/TT
SOL8 629      TC(3,2)=B(2,I)/TT
SOL8 630      TC(3,3)=B(3,I)/TT
SOL8 631      I=IRF(LL,2)
SOL8 632      TT=A(I,1)*A(1,1)+A(I,2)*A(1,2)+A(I,3)*A(1,3)
SOL8 633      TT=SQRT(TT)
SOL8 634      TC(1,1)=A(I,1)/TT
SOL8 635      TC(1,2)=A(I,2)/TT
SOL8 636      TC(1,3)=A(I,3)/TT
SOL8 637      TC(2,1)=TC(3,2)*TC(1,3)-TC(3,3)*TC(1,2)
SOL8 638      TC(2,2)=TC(3,3)*TC(1,1)-TC(3,1)*TC(1,3)
SOL8 639      TC(2,3)=TC(3,1)*TC(1,2)-TC(3,2)*TC(1,1)
SOL8 640      C
SOL8 641      TR(1,1)=TC(1,1)*TC(1,1)
SOL8 642      TR(1,2)=TC(1,2)*TC(1,2)
SOL8 643      TR(1,3)=TC(1,3)*TC(1,3)

```

```

SOL8 644      TR(1,4)=TC(1,1)*TC(1,2)*2.
SOL8 645      TR(1,5)=TC(1,2)*TC(1,3)*2.
SOL8 646      TR(1,6)=TC(1,1)*TC(1,3)*2.
SOL8 647      TR(2,1)=TC(2,1)*TC(2,1)
SOL8 648      TR(2,2)=TC(2,2)*TC(2,2)
SOL8 649      TR(2,3)=TC(2,3)*TC(2,3)
SOL8 650      TR(2,4)=TC(2,1)*TC(2,2)*2.
SOL8 651      TR(2,5)=TC(2,2)*TC(2,3)*2.
SOL8 652      TR(2,6)=TC(2,1)*TC(2,3)*2.
SOL8 653      TR(3,1)=TC(3,1)*TC(3,1)
SOL8 654      TR(3,2)=TC(3,2)*TC(3,2)
SOL8 655      TR(3,3)=TC(3,3)*TC(3,3)
SOL8 656      TR(3,4)=TC(3,1)*TC(3,2)*2.
SOL8 657      TR(3,5)=TC(3,2)*TC(3,3)*2.
SOL8 658      TR(3,6)=TC(3,1)*TC(3,3)*2.
SOL8 659      TR(4,1)=TC(1,1)*TC(2,1)
SOL8 660      TR(4,2)=TC(1,2)*TC(2,2)
SOL8 661      TR(4,3)=TC(1,3)*TC(2,3)
SOL8 662      TR(4,4)=TC(1,1)*TC(2,2)+TC(1,2)*TC(2,1)
SOL8 663      TR(4,5)=TC(1,2)*TC(2,3)+TC(1,3)*TC(2,2)
SOL8 664      TR(4,6)=TC(1,1)*TC(2,3)+TC(1,3)*TC(2,1)
SOL8 665      TR(5,1)=TC(2,1)*TC(3,1)
SOL8 666      TR(5,2)=TC(2,2)*TC(3,2)
SOL8 667      TR(5,3)=TC(2,3)*TC(3,3)
SOL8 668      TR(5,4)=TC(2,1)*TC(3,2)+TC(2,2)*TC(3,1)
SOL8 669      TR(5,5)=TC(2,2)*TC(3,3)+TC(2,3)*TC(3,2)
SOL8 670      TR(5,6)=TC(2,1)*TC(3,3)+TC(2,3)*TC(3,1)
SOL8 671      TR(6,1)=TC(3,1)*TC(1,1)
SOL8 672      TR(6,2)=TC(3,2)*TC(1,2)
SOL8 673      TR(6,3)=TC(3,3)*TC(1,3)
SOL8 674      TR(6,4)=TC(3,1)*TC(1,2)+TC(3,2)*TC(1,1)
SOL8 675      TR(6,5)=TC(3,2)*TC(1,3)+TC(3,3)*TC(1,2)
SOL8 676      TR(6,6)=TC(3,1)*TC(1,3)+TC(3,3)*TC(1,1)
SOL8 677      C
SOL8 678      IL=(L-1)*6
SOL8 679      DO 100 I=1,6
SOL8 680      DO 100 J=1,24
SOL8 681      TC(I,J)=0.
SOL8 682      DO 100 K=1,6
SOL8 683      100 TC(I,J)=TC(I,J)+TR(1,K)*SA(IL+K,J)
SOL8 684      DO 110 I=1,6
SOL8 685      DO 110 J=1,24
SOL8 686      110 SA(IL+I,J)=TC(I,J)
SOL8 687      C
SOL8 688      DO 120 I=1,6
SOL8 689      DO 120 J=1,4
SOL8 690      TC(I,J)=0.
SOL8 691      DO 120 K=1,6
SOL8 692      120 TC(I,J)=TC(I,J)+TR(I,K)*SF(IL+K,J)
SOL8 693      DO 130 I=1,6
SOL8 694      DO 130 J=1,4
SOL8 695      130 SF(IL+I,J)=TC(I,J)
SOL8 696      C
SOL8 697      RETURN
SOL8 698      END

```

```

SOL8 699      SUBROUTINE PRIST (NS,IS1,IS2,SIG,SPR)
SOL8 700      DIMENSION SIG(12),SPR(6),IS(2),SG(6)
SOL8 701      C
SOL8 702      IS(1)=IS1
SOL8 703      IS(2)=IS2
SOL8 704      NNS=1
SOL8 705      IF (NS.EQ.12) NNS=2
SOL8 706      DO 900 N=1,NNS
SOL8 707      IN=3*N-3
SOL8 708      II=IN*2
SOL8 709      IF (IS(N).EQ.0) GO TO 200
SOL8 710      C
SOL8 711      CC=(SIG(II+1)+SIG(II+2))/2.

```

```

SOL8 712      BB=(SIG(II+1)-SIG(II+2))/2.
SOL8 713      CR=SQR( BB**2+SIG(II+4)**2)
SOL8 714      SPR(IN+1)=CC+CR
SOL8 715      SPR(IN+2)=CC-CR
SOL8 716      SPR(IN+3)=0.
SOL8 717      IF (BB.NE.0.)SPR(IN+3)=28.648*ATAN2(SIG(II+4),BB)
SOL8 718      GO TO 900
SOL8 719      C
SOL8 720      200 CC=(SIG(II+1)+SIG(II+2)+SIG(II+3))/3.
SOL8 721      DO 210 I=1,3
SOL8 722      SG(I)=SIG(II+I)-CC
SOL8 723      210 SG(II+3)=SIG(II+I+3)
SOL8 724      C2=(SG(1)**2+SG(2)**2+SG(3)**2)*.5+SG(4)**2+SG(5)**2+SG(6)**2
SOL8 725      C3=SG(1)*(SG(2)*SG(3)-SG(5)*SG(5))+SG(4)*(SG(5)*SG(6)-SG(4)*SG(3))
SOL8 726      1+SG(6)*(SG(4)*SG(5)-SG(2)*SG(6))
SOL8 727      T=SQR(C2/1.5)
SOL8 728      A=C3*1.414214/T**3
SOL8 729      IF ( A .LT. -1.0 ) A=-1.0
SOL8 730      IF ( A .GT. 1.0 ) A= 1.0
SOL8 731      A=ACOS(A)/3.
SOL8 732      T=T*1.414214
SOL8 733      SPR(IN+1)=T*COS(A)
SOL8 734      SPR(IN+2)=T*COS(A+2.0944)
SOL8 735      SPR(IN+3)=T*COS(A-2.0944)
SOL8 736      DO 220 I=2,3
SOL8 737      IF (SPR(IN+1).GT.SPR(IN+I)) GC TO 220
SOL8 738      C3=SPR(IN+1)
SOL8 739      SPR(IN+1)=SPR(IN+I)
SOL8 740      SPR(IN+I)=C3
SOL8 741      220 CONTINUE
SOL8 742      IF (SPR(IN+2).LE.SPR(IN+3)) GO TO 230
SOL8 743      C3=SPR(IN+2)
SOL8 744      SPR(IN+2)=SPR(IN+3)
SOL8 745      SPR(IN+3)=C3
SOL8 746      230 DO 240 I=1,3
SOL8 747      240 SPR(IN+I)=SPR(IN+I)+CC
SOL8 748      900 CONTINUE
SOL8 749      C
SOL8 750      RETURN
SOL8 751      END

```

```

SHEL 1 SUBROUTINE SHELL
SHEL 2 COMMON A(1)
SHEL 3 COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
SHEL 4 COMMON / JUNK / LT,LH,L,SIG(20)
SHEL 5 C
SHEL 6 IF (NPAR(1).EQ.0) GO TO 500
SHEL 7 C
SHEL 8 PROTECT NODAL TEMPERATURES
SHEL 9 N6=N5+NUMNP+12*NPAR(3)
SHEL 10 IF (N6.GT.MTOT) CALL ERROR (N6-MTOT)
SHEL 11 CALL TPLATE(NPAR(2),NPAR(3),A(N1),A(N2),A(N3),A(N4),A(N5),NUMNP,
SHEL 12 1 MBAND)
SHEL 13 RETURN
SHEL 14 C
SHEL 15 500 WRITE (6,2002)
SHEL 16 NUME=NPAR(2)
SHEL 17 DO 800 MM=1,NUME
SHEL 18 CALL STRSC (A(N1),A(N3),NEQ,0)
SHEL 19 WRITE (6,2001)
SHEL 20 DO 800 L=L1,LH
SHEL 21 CALL STRSC (A(N1),A(N3),NEQ,1)
SHEL 22 WRITE (6,3002) MM,L,(SIG(I),I=1,6)
SHEL 23 800 CONTINUE
SHEL 24 C
SHEL 25 RETURN
SHEL 26 C
SHEL 27 2001 FORMAT (//)
SHEL 28 2002 FORMAT (24+1 SHELL ELEMENT STRESSES//
SHEL 29 1 102H ELEMENT LOAD MEMBRANE STRESS COMPONENTS
SHEL 30 2 BENDING MOMENT COMPONENTS /
SHEL 31 3 102H NUMBER CASE SXX SYY S
SHEL 32 4XY MXX MYY MXY //)
SHEL 33 3002 FORMAT (10X,2I10,6E12.4)
SHEL 34 END

```

```

SHEL 34 SUBROUTINE TPLATE(NUMEL,NUMMAT, ID,X,Y,Z,C,NUMNP,MBAND)
SHEL 35 C
SHEL 36 COMMON/QTSARG/
SHEL 37 IXX(5),YY(5),ZZ(5),HM(5),HP(5),CM(3,3),ALFA(3),HM(5),RHO(5,3),P(5)
SHEL 38 2,T(5),DT(5),SM(5,3),BM(5,3),TDIS(3,3,4),TROT(3,3,4),S(42,42),R(42)
SHEL 39 COMMON/EM/LM(24),ND,NS,ASA(24,24),RF(24,4),XM(24),SA(12,24)
SHEL 40 1,SP(12,4)
SHEL 41 DIMENSION X(1),Y(1),Z(1),IO(NUMNP,1),C(12,1),IX(7),IY(7),EL(6)
SHEL 42 1,TO(3,3),TLO(5,4)
SHEL 43 C
SHEL 44 LL=4
SHEL 45 C
SHEL 46 DO 4 I=1,12
SHEL 47 DO 4 J=1,4
SHEL 48 4 SF(I,J)=0.
SHEL 49 DO 5 I=1,24
SHEL 50 DO 5 J=1,4
SHEL 51 5 RF(I,J)=0.0
SHEL 52 NPF=6
SHEL 53 NS=6
SHEL 54 MID=4
SHEL 55 MID=0
SHEL 56 IDIS=0
SHEL 57 IROT=0
SHEL 58 ISTOP=0
SHEL 59 MTYPE=6
SHEL 60 WRITE (6,2000) MTYPE,NUMEL,NUMMAT
SHEL 61 C
SHEL 62 C *** READ AND PRINT OF MATERIAL PROPERTIES
SHEL 63 C
SHEL 64 WRITE (6,2001)
SHEL 65 DO 10 N=1,NUMMAT
SHEL 66 READ (5,1000) N,(C(I,N),I=1,12)
SHEL 67 10 WRITE (6,2002) N,(C(I,N),I=3,12)
SHEL 68 C

```

```

SHEL 69 C *** READ AND PRINT OF ELEMENT LOAD MULTIPLIERS
SHEL 70 C
SHEL 71 READ (5,1002) ((TLO(I,J),J=1,4),I=1,5)
SHEL 72 WRITE (6,2006)
SHEL 73 WRITE (6,2007) (J,(TLO(I,J),I=1,5),J=1,4)
SHEL 74 C
SHEL 75 C *** READ AND PRINT OF ELEMENT DATA
SHEL 76 C
SHEL 77 WRITE (6,2003)
SHEL 78 NN=0
SHEL 79 100 READ (5,1001) MM,IY,EL
SHEL 80 110 NN=NN+1
SHEL 81 IF (MM-NN) 440,50,60
SHEL 82 50 DO 45 I=1,7
SHEL 83 45 IX(I)=IY(I)
SHEL 84 INCL=IY(7)
SHEL 85 IF (IY(6).EQ.0) IY(6)=1
SHEL 86 IM=IY(6)
SHEL 87 IF (INCL.EQ.0) INCL=1
SHEL 88 NNS=5
SHEL 89 IF (IY(5).EQ.0) NNS=4
SHEL 90 IF (IY(4).EQ.0) NNS=3
SHEL 91 RHOM=C(3,1M)
SHEL 92 ALFA(1)=C(4,1M)
SHEL 93 ALFA(2)=C(5,1M)
SHEL 94 ALFA(3)=C(6,1M)
SHEL 95 CM(1,1)=C(7,1M)
SHEL 96 CM(1,2)=C(8,1M)
SHEL 97 CM(1,3)=C(9,1M)
SHEL 98 CM(2,2)=C(10,1M)
SHEL 99 CM(2,3)=C(11,1M)
SHEL 100 CM(3,3)=C(12,1M)
SHEL 101 CM(2,1)=CM(1,2)
SHEL 102 CM(3,1)=CM(1,3)
SHEL 103 CM(3,2)=CM(2,3)
SHEL 104 C
SHEL 105 DO 30 I=1,NNS
SHEL 106 HM(I)=EL(I)
SHEL 107 HP(I)=EL(I)
SHEL 108 HM(I)=EL(I)
SHEL 109 SM(1,1)=0.0
SHEL 110 SM(1,2)=0.0
SHEL 111 SM(1,3)=0.0
SHEL 112 BM(1,1)=0.0
SHEL 113 BM(1,2)=0.0
SHEL 114 30 BM(1,3)=0.0
SHEL 115 GO TO 70
SHEL 116 C
SHEL 117 60 DO 65 I=1,NNS
SHEL 118 65 IX(I)=IX(I)+INCL
SHEL 119 C
SHEL 120 70 DO 40 I=1,NNS
SHEL 121 J=IX(I)
SHEL 122 XX(I)=X(J)
SHEL 123 YY(I)=Y(J)
SHEL 124 40 ZZ(I)=Z(J)
SHEL 125 C
SHEL 126 DO 550 IL=1,LL
SHEL 127 DO 520 I=1,NNS
SHEL 128 RHO(I,1)=TLO(3,IL)*RHCP
SHEL 129 RHO(I,2)=TLO(4,IL)*RHOM
SHEL 130 RHO(I,3)=TLO(5,IL)*RHCM
SHEL 131 P(I)=TLO(1,IL)*EL(2)
SHEL 132 T(I)=TLO(2,IL)*EL(3)
SHEL 133 520 DT(I)=TLO(2,IL)*EL(4)
SHEL 134 IF (IL.GE.2) GO TO 530
SHEL 135 C
SHEL 136 CALL QTSHEL (0,NNS,NPF,MID,IOIS,IROT,ND ,NTFI)
SHEL 137 C
SHEL 138 DO 510 I=1,ND
SHEL 139 RF(I)=R(I)
SHEL 140 XM(I)=0.
SHEL 141 DO 510 J=1,ND
SHEL 142 510 ASA(I,J)=S(I,J)
SHEL 143 GO TO 550

```

```

SHEL 144 C
SHEL 145 530 CALL QTSHEL (1,NNS,NPF,MID,IOIS,IROT,ND ,NTF)
SHEL 146 C
SHEL 147 DO 540 I=1,ND
SHEL 148 540 RF(I,IL)=R(II)
SHEL 149 550 CONTINUE
SHEL 150 WRITE (6,2004) NN,(IX(I),I=1,6),HM(1),P(1),T(1),DT(1)
SHEL 151 C
SHEL 152 C *** FORM LM ARRAY AND COMPUTE BANDWIDTH
SHEL 153 C
SHEL 154 DO 410 I=1,NNS
SHEL 155 J=NPF*I-NPF
SHEL 156 N=IX(I)
SHEL 157 DO 410 K=1,NPF
SHEL 158 410 LM(J+K)=ID(N,K)
SHEL 159 C
SHEL 160 CALL QDCOS (4,XX,YY,ZZ,TO)
SHEL 161 H=HM(1)
SHEL 162 C
SHEL 163 CALL STRETR (TO,XX,YY,ZZ,CM,SA,H,RHOM,XM)
SHEL 164 C
SHEL 165 NDM=24
SHEL 166 CALL CALBAN (MBAND,NDIF,LM,XM,ASA,RF,NO,NDM)
SHEL 167 WRITE (1) NO,NS,(LM(I),I=1,ND),((SA(I,J),I=1,NS),J=1,ND),
SHEL 168 1 ((SF(I,J),I=1,NS),J=1,4)
SHEL 169 C
SHEL 170 GO TO 500
SHEL 171 440 WRITE (6,2005) MM
SHEL 172 ISTOP=1
SHEL 173 500 IF (NN.LT.MM) GO TO 110
SHEL 174 IF (NN.EQ.NUMEL) RETURN
SHEL 175 IF (ISTOP.EQ.1) STOP
SHEL 176 GO TO 100
SHEL 177 C
SHEL 178 2000 FORMAT (34H1...THIN ELASTIC SHELL ELEMENTS...//
SHEL 179 1 25H ELEMENT TYPE.....I5//
SHEL 180 2 25H NUMBER OF ELEMENTS.....I5//
SHEL 181 3 25H NUMBER OF MATERIALS.....I9//
SHEL 182 1000 FORMAT (I10,6F10.0/6F10.0)
SHEL 183 2001 FORMAT (36H1...MATERIAL PROPERTY INFORMATION...//
SHEL 184 1 131H MASS THERMAL EXPANSION COEF
SHEL 185 2FICIENTS .....ELASTIC COEFFICIENTS.....
SHEL 186 3...../
SHEL 187 4 131H TYPE DENSITY AX AY
SHEL 188 5 AZ CXX CXY CXG CYY CYG
SHEL 189 6GX //)
SHEL 190 2002 FORMAT (1H I4,22X,4E11.4,6E10.4)
SHEL 191 1002 FORMAT (4F10.0)
SHEL 192 2006 FORMAT (31H1...ELEMENT LOAD MULTIPLIERS...//
SHEL 193 1 90H .....GRAVI
SHEL 194 2TY ACCELERATION..... /
SHEL 195 3 90H LOAD CASE LATERAL LOAD TEMPERATURE X-COMPONENT
SHEL 196 4Y-COMPONENT Z-COMPONENT //)
SHEL 197 2007 FORMAT (1X,I10,5X,5F15.5)
SHEL 198 2003 FORMAT (15H1*ELEMENT DATA**//
SHEL 199 1 96H ELEMENT NODAL PCINT NUMBERS MATL
SHEL 200 2 TEMPERATURE/
SHEL 201 3 96H NUMBER I J K L O TYPE THICKNESS
SHEL 202 4 PRESSURE TEMPERATURE GRADIENT //)
SHEL 203 1001 FORMAT (8I5,4F10.0)
SHEL 204 2004 FORMAT (3X,I5,5X,5I5,5X,I5,4F12.4)
SHEL 205 2005 FORMAT (23H0 ELEMENT CARD IN ERROR,I5)
SHEL 206 C
SHEL 207 END

```

```

SUBROUTINE STRETR (T,X,Y,Z,CM,SA,H,RHOM,XM)
SHEL 208
SHEL 209 C
SHEL 210 C *** FORM DISPLACEMENT-STRESS TRANSFORMATION MATRIX
SHEL 211 C
SHEL 212 DIMENSION T(3,3),X(1),Y(1),Z(1),CM(3,3),SA(12,24),AA(12,12),MM(12)
SHEL 213 1,A(12,12),AM(4),S1(3,9),S2(3,9),S3(3,12),V(3,3),XY(3,2),XM(1)
SHEL 214 C
SHEL 215 DO 5 I=1,12
SHEL 216 DO 5 J=1,24
SHEL 217 5 SA(I,J)=0.
SHEL 218 DO 10 I=1,3
SHEL 219 V(I,1)=X(I+1)-X(1)
SHEL 220 V(I,2)=Y(I+1)-Y(1)
SHEL 221 10 V(I,3)=Z(I+1)-Z(1)
SHEL 222 DO 20 I=1,3
SHEL 223 DO 20 J=1,2
SHEL 224 XY(I,J)=0.
SHEL 225 DO 20 K=1,3
SHEL 226 20 XY(I,J)=XY(I,J)+V(I,K)*T(J,K)
SHEL 227 XO=(XY(1,1)+XY(2,1)+XY(3,1))/4.0
SHEL 228 YO=(XY(1,2)+XY(2,2)+XY(3,2))/4.0
SHEL 229 C
SHEL 230 C *** FORM LUMP MASS MATRIX
SHEL 231 C
SHEL 232 AM1=XY(1,1)*YO-XO*XY(1,2)
SHEL 233 AM2=(XY(1,1)-XO)*(XY(2,2)-YO)-(XY(2,1)-XO)*(XY(1,2)-YO)
SHEL 234 AM3=(XY(2,1)-XO)*(XY(3,2)-YO)-(XY(3,1)-XO)*(XY(2,2)-YO)
SHEL 235 AM4=XO*XY(3,2)-XY(3,1)*YO
SHEL 236 AC=0.25*H*RHOM
SHEL 237 AM(1)=AC*(AM1+AM4)
SHEL 238 AM(2)=AC*(AM1+AM2)
SHEL 239 AM(3)=AC*(AM2+AM3)
SHEL 240 AM(4)=AC*(AM3+AM4)
SHEL 241 C
SHEL 242 DO 25 L=1,4
SHEL 243 XX=AM(L)
SHEL 244 DO 25 J=1,3
SHEL 245 JS=6*(L-1)+J
SHEL 246 XM(JS)=XX
SHEL 247 25 XM(JS+3)=0.
SHEL 248 C
SHEL 249 C *** FORM STRESS TRANSFORMATION FOR IN PLANE STRESS
SHEL 250 C
SHEL 251 S=0.5
SHEL 252 R=0.5
SHEL 253 DO 100 I=1,3
SHEL 254 DO 100 J=1,8
SHEL 255 100 S1(I,J)=0.0
SHEL 256 C
SHEL 257 XYJ=XY(1,1)*XY(3,2)-XY(3,1)*XY(1,2)+IXY(1,1)*(XY(2,2)-XY(3,2))
SHEL 258 1 -(XY(2,2)-XY(3,1))*XY(1,2))*S-((XY(1,1)-XY(2,1))*XY(3,2)
SHEL 259 2 -XY(3,1)*(XY(1,2)-XY(2,1)))*R
SHEL 260 X1=XY(1,2)-XY(3,2)-(XY(2,2)-XY(3,2))*S-(XY(1,2)-XY(2,2))*R
SHEL 261 X2=XY(3,2)+(XY(2,2)-XY(3,2))*S-XY(3,2)*R
SHEL 262 X3=-XY(1,2)*S+XY(3,2)*R
SHEL 263 X4=-XY(1,2)+XY(1,2)*S+IXY(1,2)-XY(2,2))*R
SHEL 264 Y1=-XY(1,1)+XY(3,1)+(XY(2,1)-XY(3,1))*S+(XY(1,1)-XY(2,1))*R
SHEL 265 Y2=-XY(3,1)-IXY(2,1)-XY(3,1))*S+XY(3,1))*R
SHEL 266 Y3=XY(1,1)*S-XY(3,1)*R
SHEL 267 Y4=XY(1,1)-XY(1,1)*S-(XY(1,1)-XY(2,1))*R
SHEL 268 C
SHEL 269 S1(1,1)=X1/XYJ
SHEL 270 S1(1,3)=X2/XYJ
SHEL 271 S1(1,5)=X3/XYJ
SHEL 272 S1(1,7)=X4/XYJ
SHEL 273 S1(2,2)=Y1/XYJ
SHEL 274 S1(2,4)=Y2/XYJ
SHEL 275 S1(2,6)=Y3/XYJ
SHEL 276 S1(2,8)=Y4/XYJ
SHEL 277 S1(3,1)=S1(2,2)
SHEL 278 S1(3,2)=S1(1,1)
SHEL 279 S1(3,3)=S1(2,4)
SHEL 280 S1(3,4)=S1(1,3)
SHEL 281 S1(3,5)=S1(2,6)
SHEL 282 S1(3,6)=S1(1,5)

```

D-24

```

SHEL 283      S1(3,7)=S1(2,8)
SHEL 284      S1(3,8)=S1(1,7)
SHEL 285      DO 105 I=1,3
SHEL 286      DO 105 J=1,8
SHEL 287      S2(I,J)=0.
SHEL 288      DO 104 K=1,3
SHEL 289      104 S2(I,J)=S2(I,J)+CM(I,K)*S1(K,J)
SHEL 290      105 S2(I,J)=S2(I,J)*H
SHEL 291      C
SHEL 292      H=H**3/12.
SHEL 293      DO 110 L=1,4
SHEL 294      DO 110 I=1,3
SHEL 295      DO 110 J=1,3
SHEL 296      JS=6*(L-1)+J
SHEL 297      KA=2*(L-1)+1
SHEL 298      110 SA(I,JS)=S2(I,KA)*T(I,J)+S2(I,KA+1)*T(2,J)
SHEL 299      C
SHEL 300      C *** FORM STRESS TRANSFORMATION FOR MOMENT
SHEL 301      C
SHEL 302      DO 30 I=1,12
SHEL 303      DO 30 J=1,12
SHEL 304      AA(I,J)=0.
SHEL 305      30 A(I,J)=0.
SHEL 306      C
SHEL 307      DO 40 I=1,3
SHEL 308      J=3*I+1
SHEL 309      A(1,J)=1.
SHEL 310      A(2,J)=XY(I,1)
SHEL 311      A(3,J)=XY(I,2)
SHEL 312      A(4,J)=XY(I,1)*XY(I,1)
SHEL 313      A(5,J)=XY(I,1)*XY(I,2)
SHEL 314      A(6,J)=XY(I,2)*XY(I,2)
SHEL 315      A(7,J)=A(4,J)*XY(I,1)
SHEL 316      A(8,J)=A(5,J)*XY(I,1)
SHEL 317      A(9,J)=A(5,J)*XY(I,2)
SHEL 318      A(10,J)=A(6,J)*XY(I,2)
SHEL 319      A(11,J)=XY(I,1)*XY(I,1)*XY(I,1)*XY(I,2)
SHEL 320      A(12,J)=XY(I,1)*XY(I,2)*XY(I,2)*XY(I,2)
SHEL 321      A(3,J+1)=1.0
SHEL 322      A(5,J+1)=XY(I,1)
SHEL 323      A(6,J+1)=2.0*XY(I,2)
SHEL 324      A(8,J+1)=XY(I,1)*XY(I,1)
SHEL 325      A(9,J+1)=2.0*XY(I,1)*XY(I,2)
SHEL 326      A(10,J+1)=3.0*XY(I,2)*XY(I,2)
SHEL 327      A(11,J+1)=XY(I,1)*XY(I,1)*XY(I,1)
SHEL 328      A(12,J+1)=3.0*XY(I,1)*XY(I,2)*XY(I,2)
SHEL 329      A(2,J+2)=-1.0
SHEL 330      A(4,J+2)=-2.0*XY(I,1)
SHEL 331      A(5,J+2)=-XY(I,2)
SHEL 332      A(7,J+2)=-3.0*XY(I,1)*XY(I,1)
SHEL 333      A(8,J+2)=-2.0*XY(I,1)*XY(I,2)
SHEL 334      A(9,J+2)=-XY(I,2)*XY(I,2)
SHEL 335      A(11,J+2)=-3.0*XY(I,1)*XY(I,1)*XY(I,2)
SHEL 336      A(12,J+2)=-XY(I,2)*XY(I,2)*XY(I,2)
SHEL 337      40 CONTINUE
SHEL 338      A(1,1)=1.0
SHEL 339      A(3,2)=1.0
SHEL 340      A(2,3)=-1.0
SHEL 341      C
SHEL 342      CALL INVERT (A,12,12,MM,AA)
SHEL 343      C
SHEL 344      DO 210 I=1,3
SHEL 345      DO 210 J=1,9
SHEL 346      210 S1(I,J)=0.
SHEL 347      C
SHEL 348      S1(1,1)=2.0
SHEL 349      S1(1,4)=6.0*X0
SHEL 350      S1(1,5)=2.0*Y0
SHEL 351      S1(1,8)=6.0*X0*Y0
SHEL 352      S1(2,3)=2.0
SHEL 353      S1(2,6)=2.0*X0
SHEL 354      S1(2,7)=6.0*Y0
SHEL 355      S1(2,9)=6.0*X0*Y0
SHEL 356      S1(3,2)=2.0
SHEL 357      S1(3,5)=4.0*X0

```

```

SHEL 358      S1(3,6)=4.0*Y0
SHEL 359      S1(3,8)=6.0*X0*X0
SHEL 360      S1(3,9)=6.0*Y0*Y0
SHEL 361      C
SHEL 362      DO 220 I=1,3
SHEL 363      DO 220 J=1,9
SHEL 364      S2(I,J)=0.
SHEL 365      DO 215 K=1,3
SHEL 366      215 S2(I,J)=S2(I,J)+CM(I,K)*S1(K,J)
SHEL 367      220 S2(I,J)=S2(I,J)*H
SHEL 368      C
SHEL 369      DO 230 I=1,3
SHEL 370      DO 230 J=1,12
SHEL 371      S3(I,J)=0.
SHEL 372      DO 230 K=1,9
SHEL 373      230 S3(I,J)=S3(I,J)+S2(I,K)*A(J,3+K)
SHEL 374      C
SHEL 375      DO 240 L=1,4
SHEL 376      DO 240 I=1,3
SHEL 377      DO 240 J=1,3
SHEL 378      JS=6*(L-1)+J
SHEL 379      KA=3*(L-1)+1
SHEL 380      SA(I+3,JS)=S3(I,KA)*T(3,J)
SHEL 381      JS=JS+3
SHEL 382      SA(I+3,JS)=S3(I,KA+1)*T(1,J)+S3(I,KA+2)*T(2,J)
SHEL 383      240 CONTINUE
SHEL 384      RETURN
SHEL 385      END

```

```

SHEL 386      SUBROUTINE INVERT(A,NN,N,M,C)
SHEL 387      C
SHEL 388      GENERAL MATRIX INVERSION SUBROUTINE
SHEL 389      C
SHEL 390      DIMENSION A(1),M(1),C(1)
SHEL 391      C
SHEL 392      DO 90 I=1,NN
SHEL 393      90 M(I)=-I
SHEL 394      C
SHEL 395      DO 140 I=1,NN
SHEL 396      C
SHEL 397      LOCATE LARGEST ELEMENT
SHEL 398      C
SHEL 399      D=0.0
SHEL 400      DO 112 L=1,NN
SHEL 401      IF (M(L)) 100,100,112
SHEL 402      100 J=L
SHEL 403      DO 110 K=1,NN
SHEL 404      IF (M(K)) 103,103,108
SHEL 405      103 IF (ABS(D)-ABS(A(J,I))) 105,105,108
SHEL 406      105 LD=L
SHEL 407      KD=K
SHEL 408      D=A(J)
SHEL 409      108 J=J+N
SHEL 410      110 CONTINUE
SHEL 411      112 CONTINUE
SHEL 412      C
SHEL 413      INTERCHANGE ROWS
SHEL 414      C
SHEL 415      TEMP=-N(LD)
SHEL 416      M(LD)=M(KD)
SHEL 417      M(KD)=TEMP
SHEL 418      L=LD
SHEL 419      K=KD
SHEL 420      DO 114 J=1,NN
SHEL 421      C(J)=A(L)
SHEL 422      A(L)=A(K)
SHEL 423      A(K)=C(J)
SHEL 424      L=L+N
SHEL 425      114 K=K+N

```

```

SHEL 426 C
SHEL 427 C DIVIDE COLUMN BY LARGEST ELEMENT
SHEL 428 C
SHEL 429 NR=(KD-1)*N+1
SHEL 430 NH=NR*N-1
SHEL 431 DO 115 K=NR,NH
SHEL 432 115 A(K)=A(K)/D
SHEL 433 C
SHEL 434 C REDUCE REMAINING ROWS AND COLUMNS
SHEL 435 C
SHEL 436 L=1
SHEL 437 DO 135 J=1,NN
SHEL 438 IF (J-KD) 130,125,130
SHEL 439 125 L=L+N
SHEL 440 GO TO 135
SHEL 441 130 DO 134 K=NR,NH
SHEL 442 A(L)=A(L)-C(J)*A(K)
SHEL 443 134 L=L+1
SHEL 444 135 CONTINUE
SHEL 445 C
SHEL 446 C REDUCE ROW
SHEL 447 C
SHEL 448 C(KD)=-1.0
SHEL 449 J=KD
SHEL 450 DO 140 K=1,NN
SHEL 451 A(J)=-C(K)/D
SHEL 452 140 J=J+N
SHEL 453 C
SHEL 454 C INTERCHANGE COLUMNS
SHEL 455 C
SHEL 456 DO 200 I=1,NN
SHEL 457 L=0
SHEL 458 150 L=L+1
SHEL 459 IF(M(L)-I) 150,160,150
SHEL 460 160 K=(L-I)*N+1
SHEL 461 J=(I-I)*N+1
SHEL 462 M(L)=M(I)
SHEL 463 M(I)=I
SHEL 464 DO 200 L=1,NN
SHEL 465 TEMP=A(K)
SHEL 466 A(K)=A(J)
SHEL 467 A(J)=TEMP
SHEL 468 J=J+1
SHEL 469 200 K=K+1
SHEL 470 C
SHEL 471 C RETURN
SHEL 472 C
SHEL 473 C END

SHEL 474 SUBROUTINE QTSHEL (KKK,NNS,NPF,MID,IDIS,IROT,NEF,NTF)
SHEL 475 C
SHEL 476 C THIS SUBROUTINE CAN EVALUATE
SHEL 477 C ... ELEMENT STIFFNESS MATRIX ...
SHEL 478 C ... CONSISTENT NODAL FORCE VECTOR ...
SHEL 479 C ... INTERNAL STRESSES AND MOMENTS ...
SHEL 480 C OF A SHALLOW QUADRILATERAL SHELL ELEMENT ASSEMBLED WITH 4 FLAT
SHEL 481 C TRIANGLES, OR OF A SINGLE TRIANGULAR SHELL ELEMENT.
SHEL 482 C
SHEL 483 C * * * * * CALLING ARGUMENTS * * * * *
SHEL 484 C
SHEL 485 C
SHEL 486 C INPUTS
SHEL 487 C
SHEL 488 C KKK INTEGER FLAG SPECIFYING OPERATION TO BE PERFORMED
SHEL 489 C IF KKK = -1, FORM STIFFNESS MATRIX ONLY.
SHEL 490 C IF KKK = 0, FORM STIFFNESS MATRIX AND LOAD VECTOR.
SHEL 491 C IF KKK = 1, FORM LOAD VECTOR ONLY.
SHEL 492 C IF KKK = 2, EVALUATE STRESSES AND MOMENTS.
SHEL 493 C

```

D-26

```

SHEL 494 C NNS NUMBER OF SUPPLIED NODAL POINTS
SHEL 495 C IF NNS = 5, QTSHEL FORMS A QUADRILATERAL, AND THE
SHEL 496 C PROPERTIES AT THE INTERNAL NODE 5 MUST BE INPUT.
SHEL 497 C IF NNS = 4, QTSHEL FORMS A QUADRILATERAL, AND THE
SHEL 498 C PROPERTIES AT THE INTERNAL NODE 5 ARE SET BY QTSHEL
SHEL 499 C TO BE THEIR CORNER AVERAGE.
SHEL 500 C IF NNS = 3, QDSTIF FORMS A SINGLE TRIANGLE.
SHEL 501 C
SHEL 502 C NPF NUMBER OF GLOBAL DEGREES OF FREEDOM AT EACH
SHEL 503 C EXTERNAL NODE (3, 5 OR 6)
SHEL 504 C IF NPF = 6, THE 3 DISPLACEMENTS U, V AND W AND THE
SHEL 505 C 3 ROTATIONS RX, RY AND RZ ARE INCLUDED AS D.O.F.
SHEL 506 C IF NPF = 5, THE ROTATION RZ IS IGNORED.
SHEL 507 C IF NPF = 3, ONLY U, V AND W ARE CONSIDERED AND
SHEL 508 C THE BENDING STIFFNESS IS NOT INCLUDED (MEMBRANE
SHEL 509 C SHELL ELEMENT)
SHEL 510 C
SHEL 511 C MID NUMBER OF INTERNAL MIDPOINTS IN QUADRILATERAL (0 OR 4)
SHEL 512 C IF MID = 0, THE MEMBRANE ELEMENTS ARE CST
SHEL 513 C AND THE BENDING ELEMENTS ARE LCCT-9
SHEL 514 C IF MID = 4, THE MEMBRANE ELEMENTS ARE LST-10
SHEL 515 C AND THE BENDING ELEMENTS ARE LCCT-11
SHEL 516 C IF NNS = 3 (SINGLE TRIANGLE) MID IS ASSUMED TO BE 0
SHEL 517 C
SHEL 518 C IDIS INTEGER FLAG FOR THE NODAL DISPLACEMENTS U,V,W
SHEL 519 C IF IDIS = 0, U,V,W ARE SPECIFIED IN THE GLOBAL SYSTEM
SHEL 520 C IF IDIS = 1, U,V,W ARE SPECIFIED IN THE NODAL DISPL
SHEL 521 C SYSTEMS DEFINED BY THE DIRECTION COSINE ARRAY TDIS.
SHEL 522 C
SHEL 523 C IROT INTEGER FLAG FOR THE NODAL ROTATIONS RX,RY,RZ.
SHEL 524 C IF IROT = 0, RX,RY,RZ ARE SPECIFIED IN GLOBAL SYSTEM
SHEL 525 C IF IROT = 1, RX,RY,RZ ARE SPECIFIED IN THE NODAL R3T
SHEL 526 C SYSTEMS DEFINED BY THE DIRECTION COSINE ARRAY TROT.
SHEL 527 C
SHEL 528 C OUTPUTS
SHEL 529 C
SHEL 530 C NEF NUMBER OF EXTERNAL DEGREES OF FREEDOM (NEF = NPF*NNEN,
SHEL 531 C WHERE NNEN=4 FOR QUADRILATERAL, =3 FOR SINGLE TRIANGLE)
SHEL 532 C
SHEL 533 C NTF TOTAL NUMBER OF DEGREES OF FREEDOM (EXTERNAL+INTERNAL)
SHEL 534 C
SHEL 535 C
SHEL 536 C * * * * * ARRAYS IN COMMON /QTSARG/ * * * * *
SHEL 537 C
SHEL 538 C X(I),Y(I),Z(I) I=1...NNS GLOBAL NODAL COORDINATES
SHEL 539 C
SHEL 540 C CH(I,J) I=1...3, J=1...3 PLANE STRESS MATERIAL MATRIX
SHEL 541 C RELATING STRESSES TO STRAINS IN THE LOCAL SYSTEM
SHEL 542 C
SHEL 543 C ALFA(I) I=1...3 DILATATION COEFFICIENTS RELATING IN-PLANE
SHEL 544 C THERMAL STRAINS IN THE LOCAL SYSTEM TO TEMPERATURES
SHEL 545 C
SHEL 546 C HM(I) I=1...NNS THICKNESS RESISTING MEMBRANE STRESSES
SHEL 547 C
SHEL 548 C HP(I) I=1...NNS THICKNESS RESISTING BENDING MOMENTS
SHEL 549 C
SHEL 550 C RHO(I,J) I=1...NNS, J=1...3 GLOBAL COMPONENTS RHOX (J=1),
SHEL 551 C RHOY (J=2) AND RHOZ (J=3) OF BODY FORCES PER UNIT
SHEL 552 C OF VOLUME
SHEL 553 C
SHEL 554 C HW(I) I=1...NNS THICKNESS FOR COMPUTING BODY FORCES
SHEL 555 C RHO*HW PER UNIT OF ELEMENT AREA
SHEL 556 C
SHEL 557 C P(I) I=1...NNS LATERAL PRESSURE (NORMAL TO THE FACES OF
SHEL 558 C THE COMPONENT TRIANGLES)
SHEL 559 C
SHEL 560 C T(I) I=1...NNS MEAN TEMPERATURE VARIATIONS
SHEL 561 C
SHEL 562 C DT(I) I=1...NNS MEAN TEMPERATURE THICKNESS GRADIENTS
SHEL 563 C
SHEL 564 C SM(I,J) I=1...NNS, J=1...3 ARRAY OF MEMBRANE STRESS
SHEL 565 C COMPONENTS IN THE LOCAL SYSTEM SIG-XX (J=1), SIG-YY
SHEL 566 C (J=2) AND SIG-XY (J=3). SM CONTAINS
SHEL 567 C MEMBRANE STRESSES IN THE INITIAL POSITION AS INPUT
SHEL 568 C WHEN KKK=0,1,2 (EXCLUDING THERMAL ACTIONS)

```

```

SHEL 569 C      MEMBRANE STRESSES IN THE DEFORMED POSITION AS OUTPUT
SHEL 570 C      WHEN KKK=2 (INCLUDING THERMAL ACTIONS)
SHEL 571 C
SHEL 572 C      BM(I,J)  I=1...NNS, J=1...3  ARRAY OF BENDING MOMENT
SHEL 573 C      COMPONENTS IN THE LOCAL SYSTEM MOM-XX (J=1), MOM-YY
SHEL 574 C      (J=2) AND MOM-XY (J=3). BM CONTAINS
SHEL 575 C      BENDING MOMENTS IN THE INITIAL POSITION AS INPUTS
SHEL 576 C      WHEN KKK=0,1,2 (EXCLUDING THERMAL ACTIONS)
SHEL 577 C      BENDING MOMENTS IN THE DEFORMED POSITION AS OUTPUT
SHEL 578 C      WHEN KKK=2 (INCLUDING THERMAL ACTION)
SHEL 579 C
SHEL 580 C      TOIS(I,J,K) I=1...3, J=1...3, K=1...NEN  NOT REQUIRED IF
SHEL 581 C      IDIS=0. IF IDIS=1, TOIS(1..3,1..3,K) MUST CONTAIN
SHEL 582 C      THE (3,3) DIRECTION COSINE MATRIX OF THE NODAL
SHEL 583 C      DISPLACEMENT SYSTEM AT THE K-TH ELEMENT NODE WITH
SHEL 584 C      RESPECT TO THE GLOBAL SYSTEM
SHEL 585 C
SHEL 586 C      TROT(I,J,K) I=1...3, J=1...3, K=1...NEN  NOT REQUIRED IF
SHEL 587 C      IROT=0. IF IROT=1, TROT(1..3,1..3,K) MUST CONTAIN
SHEL 588 C      THE (3,3) DIRECTION COSINE MATRIX OF THE NODAL
SHEL 589 C      ROTATION SYSTEM AT THE K-TH ELEMENT NODE WITH
SHEL 590 C      RESPECT TO THE GLOBAL SYSTEM
SHEL 591 C
SHEL 592 C      S(I,J)      I=1...NEF, J=1...NEF  EXTERNAL STIFFNESS MATRIX
SHEL 593 C      (OUTPUT IF KKK=-1,0)
SHEL 594 C
SHEL 595 C      S(I,J)      I=1...NTF, J=NEF+1...NTF  REDUCED INTERNAL STIFFNESS
SHEL 596 C      OF QUADRILATERAL ELEMENT. OUTPUT IF KKK=-1,0.
SHEL 597 C      REQUIRED INPUT IF KKK=1,2. NOT USED FOR SINGLE
SHEL 598 C      TRIANGLE.
SHEL 599 C
SHEL 600 C      R(I)       I=1...NEF  OUTPUT EXTERNAL NODAL FORCES IF KKK=0,1.
SHEL 601 C      INPUT EXTERNAL NODAL DISPLACEMENTS IF KKK=2.
SHEL 602 C
SHEL 603 C      R(I)       I=NEF+1...NTF  REDUCED INTERNAL NODAL FORCE VECTOR
SHEL 604 C      OF QUADRILATERAL ELEMENT. OUTPUT IF KKK=0,1.
SHEL 605 C      REQUIRED INPUT IF KKK=2 (RETURNS INTERNAL NODAL
SHEL 606 C      DISPLACEMENTS). NOT USED FOR SINGLE TRIANGLE.
SHEL 607 C
SHEL 608 C
SHEL 609 C      * * * * * ROLE OF ARRAYS IN COMMON /QTSARG/ * * * * *
SHEL 610 C
SHEL 611 C      ARRAYS      OPERATION  KKK = -1  KKK = 0  KKK = 1  KKK = 2
SHEL 612 C      Q T          Q T          Q T          Q T
SHEL 613 C
SHEL 614 C      X,Y,Z,CM,ALFA,HM,HP (*)  I  I      I  I      I  I      I  I
SHEL 615 C      RHO,HM,P                  -  -      I  I      I  I      -  -
SHEL 616 C      T,DT (*)                  -  -      I  I      I  I      I  I
SHEL 617 C      SM,BM (*)                  -  -      I  I      Y  I      I/O I/O
SHEL 618 C      TOIS,TROT (**)            I  I      I  I      I  I      I  I
SHEL 619 C      S(1..NEF,1..NEF)          0  0      0  0      -  -      -  -
SHEL 620 C      S(1..NTF,NEF+1..NTF)      0  -      0  -      I  -      I  -
SHEL 621 C      R(1..NEF)                 -  -      0  0      0  0      I  I
SHEL 622 C      R(NEF+1..NTF)             -  -      0  -      0  -      I/O -
SHEL 623 C
SHEL 624 C      WHERE  Q=QUADRILATERAL (NNS=4,5), T=SINGLE TRIANGLE (NNS=3)
SHEL 625 C      I=INPUT, O=OUTPUT, I/O=INPUT/OUTPUT, --=NOT USED.
SHEL 626 C
SHEL 627 C      NOTES  (*) HP,DT AND BM ARE NOT USED IF NPF=3.
SHEL 628 C      (**) TOIS IS NOT USED IF IDIS=0, AND TROT IS NOT USED
SHEL 629 C      IF IROT=0.
SHEL 630 C
SHEL 631 C
SHEL 632 C      COMMON /QTSARG/ X(5),Y(5),Z(5), HM(5),HP(5), CM(3,3),ALFA(3),
SHEL 633 C      1 HW(5),RHO(5,3),P(5), T(5),DT(5), SM(5,3),BM(5,3), TOIS(3,3,4),
SHEL 634 C      2 TROT(3,3,4), S(42,42), R(42)
SHEL 635 C      COMMON /TRIARG/ A(3),B(3), HMT(3),HPT(3), C(3,3), SMT(3,3),
SHEL 636 C      1 BMT(3,3), FT(12), P1(3),P2(3),P3(3),RM(3), ST(12,12)
SHEL 637 C      COMMON /TRANSF/ T1(3),T2(3),T3(3), TO(3,3)
SHEL 638 C      DIMENSION F(11), IPERMQ(4), MFR(5), LOC(5), NC(3), CA(3), WGT(3),
SHEL 639 C      1 TD1(13),TD2(13),TD3(9), TR1(9),TR2(9),TR3(9), U(1),V(1),W(1),
SHEL 640 C      2 RX(1),RY(1)
SHEL 641 C      EQUIVALENCE (T11,T1),(T12,T1(2)),(T13,T1(3)),(T21,T2),(T22,T2(2)),
SHEL 642 C      1 (T23,T2(3)),(T31,T3),(T32,T3(2)),(T33,T3(3)), (R,F),
SHEL 643 C      2 (U,FT), (V,FT(7)), (W,P1), (RX,P2), (RY,P3)

```

```

SHEL 644      DATA IPERMQ /2,3,4,1/, MFR /3,3,3,2/, WGT /50,50,.25/
SHEL 645      LOGICAL QUAD, TRIG, NOMP, NOST, SIST, NOLD, SILO, NOSM, SISM, SKMP
SHEL 646 C
SHEL 647 C      INITIALIZE
SHEL 648 C
SHEL 649      SIST = KKK.LE.0
SHEL 650      NOST = .NOT.SIST
SHEL 651      SILO = KKK.EQ.0.CR.KKK.EQ.1
SHEL 652      NOLD = .NOT.SILO
SHEL 653      SISM = KKK.GE.2
SHEL 654      NOSM = .NOT.SISM
SHEL 655      IF ((NNS.NE.3).AND.(NNS.NE.5)) NNS = 4
SHEL 656      IF ((NPF.NE.3).AND.(NPF.NE.6)) NPF = 5
SHEL 657      NEN = MINO (NNS,4)
SHEL 658      QUAD = NEN.EQ.4
SHEL 659      TRIG = NEN.EQ.3
SHEL 660      WG = 1.
SHEL 661      N3 = 2*NEN - 3
SHEL 662      NTRI = 3*NEN - 8
SHEL 663      NEF = NEN*NPF
SHEL 664      NSF = NEF + (NEN-3)*NPF
SHEL 665      IF (MID.NE.4) MID = 0
SHEL 666      MIDP = MID
SHEL 667      IF (TRIG) MIDP = 0
SHEL 668      NFM = 3
SHEL 669      IF (NPF.EQ.3) NFM = 2
SHEL 670      NTF = NSF + NFM*MIDP
SHEL 671      NOMP = MIDP.LE.0
SHEL 672      SKMP = NOMP.CR.NOST
SHEL 673      IF (NNS.NE.4) GO TO 130
SHEL 674      X(5) = 0.25*(X(1)+X(2)+X(3)+X(4))
SHEL 675      Y(5) = 0.25*(Y(1)+Y(2)+Y(3)+Y(4))
SHEL 676      Z(5) = 0.25*(Z(1)+Z(2)+Z(3)+Z(4))
SHEL 677      HM(5) = 0.25*(HM(1)+HM(2)+HM(3)+HM(4))
SHEL 678      HP(5) = 0.25*(HP(1)+HP(2)+HP(3)+HP(4))
SHEL 679      IF (KKK.LT.0) GO TO 130
SHEL 680      T(5) = 0.25*(T(1)+T(2)+T(3)+T(4))
SHEL 681      DT(5) = 0.25*(DT(1)+DT(2)+DT(3)+DT(4))
SHEL 682      DO 110 J = 1,3
SHEL 683      SM(5,J) = 0.25*(SM(1,J)+SM(2,J)+SM(3,J)+SM(4,J))
SHEL 684      110 BM(5,J) = 0.25*(BM(1,J)+BM(2,J)+BM(3,J)+BM(4,J))
SHEL 685      IF (NOLD) GO TO 130
SHEL 686      P(5) = 0.25*(P(1)+P(2)+P(3)+P(4))
SHEL 687      HW(5) = 0.25*(HW(1)+HW(2)+HW(3)+HW(4))
SHEL 688      DO 120 J = 1,3
SHEL 689      120 RHO(5,J) = 0.25*(RHO(1,J)+RHO(2,J)+RHO(3,J)+RHO(4,J))
SHEL 690      130 IF (NOST) GO TO 150
SHEL 691      DO 140 I = 1,NTF
SHEL 692      DO 140 J = 1,NTF
SHEL 693      140 S(I,J) = 0.
SHEL 694      150 IF (SISM) GO TO 170
SHEL 695      DO 160 I = 1,NTF
SHEL 696      160 F(I) = 0.
SHEL 697      170 IF (NOSM.OR.TRIG) GO TO 200
SHEL 698      NEF1 = NEF + 1
SHEL 699      DO 180 L = NEF1,NTF
SHEL 700      M = L - 1
SHEL 701      DO 180 I = 1,M
SHEL 702      180 R(L) = R(L) - S(I,L)*R(I)
SHEL 703      200 DO 210 I = 1,63
SHEL 704      210 A(I) = 0.
SHEL 705      DO 220 I = 1,3
SHEL 706      CA(I) = CM(1,I)*ALFA(1)+CM(2,I)*ALFA(2)+CM(3,I)*ALFA(3)
SHEL 707      DO 220 J = 1,3
SHEL 708      220 C(I,J) = CM(I,J)
SHEL 709 C
SHEL 710 C      COMPUTE DIRECTION COSINE MATRIX TO OF LOCAL ELEMENT SYSTEM
SHEL 711 C
SHEL 712 C      CALL QDCOS (NTRI,X,Y,Z,TO)
SHEL 713 C
SHEL 714 C      LOOP OVER THE NTRI TRIANGLE COMPONENTS
SHEL 715 C
SHEL 716      DO 700 NT = 1,NTRI
SHEL 717      N1 = NT
SHEL 718      N2 = IPERMQ(N1)

```

```

SHEL 719 NC(1) = N1
SHEL 720 NC(2) = N2
SHEL 721 NC(3) = N3
SHEL 722 MT = MIDP/2
SHEL 723 NOD = 3 + MT
SHEL 724 C
SHEL 725 C COMPUTE DIRECTION COSINES OF LOCAL TRIANGLE SYSTEM
SHEL 726 C AND THE TRIANGLE PROJECTIONS A,B ONTO IT
SHEL 727 C
SHEL 728 CALL TOCOS (N1,N2,N3,X,Y,Z,A,B)
SHEL 729 C
SHEL 730 C SET UP INPUTS FOR TRIANGLE SUBROUTINES
SHEL 731 C
SHEL 732 DO 240 I = 1,3
SHEL 733 L = NC(I)
SHEL 734 LOC(I) = NPF*(L-1)
SHEL 735 HMT(I) = HM(L)
SHEL 736 HP(L) = HP(L)
SHEL 737 IF (NOLD) GO TO 240
SHEL 738 ROX = RHO(L,1)
SHEL 739 ROY = RHO(L,2)
SHEL 740 ROZ = RHO(L,3)
SHEL 741 RO1 = T11*ROX+T12*ROY+T13*ROZ
SHEL 742 RO2 = T21*ROX+T22*ROY+T23*ROZ
SHEL 743 RO3 = T31*ROX+T32*ROY+T33*ROZ
SHEL 744 H1 = HW(L)
SHEL 745 P1(I) = RO1*H1
SHEL 746 P2(I) = RO2*H1
SHEL 747 P3(I) = RO3*H1 + P(L)
SHEL 748 TEMP = T(L)
SHEL 749 TMOM = DT(L)*HP(L)**3/12.
SHEL 750 DO 230 J = 1,3
SHEL 751 SMT(I,J) = SM(L,J) - CA(J)*TEMP
SHEL 752 230 SMT(I,J) = SM(L,J) - CA(J)*TMOM
SHEL 753 240 CONTINUE
SHEL 754 C
SHEL 755 C FORM TRANSFORMATIONS BETWEEN ELEMENT AND NODAL SYSTEMS
SHEL 756 C
SHEL 757 L1 = 9*N1 - 8
SHEL 758 L2 = 9*N2 - 8
SHEL 759 CALL TRFPRD (IDIS,NEN,TDIS(L1),TDIS(L2),TDIS(L9),T01,T02,T03)
SHEL 760 IF (NPF.NE.3)
SHEL 761 ICALL TRFPRD (IROT,NEN,TROT(L1),TROT(L2),TROT(L9),TR1,TR2,TR3)
SHEL 762 DO 250 I = 7,8
SHEL 763 TD1(I+3) = T01(I)
SHEL 764 TD1(I+5) = T01(I)
SHEL 765 TD2(I+3) = T02(I)
SHEL 766 250 TD2(I+5) = T02(I)
SHEL 767 LOC(4) = NSF + NFM*(N2-1)
SHEL 768 LOC(5) = NSF + NFM*(N1-1)
SHEL 769 N4 = LOC(4) + 3
SHEL 770 N5 = LOC(5) + 3
SHEL 771 C
SHEL 772 C MEMBRANE CONTRIBUTION
SHEL 773 C
SHEL 774 260 IF (SISM) GO TO 320
SHEL 775 C MEMBRANE STIFFNESS AND/OR LOAD VECTOR
SHEL 776 CALL SLST (MT,KKK)
SHEL 777 LT = 0
SHEL 778 DO 300 JJ = 1,NOD
SHEL 779 J = JJ + JJ
SHEL 780 M = LOC(JJ)
SHEL 781 LL = MFR(JJ)
SHEL 782 DO 300 L = 1,LL
SHEL 783 M = M + 1
SHEL 784 LT = LT + 1
SHEL 785 C1 = TD1(LT)
SHEL 786 C2 = TD2(LT)
SHEL 787 IF (SILD) F(M) = F(M) + FTIJ-1)*C1 + FT(J)*C2
SHEL 788 IF (NOST) GO TO 300
SHEL 789 KT = 0
SHEL 790 DO 290 II = 1,JJ
SHEL 791 I = II + II
SHEL 792 KK = MFR(II)
SHEL 793 IF (II.EQ.JJ) KK = L

```

```

SHEL 794 H1 = ST(I-1,J-1)*C1 + ST(I-1,J)*C2
SHEL 795 H2 = ST(I ,J-1)*C1 + ST(I ,J)*C2
SHEL 796 N = LOC(II)
SHEL 797 N = LOC(II)
SHEL 798 DO 290 K = 1,KK
SHEL 799 N = N + 1
SHEL 800 KT = KT + 1
SHEL 801 SQ = S(N,M) + T01(KT)*H1 + T02(KT)*H2
SHEL 802 S(N,M) = SQ
SHEL 803 290 CONTINUE
SHEL 804 300 CONTINUE
SHEL 805 320 CONTINUE
SHEL 806 400 IF (NPF.EQ.3) GO TO 560
SHEL 807 C
SHEL 808 C PLATE BENDING CONTRIBUTION
SHEL 809 C
SHEL 810 C IF (SISM) GO TO 600
SHEL 811 C BENDING STIFFNESS AND/OR LOAD VECTOR
SHEL 812 CALL SLCT (MT,KKK)
SHEL 813 DO 500 JJ = 1,3
SHEL 814 JT = 3*JJ-3
SHEL 815 J = JT + 1
SHEL 816 DO 450 L = 1,NPF
SHEL 817 M = LOC(JJ) + L
SHEL 818 L3 = L - 3
SHEL 819 IF (L3.GT.0) GO TO 420
SHEL 820 C3 = T03(JT+L)
SHEL 821 IF (SILD) F(M) = F(M) + FT(J)*C3
SHEL 822 IF (SKMP) GO TO 450
SHEL 823 S4 = S(M,N4) + ST(J,10)*C3
SHEL 824 S5 = S(M,N5) - ST(J,11)*C3
SHEL 825 GO TO 430
SHEL 826 420 C1 = TR1(JT+L3)
SHEL 827 C2 = TR2(JT+L3)
SHEL 828 IF (SILD) F(M) = F(M) + FT(J+1)*C1 + FT(J+2)*C2
SHEL 829 IF (SKMP) GO TO 450
SHEL 830 S4 = S(M,N4) + ST(J+1,10)*C1 + ST(J+2,10)*C2
SHEL 831 S5 = S(M,N5) - ST(J+1,11)*C1 - ST(J+2,11)*C2
SHEL 832 430 S(M,N4) = S4
SHEL 833 S(M,N5) = S5
SHEL 834 S(N5,M) = S5
SHEL 835 450 CONTINUE
SHEL 836 IF (NOST) GO TO 500
SHEL 837 DO 480 II = 1,JJ
SHEL 838 IT = 3*II-3
SHEL 839 I = IT + 1
SHEL 840 KK = NPF
SHEL 841 DO 480 L = 1,NPF
SHEL 842 IF (II.EQ.JJ) KK = L
SHEL 843 M = LOC(JJ) + L
SHEL 844 L3 = L - 3
SHEL 845 IF (L3.GT.0) GO TO 460
SHEL 846 C3 = T03(JT+L)
SHEL 847 H1 = ST(I ,J)*C3
SHEL 848 H2 = ST(I+1,J)*C3
SHEL 849 H3 = ST(I+2,J)*C3
SHEL 850 GO TO 470
SHEL 851 460 C1 = TR1(JT+L3)
SHEL 852 C2 = TR2(JT+L3)
SHEL 853 H1 = ST(I ,J+1)*C1 + ST(I ,J+2)*C2
SHEL 854 H2 = ST(I+1,J+1)*C1 + ST(I+1,J+2)*C2
SHEL 855 H3 = ST(I+2,J+1)*C1 + ST(I+2,J+2)*C2
SHEL 856 470 N = LOC(II)
SHEL 857 DO 480 K = 1,KK
SHEL 858 N = N + 1
SHEL 859 K3 = K - 3
SHEL 860 K1 = IT + K
SHEL 861 K2 = IT + K3
SHEL 862 IF (K3.LE.0) SQ = S(N,M) + T03(K1)*H1
SHEL 863 IF (K3.GT.0) SQ = S(N,M) + TR1(K2)*H2 + TR2(K2)*H3
SHEL 864 S(N,M) = SQ
SHEL 865 480 S(M,N) = SQ
SHEL 866 500 CONTINUE
SHEL 867 IF (NOMP) GO TO 700

```

```

SHEL 868      IF (NOLD) GO TO 540
SHEL 869      F(N4) = F(N4) + FT(10)
SHEL 870      F(N5) = F(N5) - FT(11)
SHEL 871      540 IF (NOST) GO TO 700
SHEL 872      S(N4,N4) = S(N4,N4) + ST(10,10)
SHEL 873      S(N5,N5) = S(N5,N5) + ST(11,11)
SHEL 874      S(N4,N5) = S(N4,N5) - ST(10,11)
SHEL 875      S(N5,N4) = S(N4,N5)
SHEL 876      GO TO 700
SHEL 877      560 IF (NOLD) GO TO 700
SHEL 878      FL = (P3(1)+P3(2)+P3(3))*(A(3)*B(2)-A(2)*B(3))/12.
SHEL 879      JT = 0
SHEL 880      DO 580 JJ = 1,2
SHEL 881      DO 580 L=1,3
SHEL 882      JT=JT+1
SHEL 883      M = LOC(JJ) + L
SHEL 884      580 F(M) = F(M) + FL*TD3(JT)
SHEL 885      600 CONTINUE
SHEL 886      700 CONTINUE
SHEL 887      IF (SISM.OR.TRIG) GO TO 900
SHEL 888      C
SHEL 889      C CHECK FOR POSSIBLE INTERNAL STIFFNESS SINGULARITY (FLAT
SHEL 890      C OR NEARLY FLAT QUADRILATERAL WHEN NPF = 3 OR 6)
SHEL 891      IF ((NPF.EQ.5).OR.NOST) GO TO 730
SHEL 892      DO 720 N = 3,6,3
SHEL 893      IF (NPF.NE.N) GO TO 720
SHEL 894      M = 5*N
SHEL 895      M1 = M - 1
SHEL 896      M2 = M - 2
SHEL 897      IF (S(M,M).GT.(S(M1,M1)+S(M2,M2))*1.0E-08) GO TO 720
SHEL 898      DO 710 I = 1,NTF
SHEL 899      S(I,M) = 0.
SHEL 900      710 S(M,I) = 0.
SHEL 901      720 CONTINUE
SHEL 902      C
SHEL 903      C CONDENSATION OF INTERNAL DEGREES OF FREEDOM
SHEL 904      C
SHEL 905      730 NIF = NTF - NEF
SHEL 906      DO 800 N = 1,NIF
SHEL 907      K = NTF - N
SHEL 908      L = K + 1
SHEL 909      PIVOT = S(L,L)
SHEL 910      FL = F(L)
SHEL 911      IF (PIVOT.LE.0.) GO TO 800
SHEL 912      F(L) = F(L)/PIVOT
SHEL 913      DO 780 I = 1,K
SHEL 914      G = S(I,L)
SHEL 915      IF (G) 740,780,740
SHEL 916      740 IF (NOST) GO TO 770
SHEL 917      G = G/PIVOT
SHEL 918      S(I,L) = G
SHEL 919      DO 760 J = I,K
SHEL 920      SQ = S(I,J) - G*S(L,J)
SHEL 921      S(I,J) = SQ
SHEL 922      760 S(J,I) = SQ
SHEL 923      770 F(I) = F(I) - G*FL
SHEL 924      780 CONTINUE
SHEL 925      800 CONTINUE
SHEL 926      900 RETURN
SHEL 927      END

SHEL 928      SUBROUTINE QDCOS (N,X,Y,Z,T)
SHEL 929      C
SHEL 930      C THIS SUBROUTINE COMPUTES THE DIRECTION COSINES OF THE LOCAL
SHEL 931      C ELEMENT SYSTEM OF A QUADRILATERAL (N=4) OR SINGLE TRIANGLE (N=1)
SHEL 932      C
SHEL 933      DIMENSION X(1), Y(1), Z(1), T(1)
SHEL 934      X1 = X(2)+X(3)-X(N)-X(1)
SHEL 935      Y1 = Y(2)+Y(3)-Y(N)-Y(1)

```

```

SHEL 936      Z1 = Z(2)+Z(3)-Z(N)-Z(1)
SHEL 937      X2 = X(3)+X(N)-X(1)-X(2)
SHEL 938      Y2 = Y(3)+Y(N)-Y(1)-Y(2)
SHEL 939      Z2 = Z(3)+Z(N)-Z(1)-Z(2)
SHEL 940      S1 = X1**2+Y1**2+Z1**2
SHEL 941      C = (X1*X2+Y1*Y2+Z1*Z2)/S1
SHEL 942      X2 = X2 - C*X1
SHEL 943      Y2 = Y2 - C*Y1
SHEL 944      Z2 = Z2 - C*Z1
SHEL 945      S1 = SQRT (S1)
SHEL 946      S2 = SQRT (X2**2+Y2**2+Z2**2)
SHEL 947      X1 = X1/S1
SHEL 948      Y1 = Y1/S1
SHEL 949      Z1 = Z1/S1
SHEL 950      X2 = X2/S2
SHEL 951      Y2 = Y2/S2
SHEL 952      Z2 = Z2/S2
SHEL 953      T(1) = X1
SHEL 954      T(2) = X2
SHEL 955      T(3) = Y1*Z2-Y2*Z1
SHEL 956      T(4) = Y1
SHEL 957      T(5) = Y2
SHEL 958      T(6) = Z1*X2-Z2*X1
SHEL 959      T(7) = Z1
SHEL 960      T(8) = Z2
SHEL 961      T(9) = X1*Y2-X2*Y1
SHEL 962      RETURN
SHEL 963      END

```

```

SHEL 964      SUBROUTINE TDCOS (N1,N2,N3,X,Y,Z,A,B)
SHEL 965      C
SHEL 966      C THIS SUBROUTINE COMPUTES THE DIRECTION COSINES OF THE LOCAL
SHEL 967      C SYSTEM AND THE PROJECTED DIMENSIONS OF A TRIANGLE COMPONENT
SHEL 968      C
SHEL 969      COMMON /TRANSF/ T1(3),T2(3),T3(3), T(9)
SHEL 970      EQUIVALENCE (T1,T1),(T2,T1(2)),(T3,T1(3)),(T21,T2),(T22,T2(2)),
SHEL 971      1 (T23,T2(3)),(T31,T3),(T32,T3(2)),(T33,T3(3))
SHEL 972      DIMENSION X(1), Y(1), Z(1), A(1), B(1)
SHEL 973      A1 = X(N1)-X(N3)
SHEL 974      B1 = Y(N1)-Y(N3)
SHEL 975      C1 = Z(N1)-Z(N3)
SHEL 976      A2 = X(N2)-X(N3)
SHEL 977      B2 = Y(N2)-Y(N3)
SHEL 978      C2 = Z(N2)-Z(N3)
SHEL 979      T31 = B1*C2-B2*C1
SHEL 980      T32 = C1*A2-C2*A1
SHEL 981      T33 = A1*B2-A2*B1
SHEL 982      S = SQRT (T31**2+T32**2+T33**2)
SHEL 983      T31 = T31/S
SHEL 984      T32 = T32/S
SHEL 985      T33 = T33/S
SHEL 986      Y11 = T33*T(5)-T32*T(8)
SHEL 987      T12 = T31*T(8)-T33*T(2)
SHEL 988      T13 = T32*T(2)-T31*T(5)
SHEL 989      S = SQRT (T11**2+T12**2+T13**2)
SHEL 990      T11 = T11/S
SHEL 991      T12 = T12/S
SHEL 992      T13 = T13/S
SHEL 993      Y21 = T13*T32-T12*T33
SHEL 994      T22 = T11*T33-T13*T31
SHEL 995      T23 = T12*T31-T11*T32
SHEL 996      A(1) = -T11*A2-T12*B2-T13*C2
SHEL 997      A(2) = T11*A1+T12*B1+T13*C1
SHEL 998      B(1) = T21*A2+T22*B2+T23*C2
SHEL 999      B(2) = -T21*A1-T22*B1-T23*C1
SHEL1000      A(3) = -A(1)-A(2)
SHEL1001      B(3) = -B(1)-B(2)
SHEL1002      RETURN
SHEL1003      END

```

```

SHEL1004 SUBROUTINE TRFPRD (M,NEN,Q1,Q2,Q3,P1,P2,P3)
SHEL1005 C
SHEL1006 C THIS SUBROUTINE GENERATES THE TRANSFORMATIONS RELATING A LOCAL
SHEL1007 C SUBTRIANGLE SYSTEM TO THE NODAL DIS/ROT SYSTEMS AT ITS 3 CORNERS
SHEL1008 C
SHEL1009 C COMMON /TRANS/ T1(3),T2(3),T3(3), T(9)
SHEL1010 C DIMENSION P1(1), P2(1), P3(1), Q1(1), Q2(1), Q3(1)
SHEL1011 C EQUIVALENCE (T11,T11),(T12,T1(2)),(T13,T1(3)),(T21,T2), (T22,T2(2)),
SHEL1012 C 1 (T23,T2(3)),(T31,T3), (T32,T3(2)),(T33,T3(3))
SHEL1013 C DO 300 I = 1,3
SHEL1014 C J = I + 3
SHEL1015 C K = I + 6
SHEL1016 C P1(I) = T1(I)
SHEL1017 C P1(J) = T1(I)
SHEL1018 C P2(I) = T2(I)
SHEL1019 C P2(J) = T2(I)
SHEL1020 C P3(I) = T3(I)
SHEL1021 C P3(J) = T3(I)
SHEL1022 C IF (NEN.NE.4) GO TO 150
SHEL1023 C CI = T(I)
SHEL1024 C CJ = T(J)
SHEL1025 C CK = T(K)
SHEL1026 C IF (M) 260,260,240
SHEL1027 C 150 IF (M) 180,180,200
SHEL1028 C 180 P1(K) = T1(I)
SHEL1029 C P2(K) = T2(I)
SHEL1030 C P3(K) = T3(I)
SHEL1031 C GO TO 300
SHEL1032 C 200 CI = Q3(I)
SHEL1033 C CJ = Q3(J)
SHEL1034 C CK = Q3(K)
SHEL1035 C 240 P1(I) = T11*Q1(I) + T12*Q1(J) + T13*Q1(K)
SHEL1036 C P1(J) = T11*Q2(I) + T12*Q2(J) + T13*Q2(K)
SHEL1037 C P2(I) = T21*Q1(I) + T22*Q1(J) + T23*Q1(K)
SHEL1038 C P2(J) = T21*Q2(I) + T22*Q2(J) + T23*Q2(K)
SHEL1039 C P3(I) = T31*Q1(I) + T32*Q1(J) + T33*Q1(K)
SHEL1040 C P3(J) = T31*Q2(I) + T32*Q2(J) + T33*Q2(K)
SHEL1041 C 260 P1(K) = T11*CI + T12*CJ + T13*CK
SHEL1042 C P2(K) = T21*CI + T22*CJ + T23*CK
SHEL1043 C P3(K) = T31*CI + T32*CJ + T33*CK
SHEL1044 C 300 CONTINUE
SHEL1045 C RETURN
SHEL1046 C END

```

```

SHEL1047 SUBROUTINE SLST (M,KKK)
SHEL1048 C
SHEL1049 C THIS SUBROUTINE FORMS THE PLANE STRESS STIFFNESS MATRIX AND/OR
SHEL1050 C THE CONSISTENT LOAD VECTOR OF A LINEAR STRAIN TRIANGLE (LST) WITH
SHEL1051 C 6, 5 OR 4 NODAL POINTS, OR OF A CONSTANT STRAIN TRIANGLE (CST).
SHEL1052 C LINEAR ELASTIC ANISOTROPIC MATERIAL
SHEL1053 C
SHEL1054 C ***** INPUTS *****
SHEL1055 C
SHEL1056 C M NUMBER OF MIDPOINTS INCLUDED AS NODAL POINTS (M=3,2,1
SHEL1057 C FOR LST, M=0 FOR CST). NOTE.. MIDPOINTS 4-5-6 ARE
SHEL1058 C LOCATED ON THE SIDES 2-3, 3-1 AND 1-2, RESPECTIVELY.
SHEL1059 C
SHEL1060 C KKK OPERATION FLAG
SHEL1061 C KKK LE 0 = FORM STIFFNESS MATRIX AND LOAD VECTOR.
SHEL1062 C KKK GT 0 = FORM LOAD VECTOR ONLY.
SHEL1063 C
SHEL1064 C A(I),B(I) I=1...3 PROJECTIONS OF SIDES 2-3, 3-1 AND 1-2 ONTO
SHEL1065 C X AND -Y, RESPECTIVELY.
SHEL1066 C
SHEL1067 C C(I,J) I=1...3, J=1...3 PLANE STRESS MATERIAL MATRIX.
SHEL1068 C
SHEL1069 C H(I) I=1...3 CORNER THICKNESSES (LINEAR VARIATION ASSUMED).
SHEL1070 C
SHEL1071 C PX(I),PY(I) I=1...3 CORNER VALUES OF X-Y COMPONENTS OF BODY +ORCES

```

```

SHEL1072 C PER UNIT OF ELEMENT AREA (LINEAR VARIATION ASSUMED).
SHEL1073 C
SHEL1074 C SMT(I,J) I=1...3, J=1...3 INITIAL MEMBRANE STRESS COMPONENTS
SHEL1075 C SIG-XX (J=1), SIG-YY (J=2) AND SIG-XY (J=3) AT THE
SHEL1076 C CORNERS I=1,2,3 (LINEAR VARIATION ASSUMED).
SHEL1077 C
SHEL1078 C ***** OUTPUTS *****
SHEL1079 C
SHEL1080 C
SHEL1081 C ST(I,J) I=1..NDF, J=1..NDF WITH NDF (NUMBER OF DOF) = 6+2*M, IS
SHEL1082 C THE ELEMENT STIFFNESS MATRIX ASSOCIATED WITH THE NODAL
SHEL1083 C DISPLACEMENT ORDERING
SHEL1084 C U(1),V(1),U(2),V(2),U(3), .... V(3+M)
SHEL1085 C WHERE U(I), ... V(3+M), IF M GT 0, ARE DEVIATIONS
SHEL1086 C FROM LINEARITY AT THE MIDPOINTS 1...M.
SHEL1087 C
SHEL1088 C FT(I) I=1..NDF CONSISTENT NODAL FORCE VECTOR ASSOCIATED
SHEL1089 C WITH THE NODAL DISPLACEMENT ORDERING DESCRIBED ABOVE.
SHEL1090 C
SHEL1091 C
SHEL1092 C COMMON /TRIARG/ A(3),B(3), H(3), HPT(3), C(3,3), SMT(3,3),
SHEL1093 C 1 BMT(3,3), FT(12), PX(3),PY(3),PT(3),RM(3), ST(12,12)
SHEL1094 C DIMENSION Q(3,3), QA(3), QB(3), A4(3), B4(3), IPERM(3),
SHEL1095 C 1 SX(3),SY(3), SXV(3)
SHEL1096 C EQUIVALENCE (SXX,SMT),(SYY,SMT(4)),(SXY,SMT(7))
SHEL1097 C LOGICAL NOS
SHEL1098 C DATA IPERM /2,3,1/
SHEL1099 C NOS = KKK.GT.0
SHEL1100 C NDF = 6 + 2*M
SHEL1101 C AREA = A(3)*B(2)-A(2)*B(3)
SHEL1102 C SUMH = H(1)+H(2)+H(3)
SHEL1103 C HO = SUMH/3.
SHEL1104 C IF (HO) 500,500,140
SHEL1105 C 140 PXS = PX(1)+PX(2)+PX(3)
SHEL1106 C PYS = PY(1)+PY(2)+PY(3)
SHEL1107 C SXXH = 0.
SHEL1108 C SYXH = 0.
SHEL1109 C SKXH = 0.
SHEL1110 C DO 150 I = 1,3
SHEL1111 C CH = (SUMH+H(I))/24.
SHEL1112 C SXXH = SXXH + CH*SXX(I)
SHEL1113 C SYXH = SYXH + CH*SYX(I)
SHEL1114 C 150 SKXH = SKXH + CH*SKX(I)
SHEL1115 C FAC = HO/(2.*AREA)
SHEL1116 C C11 = C(1,1)*FAC
SHEL1117 C C22 = C(2,2)*FAC
SHEL1118 C C33 = C(3,3)*FAC
SHEL1119 C C12 = C(1,2)*FAC
SHEL1120 C C13 = C(1,3)*FAC
SHEL1121 C C23 = C(2,3)*FAC
SHEL1122 C DO 200 J = 1,3
SHEL1123 C L = J + J
SHEL1124 C FY(L-1) = (PXS+PX(J))*AREA/24. - (B(J)*SXXH+A(J)*SXYH)
SHEL1125 C FT(L) = (PYS+PY(J))*AREA/24. - (A(J)*SYXH+B(J)*SKXH)
SHEL1126 C IF (NOS) GO TO 200
SHEL1127 C 180 DO 190 I = 1,J
SHEL1128 C K = I + I
SHEL1129 C AA = A(I)*A(J)
SHEL1130 C BB = B(I)*B(J)
SHEL1131 C AB = A(I)*B(J)
SHEL1132 C BA = B(I)*A(J)
SHEL1133 C ABA = AB*BA
SHEL1134 C ST(K-1,L-1) = C11*BB + C33*AA + C13*ABA
SHEL1135 C ST(K ,L ) = C22*AA + C33*BB + C23*ABA
SHEL1136 C ST(K-1,L ) = C12*BA + C33*AB + C13*AA + C23*BB
SHEL1137 C 190 ST(K ,L-1) = C12*AB + C33*BA + C13*BB + C23*AA
SHEL1138 C 200 CONTINUE
SHEL1139 C IF (M) 350,350,220
SHEL1140 C 220 DO 240 I = 1,3
SHEL1141 C A4(I) = 4.*A(I)
SHEL1142 C B4(I) = 4.*B(I)
SHEL1143 C J = IPERM(I)
SHEL1144 C K = IPERM(J)
SHEL1145 C R = H(I)/HO
SHEL1146 C Q(I,I) = 0.1+R/15.

```

```

SHEL1147 Q(J,K) = 0.1-R/60.
SHEL1148 240 Q(K,J) = Q(J,K)
SHEL1149 DO 300 J = 1,M
SHEL1150 J1 = IPERM(J)
SHEL1151 J2 = IPERM(J1)
SHEL1152 L = J + J + 6
SHEL1153 FX = 0.
SHEL1154 FY = 0.
SHEL1155 DO 250 N = 1,3
SHEL1156 Q1 = Q(N,J1)
SHEL1157 Q2 = Q(N,J2)
SHEL1158 QA(N) = Q2*A4(J1)+Q1*A4(J2)
SHEL1159 QB(N) = Q2*B4(J1)+Q1*B4(J2)
SHEL1160 FX = FX - QB(N)*SXX(N)-QA(N)*SXY(N)
SHEL1161 250 FY = FY - QA(N)*SYY(N)-QB(N)*SXY(N)
SHEL1162 FT(L-1) = (PXS-PX(J))*AREA/12. + FX*HO/2.
SHEL1163 FT(L) = (PYS-PY(J))*AREA/12. + FY*HO/2.
SHEL1164 IF (NOS) GO TO 300
SHEL1165 SUMQA = QA(1)+QA(2)+QA(3)
SHEL1166 SUMQB = QB(1)+QB(2)+QB(3)
SHEL1167 JM = J + 3
SHEL1168 DO 290 I = 1,JM
SHEL1169 K = I + 1
SHEL1170 IF (I.GT.3) GO TO 260
SHEL1171 AA = A(I)*SUMQA
SHEL1172 AB = A(I)*SUMQB
SHEL1173 BA = B(I)*SUMQA
SHEL1174 BB = B(I)*SUMQB
SHEL1175 GO TO 280
SHEL1176 260 I1 = IPERM(I-3)
SHEL1177 I2 = IPERM(I1)
SHEL1178 AA = A4(I2)*QA(I1)+A4(I1)*QA(I2)
SHEL1179 AB = A4(I2)*QB(I1)+A4(I1)*QB(I2)
SHEL1180 BA = B4(I2)*QA(I1)+B4(I1)*QA(I2)
SHEL1181 BB = B4(I2)*QB(I1)+B4(I1)*QB(I2)
SHEL1182 280 ABA = AB+BA
SHEL1183 ST(K-1,L-1) = C11*BB + C33*AA + C13*ABA
SHEL1184 ST(K ,L ) = C22*AA + C33*BB + C23*ABA
SHEL1185 ST(K-1,L ) = C12*BA + C33*AB + C13*AA + C23*BB
SHEL1186 290 ST(K ,L-1) = C12*AB + C33*BA + C13*BB + C23*AA
SHEL1187 300 CONTINUE
SHEL1188 350 DO 400 I = 2,NDF
SHEL1189 DO 400 J = 1,I
SHEL1190 400 ST(I,J) = ST(J,I)
SHEL1191 500 RETURN
SHEL1192 END

```

```

SHEL1193 SUBROUTINE SLCC (M,KKK)
SHEL1194 C
SHEL1195 C THIS SUBROUTINE FORMS THE PLATE BENDING STIFFNESS AND/OR THE
SHEL1196 C CONSISTENT LOAD VECTOR OF A LINEAR CURVATURE COMPATIBLE TRIANGLE
SHEL1197 C (LCC) WITH 6, 5, 4 OR 3 NODAL POINTS
SHEL1198 C
SHEL1199 C
SHEL1200 C ***** INPUTS *****
SHEL1201 C
SHEL1202 C M NUMBER OF MIDPOINT DEGREES OF FREEDOM (M =3,2,1,0).
SHEL1203 C NOTE.. MIDPOINTS 4-5-6 (IF INCLUDED) ARE LOCATED ON
SHEL1204 C SIDES 2-3, 3-1 AND 1-2, RESPECTIVELY.
SHEL1205 C
SHEL1206 C KKK OPERATION FLAG
SHEL1207 C KKK LE 0 = FORM STIFFNESS MATRIX AND LOAD VECTOR.
SHEL1208 C KKK GT 0 = FORM LOAD VECTOR ONLY.
SHEL1209 C
SHEL1210 C A(I),B(I) I=1...3 PROJECTIONS OF SIDES 2-3, 3-1 AND 1-2 ONTO
SHEL1211 C X AND -Y, RESPECTIVELY.
SHEL1212 C
SHEL1213 C C(I,J) I=1...3, J=1...3 PLANE STRESS MATERIAL MATRIX.
SHEL1214 C

```

```

SHEL1215 C H(I) I=1...3 CORNER THICKNESSES (LINEAR VARIATION ASSUMED).
SHEL1216 C
SHEL1217 C PT(I) I=1...3 CORNER VALUES OF LATERAL DISTRIBUTED LOAD
SHEL1218 C (LINEAR VARIATION ASSUMED).
SHEL1219 C
SHEL1220 C BMT(I,J) I=1...3, J=1...3 INITIAL BENDING MOMENT COMPONENTS
SHEL1221 C MCM-XX (J=1), MOM-YY (J=2) AND MOM-XY (J=3) AT THE
SHEL1222 C CORNERS I=1...3 (LINEAR VARIATION ASSUMED).
SHEL1223 C
SHEL1224 C ***** OUTPUTS *****
SHEL1225 C
SHEL1226 C
SHEL1227 C ST(I,J) I=1..NDF, J=1..NDF WITH NDF (NUMBER OF DOF) = 9+M, IS
SHEL1228 C THE ELEMENT STIFFNESS MATRIX ASSOCIATED WITH THE NODAL
SHEL1229 C DISPLACEMENT ORDERING
SHEL1230 C W(1),RX(1),RY(1),W(2), ... RY(3),RM(1), ... RM(M)
SHEL1231 C WHERE RM(1), ... RM(M), IF M GT 0, ARE MIDPOINT
SHEL1232 C DEVIATIONS FROM NORMAL SLOPE LINEARITY
SHEL1233 C
SHEL1234 C FT(I) I=1..NDF CONSISTENT NODAL FORCE VECTOR ASSOCIATED
SHEL1235 C WITH THE NODAL DISPLACEMENT ORDERING DESCRIBED ABOVE.
SHEL1236 C
SHEL1237 C
SHEL1238 C COMMON /TRTARG/ A(3),B(3), HMT(3), H(3), C(3,3), SMT(3,3),
SHEL1239 C I BMT(3,3), FT(12), PX(3),PY(3),PT(3),RM(3), ST(12,12)
SHEL1240 C DIMENSION P(21,12), G(21), Q(3,6), QB(3,6), T(3), U(3), HT(3),
SHEL1241 C I TX(3), TY(3), IPERM(3), XM(3,3), XMO(3)
SHEL1242 C EQUIVALENCE (CML1,C(1)),(CM12,C(2)),(CM13,C(3)),(CM22,C(5)),
SHEL1243 C I (CM23,C(6)),(CM33,C(9))
SHEL1244 C LOGICAL NOS, FLAT
SHEL1245 C DATA IPERM/2,3,1/
SHEL1246 C HO = (H(1)+H(2)+H(3))/3.
SHEL1247 C IF (HO.LE.0.) GO TO 1000
SHEL1248 C NDF = 9 + M
SHEL1249 C NOS = KKK.GT.0
SHEL1250 C FLAT = (H(1).EQ.H(2)).AND.(H(2).EQ.H(3))
SHEL1251 C AREA = A(3)*B(21)-A(2)*B(3)
SHEL1252 C FAC = HO**3*AREA/864.
SHEL1253 C PTF = AREA/6480.
SHEL1254 C T(3) = L.
SHEL1255 C DO 150 I = 1,3
SHEL1256 C J = IPERM(I)
SHEL1257 C K = IPERM(J)
SHEL1258 C X = A(I)**2+B(I)**2
SHEL1259 C U(I) = -(A(I)*A(J)+B(I)*B(J))/X
SHEL1260 C X = SQRT(X)
SHEL1261 C Y = 2.*AREA/X
SHEL1262 C HT(I) = 2.*Y
SHEL1263 C TX(I) = Y*A(I)/X
SHEL1264 C TY(I) = -Y*B(I)/X
SHEL1265 C A1 = A(I)/AREA
SHEL1266 C A2 = A(J)/AREA
SHEL1267 C B1 = B(I)/AREA
SHEL1268 C B2 = B(J)/AREA
SHEL1269 C Q(1,I) = B1*B1
SHEL1270 C Q(2,I) = A1*A1
SHEL1271 C Q(3,I) = 2.*A1*B1
SHEL1272 C Q(1,I+3) = 2.*B1*B2
SHEL1273 C Q(2,I+3) = 2.*A1*A2
SHEL1274 C Q(3,I+3) = 2.*(A1*B2+A2*B1)
SHEL1275 C DO 120 N = 1,3
SHEL1276 C 120 XM(N,I) = BMT(N,I)*AREA/72.
SHEL1277 C IF (FLAT) GO TO 150
SHEL1278 C DO 140 N = 1,3
SHEL1279 C L = IPERM(N)
SHEL1280 C T(1) = H(N)/HO
SHEL1281 C T(2) = H(L)/HO
SHEL1282 C IF (T(1).GT.0.) XM(N,I) = XM(N,I)/T(1)**3
SHEL1283 C C1 = T(1)
SHEL1284 C C2 = T(2)
SHEL1285 C C3 = T(3)
SHEL1286 C C4 = C2+C3
SHEL1287 C C11 = C1*C1
SHEL1288 C C23 = C2*C3
SHEL1289 C C5 = C4*(3.*C1+C4) + 6.*C11 - 2.*C23

```

```

SHELL1290 C6 = C5 + 3.*C4*C4 - 4.*(C11+C23)
SHELL1291 QB(N,I) = (C1*(10.*C11-3.*C23)+C4*C5)/17.5 - 2.0
SHELL1292 140 QB(N,I+3) = (C1*(C11-2.*C23)+C4*C6)/35.0 - 1.0
SHELL1293 150 CONTINUE
SHELL1294 DO 200 I = 1,3
SHELL1295 J = IPERM(I)
SHELL1296 K = IPERM(J)
SHELL1297 II = 3*I
SHELL1298 JJ = 3*J
SHELL1299 KK = 3*K
SHELL1300 A1 = A(I)
SHELL1301 A2 = A(J)
SHELL1302 A3 = A(K)
SHELL1303 B1 = B(I)
SHELL1304 B2 = B(J)
SHELL1305 B3 = B(K)
SHELL1306 U1 = U(I)
SHELL1307 U2 = U(J)
SHELL1308 U3 = U(K)
SHELL1309 W1 = 1.-U1
SHELL1310 W2 = 1.-U2
SHELL1311 W3 = 1.-U3
SHELL1312 B1D = B1 + B1
SHELL1313 B2D = B2 + B2
SHELL1314 B3D = B3 + B3
SHELL1315 A1D = A1 + A1
SHELL1316 A2D = A2 + A2
SHELL1317 A3D = A3 + A3
SHELL1318 C21 = B1-B3*U3 + TX(K)
SHELL1319 C22 = -B1D+B2*W2+B3*U3 + TX(J)-TX(K)
SHELL1320 C31 = A1-A3*U3 + TY(K)
SHELL1321 C22 = -B1D+B2*W2+B3*U3 + TX(J)-TX(K)
SHELL1322 C31 = A1-A3*U3 + TY(K)
SHELL1323 C32 = -A1D+A2*W2+A3*U3 + TY(J)-TY(K)
SHELL1324 C51 = B3*W3-B2 + TX(K)
SHELL1325 C52 = B2D-B3*W3-B1*U1 + TX(I)-TX(K)
SHELL1326 C61 = A3*W3-A2 + TY(K)
SHELL1327 C62 = A2D-A3*W3-A1*U1 + TY(I)-TY(K)
SHELL1328 C81 = B3-B2D-B2*U2 + TX(J)
SHELL1329 C82 = B1D-B3+B1*W1 + TX(I)
SHELL1330 C91 = A3-A2D-A2*U2 + TY(J)
SHELL1331 C92 = A1D-A3+A1*W1 + TY(I)
SHELL1332 P1 = PT(I)*PTF
SHELL1333 P2 = PT(J)*PTF
SHELL1334 P3 = PT(K)*PTF
SHELL1335 U37 = 7.*U3
SHELL1336 W27 = 7.*W2
SHELL1337 W24 = 4.*W2
SHELL1338 U34 = 4.*U3
SHELL1339 C1 = 54.*W27
SHELL1340 C2 = 54.*U37
SHELL1341 C3 = 15.*W24
SHELL1342 C4 = 39.*U37
SHELL1343 C5 = 39.*W27
SHELL1344 C6 = 15.*U34
SHELL1345 TKS = TX(J)+TX(K)
SHELL1346 TYS = TY(J)+TY(K)
SHELL1347 FT(II-2) = 6.*(190.+U37+W27)*P1+(136.+U37+W24)*P2+(136.+U34+W27)*P3)
SHELL1348 FT(II-1) = (C1*B2-C2*B3+7.*TKS)*P1 + (C3*B2-C4*B3+4.*TKS+
SHELL1349 13.*TX(K))*P2 + (C5*B2-C6*B3+4.*TKS+3.*TX(J))*P3
SHELL1350 FT(II) = (C1*A2-C2*A3+7.*TYS)*P1 + (C3*A2-C4*A3+4.*TYS+
SHELL1351 13.*TY(K))*P2 + (C5*A2-C6*A3+4.*TYS+3.*TY(J))*P3
SHELL1352 FT(K+9) = (7.*(P1+P2)+4.*P3)*HT(K)
SHELL1353 XMO(I) = (XM(1,I)+XM(2,I)+XM(3,I))/3.
SHELL1354 DO 200 N = 1,3
SHELL1355 L = 6*(I-1) + N
SHELL1356 Q11 = Q(N,I)
SHELL1357 Q22 = Q(N,J)
SHELL1358 Q33 = Q(N,K)
SHELL1359 Q12 = Q(N,I+3)
SHELL1360 Q23 = Q(N,J+3)
SHELL1361 Q31 = Q(N,K+3)
SHELL1362 Q2333 = Q23-Q33
SHELL1363 Q3133 = Q31-Q33
SHELL1364 P(L,II-2) = 6.*(-Q11+W2*Q33+U3*Q2333)

```

```

SHELL1365 P(L,II-1) = C21*Q23+C22*Q33-B3D*Q12+B2D*Q31
SHELL1366 P(L,II) = C31*Q23+C32*Q33-A3D*Q12+A2D*Q31
SHELL1367 P(L,II-2) = 6.*(Q22+W3*Q2333)
SHELL1368 P(L,II-1) = C51*Q2333+B3D*Q22
SHELL1369 P(L,II) = C61*Q2333+A3D*Q22
SHELL1370 P(L,II-2) = 6.*(1.+U2)*Q33
SHELL1371 P(L,II-1) = C81*Q33
SHELL1372 P(L,II) = C91*Q33
SHELL1373 P(L,II+9) = 0.
SHELL1374 P(L,II+9) = HT(J)*Q33
SHELL1375 P(L,II+9) = HT(K)*Q2333
SHELL1376 P(L+3,II-2) = 6.*(Q11+U3*Q3133)
SHELL1377 P(L+3,II-1) = C21*Q3133-B3D*Q11
SHELL1378 P(L+3,II) = C31*Q3133-A3D*Q11
SHELL1379 P(L+3,II-2) = 6.*(Q22+U1*Q33+W3*Q3133)
SHELL1380 P(L+3,II-1) = C51*Q31+C52*Q33+B3D*Q12-B1D*Q23
SHELL1381 P(L+3,II) = C61*Q31+C62*Q33+A3D*Q12-A1D*Q23
SHELL1382 P(L+3,II-2) = 6.*(1.+W1)*Q33
SHELL1383 P(L+3,II-1) = C82*Q33
SHELL1384 P(L+3,II) = C92*Q33
SHELL1385 P(L+3,II+9) = HT(I)*Q33
SHELL1386 P(L+3,II+9) = 0.
SHELL1387 P(L+3,II+9) = HT(K)*Q3133
SHELL1388 P(N+18,II-2) = 2.*(Q11+U3*Q12+W2*Q31)
SHELL1389 P(N+18,II-1) = (B1D-B2D)*Q33+C82*Q23+C81*Q31)/3.
SHELL1390 P(N+18,II) = (A1D-A2D)*Q33+C92*Q23+C91*Q31)/3.
SHELL1391 200 P(N+18,K+9) = HT(K)*Q12/3.
SHELL1392 300 DO 400 J = 1,NDF
SHELL1393 DO 340 L = 1,3
SHELL1394 II = L
SHELL1395 KK = L + 18
SHELL1396 P3 = P(KK,J)
SHELL1397 G(KK) = 0.
SHELL1398 DO 340 N = 1,3
SHELL1399 I = IPERM(N)
SHELL1400 JJ = II + 3
SHELL1401 P1 = P(II,J)
SHELL1402 P2 = P(JJ,J)
SHELL1403 SUM = P1 + P2 + P3
SHELL1404 G1 = SUM + P1
SHELL1405 G2 = SUM + P2
SHELL1406 G3 = SUM + P3
SHELL1407 IF (FLAT) GO TO 320
SHELL1408 G1 = G1 + Q8(N,1)*P1 + Q8(N,6)*P2 + Q8(N,5)*P3
SHELL1409 G2 = G2 + Q8(N,6)*P1 + Q8(N,2)*P2 + Q8(N,4)*P3
SHELL1410 G3 = G3 + Q8(N,5)*P1 + Q8(N,4)*P2 + Q8(N,3)*P3
SHELL1411 320 G(II) = G1
SHELL1412 G(JJ) = G2
SHELL1413 G(KK) = G3 + G(KK)
SHELL1414 II = II + 6
SHELL1415 340 FT(J) = FT(J) - XM(N,L)*G1 - XM(I,L)*G2 - XMO(L)*G3
SHELL1416 IF (NDS) GO TO 400
SHELL1417 DO 360 N = 1,19,3
SHELL1418 G1 = G(N)
SHELL1419 G2 = G(N+1)
SHELL1420 G3 = G(N+2)
SHELL1421 G(N) = CM1*G1 + CM12*G2 + CM13*G3
SHELL1422 G(N+1) = CM12*G1 + CM22*G2 + CM23*G3
SHELL1423 360 G(N+2) = CM13*G1 + CM23*G2 + CM33*G3
SHELL1424 DO 390 I = 1,J
SHELL1425 X = 0.
SHELL1426 DO 380 N = 1,21
SHELL1427 380 K = X + G(N)*P(N,I)
SHELL1428 X = X*FAC
SHELL1429 ST(I,J) = X
SHELL1430 390 ST(J,I) = X
SHELL1431 400 CONTINUE
SHELL1432 1000 RETURN
SHELL1433 END

```

```

BOUN 1      SUBROUTINE BOUND
BOUN 2      COMMON A(11)
BOUN 3      COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
BOUN 4      COMMON / JUNK / LT,LH,L,SIG(20)
BOUN 5      C
BOUN 6      IF (NPAR(1).EQ.0) GO TO 500
BOUN 7      CALL CLAMP (NPAR(2),A(N1),A(N2),A(N3),A(N4),NUMNP,MBAND)
BOUN 8      RETURN
BOUN 9      C
BOUN 10     500 WRITE (6,2002)
BOUN 11     NUME=NPAR(2)
BOUN 12     DO 800 MM=1,NUME
BOUN 13     CALL STRSC (A(N1),A(N3),NEQ,0)
BOUN 14     WRITE (6,2001)
BOUN 15     DO 800 L=LT,LH
BOUN 16     CALL STRSC (A(N1),A(N3),NEQ,1)
BOUN 17     WRITE (6,3002) MM,L,(SIG(I),I=1,2)
BOUN 18     800 CONTINUE
BOUN 19     RETURN
BOUN 20     C
BOUN 21     2001 FORMAT (/)
BOUN 22     2002 FORMAT (/18HD CONSTRAINT FORCE //
BOUN 23     1 5H NODE NUMBER   LOAD CASE      FORCE      MOMENT//)
BOUN 24     3002 FORMAT (1X,2I10,4X,2E15.5)
BOUN 25     END

```

```

BOUN 26     SUBROUTINE CLAMP (NUMEL,IO,X,Y,Z,NUMNP,MBAND)
BOUN 27     COMMON/EM/LM(24),ND,NS,S(24,24),P(24,4),XM(24),SA(12,24),TT(12,4)
BOUN 28     DIMENSION X(1),Y(1),Z(1),ID(NUMNP,1)
BOUN 29     COMMON / JUNK / R(6),RM(4)
BOUN 30     C
BOUN 31     DO 30 I=1,1058
BOUN 32     30 LM(I)=0
BOUN 33     NS=Z
BOUN 34     ND=6
BOUN 35     READ (5,1005) RM
BOUN 36     C
BOUN 37     NE=0
BOUN 38     WRITE (6,2000) NUMEL
BOUN 39     210 KG=0
BOUN 40     MARK=0
BOUN 41     C
BOUN 42     200 READ (5,1000) NP,NI,NJ,NK,NL,KD,KR,KN,SD,SR,TRACE
BOUN 43     IF (TRACE.EQ.0.) TRACE=1.0E+10
BOUN 44     IF (KG.GT.0) GO TO 550
BOUN 45     C
BOUN 46     KG=KN
BOUN 47     IF(NJ.EQ.0) GO TO 250
BOUN 48     X1=X(NJ)-X(NI)
BOUN 49     Y1=Y(NJ)-Y(NI)
BOUN 50     Z1=Z(NJ)-Z(NI)
BOUN 51     X2=X(NL)-X(NK)
BOUN 52     Y2=Y(NL)-Y(NK)
BOUN 53     Z2=Z(NL)-Z(NK)
BOUN 54     T1=Y1*Z2-Y2*Z1
BOUN 55     T2=Z1*X2-Z2*X1
BOUN 56     T3=X1*Y2-X2*Y1
BOUN 57     GO TO 260
BOUN 58     250 T1=X(NI)-X(NP)
BOUN 59     T2=Y(NI)-Y(NP)
BOUN 60     T3=Z(NI)-Z(NP)
BOUN 61     260 XL=T1*T1+T2*T2+T3*T3
BOUN 62     XL=SQRT(XL)
BOUN 63     T1=T1/XL
BOUN 64     T2=T2/XL
BOUN 65     T3=T3/XL
BOUN 66     C
BOUN 67     IF (KD.EQ.0) GO TO 300
BOUN 68     SA(1,1)=T1*TRACE

```

```

BOUN 69     SA(1,2)=T2*TRACE
BOUN 70     SA(1,3)=T3*TRACE
BOUN 71     S(1,1)=T1*T1*TRACE
BOUN 72     S(1,2)=T1*T2*TRACE
BOUN 73     S(1,3)=T1*T3*TRACE
BOUN 74     S(2,2)=T2*T2*TRACE
BOUN 75     S(2,3)=T2*T3*TRACE
BOUN 76     S(3,3)=T3*T3*TRACE
BOUN 77     PP=TRACE*SD
BOUN 78     R(1)=T1*PP
BOUN 79     R(2)=T2*PP
BOUN 80     R(3)=T3*PP
BOUN 81     GO TO 350
BOUN 82     300 S(1,1)=0.
BOUN 83     S(1,2)=0.
BOUN 84     S(1,3)=0.
BOUN 85     S(2,2)=0.
BOUN 86     S(2,3)=0.
BOUN 87     S(3,3)=0.
BOUN 88     SA(1,1)=0.
BOUN 89     SA(1,2)=0.
BOUN 90     SA(1,3)=0.
BOUN 91     R(1)=0.
BOUN 92     R(2)=0.
BOUN 93     R(3)=0.
BOUN 94     C
BOUN 95     350 IF (KR.EQ.0) GO TO 400
BOUN 96     SA(2,4)=T1*TRACE
BOUN 97     SA(2,5)=T2*TRACE
BOUN 98     SA(2,6)=T3*TRACE
BOUN 99     S(4,4)=T1*T1*TRACE
BOUN 100    S(4,5)=T1*T2*TRACE
BOUN 101    S(4,6)=T1*T3*TRACE
BOUN 102    S(5,5)=T2*T2*TRACE
BOUN 103    S(5,6)=T2*T3*TRACE
BOUN 104    S(6,6)=T3*T3*TRACE
BOUN 105    PP=TRACE*SR
BOUN 106    R(4)=T1*PP
BOUN 107    R(5)=T2*PP
BOUN 108    R(6)=T3*PP
BOUN 109    GO TO 450
BOUN 110    400 S(4,4)=0.
BOUN 111    S(4,5)=0.
BOUN 112    S(4,6)=0.
BOUN 113    S(5,5)=0.
BOUN 114    S(5,6)=0.
BOUN 115    S(6,6)=0.
BOUN 116    SA(2,4)=0.
BOUN 117    SA(2,5)=0.
BOUN 118    SA(2,6)=0.
BOUN 119    R(4)=0.
BOUN 120    R(5)=0.
BOUN 121    R(6)=0.
BOUN 122    450 DO 500 I=2,6
BOUN 123    I1=I-1
BOUN 124    DO 500 J=1,I1
BOUN 125    500 S(I,J)=S(J,I)
BOUN 126    DO 520 I=1,ND
BOUN 127    DO 520 J=1,4
BOUN 128    520 P(I,J)=R(I)*RM(J)
BOUN 129    NN=NP
BOUN 130    NN1=NI
BOUN 131    NNJ=NJ
BOUN 132    NNK=NK
BOUN 133   >NNL=NL
BOUN 134    NND=KD
BOUN 135    NNR=KR
BOUN 136    SSD=SD
BOUN 137    SSR=SR
BOUN 138    TTR=TRACE
BOUN 139    GO TO 560
BOUN 140    550 MARK=1
BOUN 141    555 NN=NN+KG
BOUN 142    NNI=NNI+KG
BOUN 143    560 WRITE (6,2100) NN,NNI,NNJ,NNK,NNI,NKD,NKR,KN,SSD,SSR,TTR

```

```

BOUN 144      NE=NE+1
BOUN 145      C
BOUN 146      DO 600 I=1,ND
BOUN 147      600 LM(I)=10(NN,I)
BOUN 148      C
BOUN 149      NDM=24
BOUN 150      CALL CALBAN (MBAND,NDIF,LM,XM,S,P,ND,NDM)
BOUN 151      WRITE (1) ND,NS,(LM(L),L=1,ND),((SA(L,K),L=1,NS),K=1,ND),
BOUN 152      1 ((TT(L,K),L=1,NS),K=1,4)
BOUN 153      C
BOUN 154      IF ( N .EQ. NUMEL ) GO TO 700
BOUN 155      IF (NN.LT.NP) GO TO 555
BOUN 156      IF (MARK.EQ.1) GO TO 210
BOUN 157      GO TO 200
BOUN 158      700 WRITE (6,2005) RM
BOUN 159      RETURN
BOUN 160      C
BOUN 161      1000 FORMAT (8I5,3F10.0)
BOUN 162      1005 FORMAT (4F10.0)
BOUN 163      2000 FORMAT (25H1 ...BOUNDARY ELEMENTS...//28H ELEMENT TYPE.....*
BOUN 164      . 7//23H NUMBER OF ELEMENTS...= I5 ///
BOUN 165      . 6X 4HNODE 40H ..NODES DEFINING CONSTRAINT DIRECTION.. 5X 5HCODES
BOUN 166      . 5X 7X 5HDISPL 5X 0HROTATION 7X 5HSTIFF/9X 1HN 8X 2HNI 8X 2HNJ
BOUN 167      . 8X 2HNK 8X 2HNL 3X 2HKD 3X 2HKR 3X 2HKN 11X 1HD 11X 1HR 11X 1HS)
BOUN 168      2005 FORMAT (////25H ELEMENT LOAD MULTIPLIERS//
BOUN 169      . 9X 1HA 9X 1HB 9X 1HC 9X 1HD /4F10.4)
BOUN 170      2100 FORMAT (5I10,3I5,1P3E12.2)
BOUN 171      END

```

```

TSHL 1 SUBROUTINE THKSHL
TSHL 2 C
TSHL 3 COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
TSHL 4 COMMON /JUNK/ MM,L,K,NTAG,SIG(200)
TSHL 5 COMMON A(1)
TSHL 6 C
TSHL 7 IF(NPAR(1).EQ.0) GO TO 500
TSHL 8 N6=N5+NPAR(3)
TSHL 9 N7=N6+NPAR(3)
TSHL 10 N8=N7+NPAR(3)
TSHL 11 N9=N8+NPAR(3)
TSHL 12 N10=N9+NPAR(4)
TSHL 13 N11=N10+NPAR(4)
TSHL 14 N12=N11+NPAR(4)
TSHL 15 N13=N12+NPAR(4)
TSHL 16 N14=N13+63*63
TSHL 17 N15=N14+62*63
TSHL 18 IF(N15.GT.MTOT) CALL ERROR(N15-MTOT)
TSHL 19 CALL ELST3D (A(N13),A(N14),NPAR(2),NPAR(3),NPAR(4),A(N1),A(N2),
TSHL 20 1A(N3),A(N4),A(N5),A(N6),A(N7),A(N8),A(N9),A(N10),A(N11),A(N12),
TSHL 21 2NUMNP)
TSHL 22 RETURN
TSHL 23 C
TSHL 24 500 CONTINUE
TSHL 25 CALL ST3D1E (A(N3),A(N1),NEQ,NPAR(2))
TSHL 26 C
TSHL 27 RETURN
TSHL 28 C
TSHL 29 END

```

```

TSHL 30 SUBROUTINE ELST3D (S,STR,N3DEL,NMAT,NLD,IO,X,Y,Z,EE,ENU,RHO,ALPT,
TSHL 31 1KTYPE,PR,YREF,NFACE,NUMNP)
TSHL 32 C
TSHL 33 STIFFNESS SUBROUTINE FOR 48 D.F. ISOPARAMETRIC3D THICK SHELL ELEM.
TSHL 34 C LINEAR ELASTIC ISOTROPIC MATERIAL
TSHL 35 C NINT=NINT*(NINT-1) GAUSSIAN INTEGRATION RULE USED
TSHL 36 C NINT=1,2,3,4
TSHL 37 C
TSHL 38 COMMON /ELPAR/ NPAR(14),NUMN,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,NEQ
TSHL 39 COMMON/EM/LM(48),ND,NDM,RF(63,4),XM(48),SA(21,3),DL(16),TI(48),
TSHL 40 1XLF(4),YLF(4),ZLF(4),TLF(4),PLF(4),NP(16),INP(16),DETT(15)
TSHL 41 COMMON /JUNK/ E1,E2,E3,DET,MLD(4),KLD(4),MULT(4),A(3,3),
TSHL 42 1P(3,21),B(3,3),XX(16,3),Q(19),REFT,INEL,ININT,IMAT,IINC,TTEMP,NEL,
TSHL 43 2ML,NINT,MAT,INC,TAG,TEMP,SKIP,I,J,K,L,FAC,CC1,CC2,CC3,CC4,G,DEN,
TSHL 44 3FACT,GT,GG,C1,C2,C3,C,K1,K2
TSHL 45 COMMON/GASS/XK(4,4),WGT(4,4),IPERM(3)
TSHL 46 C
TSHL 47 DIMENSION S(63,63),STR(42,63),KTYPE(1),PR(1),YREF(1),NFACE(1),
TSHL 48 1X(1),Y(1),Z(1),TD(NUMNP,1),EE(1),ENU(1),RHO(1),ALPT(1)
TSHL 49 DIMENSION STPTS(7,3)
TSHL 50 C
TSHL 51 DATA IPERM / 2,3,1 /
TSHL 52 DATA XK / 0., 0., 0., 0., 0., 0.,
TSHL 53 1 -.5773502691896, .5773502691896, 0., 0., 0.,
TSHL 54 2 -.7745966692415, .000000000000, .7745966692415, 0.,
TSHL 55 3 -.8611363115941, -.3399810435849, .3399810435849, .8611363115941/
TSHL 56 DATA WGT / 2.000, 0., 0., 0., 0.,
TSHL 57 1 1.000000000000,1.000000000000, 0., 0.,
TSHL 58 2 .5555555555556, .8888888888889, .5555555555556, 0.,
TSHL 59 3 .3478548451375, .6521451548625, .6521451548625, .3478548451375/
TSHL 60 DATA STPTS / 0., 1., -1., 0., 0., 0., 0.,
TSHL 61 1 0., 0., 0., 1., -1., 0., 0.,
TSHL 62 2 0., 0., 0., 0., 0., 1., -1. /
TSHL 63 C
TSHL 64 DO 5 I=1,350
TSHL 65 5 LM(I)=0
TSHL 66 C
TSHL 67 MATERIAL PROPERTIES
TSHL 68 C

```

```

TSHL 69 WRITE (6,1300)
TSHL 70 DO 10 I=1,NMAT
TSHL 71 READ (5,1001) N,EE(N),ENU(N),RHO(N),ALPT(N)
TSHL 72 10 WRITE (6,2001) N,EE(N),ENU(N),RHO(N),ALPT(N)
TSHL 73 C
TSHL 74 ELEMENT DISTRIBUTED LOAD CARDS
TSHL 75 C
TSHL 76 IF(NLD) 23,23,15
TSHL 77 15 WRITE (6,1302)
TSHL 78 DO 16 I=1,NLD
TSHL 79 READ (5,1002) N,KTYPE(N),PR(N),YREF(N),NFACE(N)
TSHL 80 16 WRITE (6,2002) N,KTYPE(N),PR(N),YREF(N),NFACE(N)
TSHL 81 C
TSHL 82 23 READ (5,1003) REFT,GRAV,PLF,TLF,XLF,YLF,ZLF
TSHL 83 WRITE (6,2003) REFT,PLF,TLF,XLF,YLF,ZLF
TSHL 84 GRAV=1.0
TSHL 85 WRITE (6,1301)
TSHL 86 NEL=0
TSHL 87 30 READ (5,1000) INEL,ININT,IMAT,IINC,MLD,TTEMP,INP
TSHL 88 DO 39 I=1,4
TSHL 89 39 MULT(I)=1
TSHL 90 IF(IINC.EQ.0) IINC=1
TSHL 91 IF(IMAT.EQ.0) IMAT=1
TSHL 92 40 NEL=NEL+1
TSHL 93 ML=INEL-NEL
TSHL 94 IF(ML) 50,55,60
TSHL 95 50 WRITE (6,4003) INEL
TSHL 96 STOP
TSHL 97 55 DO 56 I=1,16
TSHL 98 56 NP(I)=INP(I)
TSHL 99 NINT=ININT
TSHL 100 MAT=IMAT
TSHL 101 INC=IINC
TSHL 102 TAG=1HI
TSHL 103 TEMP=TTEMP
TSHL 104 SKIP=999.
TSHL 105 IF(NINT) 33,33,57
TSHL 106 33 NINT=IABS(NINT)
TSHL 107 SKIP=1.
TSHL 108 IF(NINT.EQ.0) SKIP=0.
TSHL 109 57 CONTINUE
TSHL 110 DO 59 I=1,4
TSHL 111 KLD(I)=IABS(MLD(I))
TSHL 112 IF(MLD(I)) 58,58,59
TSHL 113 58 MULT(I)=0
TSHL 114 59 CONTINUE
TSHL 115 GO TO 62
TSHL 116 C
TSHL 117 60 DO 61 I=1,16
TSHL 118 61 NP(I)=NP(I)+INC
TSHL 119 TAG=1H
TSHL 120 DO 64 I=1,4
TSHL 121 64 KLD(I)=KLD(I)*MULT(I)
TSHL 122 62 WRITE (6,2000) NEL,NP,NINT,MAT,TAG,KLD,TEMP
TSHL 123 C
TSHL 124 C
TSHL 125 DO 80 I = 1,16
TSHL 126 K=NP(I)
TSHL 127 XX(I,1)=X(K)
TSHL 128 XX(I,2)=Y(K)
TSHL 129 80 XX(I,3)=Z(K)
TSHL 130 K=MAT
TSHL 131 CC4=ALPT(K)*(TEMP-REFT)
TSHL 132 FAC = EE(K)/{(1.-2.*ENU(K))*{1.+ENU(K)}}
TSHL 133 FACT=FAC*ALPT(K)*(TEMP-REFT)*{1.+ENU(K)}
TSHL 134 IF (SKIP) 70,70,63
TSHL 135 63 SKIP=SKIP-1.
TSHL 136 CC1=1.-ENU(K)
TSHL 137 CC2=ENU(K)
TSHL 138 CC3=.5-ENU(K)
TSHL 139 DEN = RHO(K)
TSHL 140 C
TSHL 141 L3=63*63
TSHL 142 DO 100 I=1,L3
TSHL 143 100 S(I)=0.

```

```

TSHL 144      DO 110 I=1,48
TSHL 145      110 TT(I)=0.
TSHL 146      DO 120 I=1,16
TSHL 147      120 DL(I)=0.
TSHL 148      C
TSHL 149      C      LOOP OVER NINT**3 INTEGRATION POINTS
TSHL 150      C
TSHL 151      MINT=NINT-1
TSHL 152      DO 300 LX = 1,NINT
TSHL 153      E1=XK(LX,NINT)
TSHL 154      DO 300 LY = 1,NINT
TSHL 155      E2=XK(LY,NINT)
TSHL 156      DO 300 LZ = 1,MINT
TSHL 157      E3=XK(LZ,MINT)
TSHL 158      C
TSHL 159      CALL FUNCT(1,SA)
TSHL 160      C
TSHL 161      G = WGT(LX,NINT)*WGT(LY,NINT)*WGT(LZ,MINT)
TSHL 162      GT=G
TSHL 163      GG=G*DET*DEN
TSHL 164      G=G*FAC/DET
TSHL 165      C1=G*CC1
TSHL 166      C2=G*CC2
TSHL 167      C3=G*CC3
TSHL 168      L=0
TSHL 169      DO 310 I=1,16
TSHL 170      DL(I)=DL(I) + GG*Q(I)
TSHL 171      DO 310 K=1,3
TSHL 172      L=L+1
TSHL 173      310 TT(L)=TT(L) + GT*SA(I,K)
TSHL 174      C
TSHL 175      C      ADD CONTRIBUTION TO STIFFNESS MATRIX
TSHL 176      C
TSHL 177      DO 300 I=1,21
TSHL 178      K3 = 3*I
TSHL 179      K2 = K3 - 1
TSHL 180      K1 = K2 - 1
TSHL 181      UI=SA(I,1)
TSHL 182      VI=SA(I,2)
TSHL 183      WI=SA(I,3)
TSHL 184      DO 300 J=1,21
TSHL 185      L3 = 3*J
TSHL 186      L2 = L3 - 1
TSHL 187      L1 = L2 - 1
TSHL 188      UJ=SA(J,1)
TSHL 189      VJ=SA(J,2)
TSHL 190      WJ=SA(J,3)
TSHL 191      UU=UI*UJ
TSHL 192      VV=VI*VJ
TSHL 193      WW=WI*WJ
TSHL 194      UV=UI*VJ
TSHL 195      VU=VI*UJ
TSHL 196      UW=UI*WJ
TSHL 197      WU=WI*UJ
TSHL 198      VW=VI*WJ
TSHL 199      WV=WI*VJ
TSHL 200      S(K1,L1) = S(K1,L1) + C1*UU + C3*(VV+WW)
TSHL 201      S(K2,L2) = S(K2,L2) + C1*VV + C3*(UU+UU)
TSHL 202      S(K3,L3) = S(K3,L3) + C1*WW + C3*(UU+VV)
TSHL 203      S(K1,L2) = S(K1,L2) + C2*UV + C3*VU
TSHL 204      S(K1,L3) = S(K1,L3) + C2*UW + C3*WU
TSHL 205      S(K2,L3) = S(K2,L3) + C2*VW + C3*WV
TSHL 206      IF (I.EQ.J) GO TO 300
TSHL 207      S(K2,L1) = S(K2,L1) + C2*VU + C3*UV
TSHL 208      S(K3,L1) = S(K3,L1) + C2*WU + C3*UW
TSHL 209      S(K3,L2) = S(K3,L2) + C2*WV + C3*VW
TSHL 210      300 CONTINUE
TSHL 211      C
TSHL 212      C      FORM STRAIN MATRIX
TSHL 213      C
TSHL 214      DO 305 I=1,2646
TSHL 215      305 STR(I)=0.
TSHL 216      L2=1
TSHL 217      DO 405 L1=1,7
TSHL 218      E1=STPTS(L1,1)

```

```

TSHL 219      E2=STPTS(L1,2)
TSHL 220      E3=STPTS(L1,3)
TSHL 221      CALL FUNCT(2,SA)
TSHL 222      DETT(L1)=DET
TSHL 223      C
TSHL 224      L3=6*L1-6
TSHL 225      DO 402 K=1,21
TSHL 226      K3=3*K
TSHL 227      K2=K3-1
TSHL 228      K1=K2-1
TSHL 229      STR(L3+1,K1) = SA(K,1)
TSHL 230      STR(L3+2,K2) = SA(K,2)
TSHL 231      STR(L3+3,K3) = SA(K,3)
TSHL 232      STR(L3+4,K1) = SA(K,2)
TSHL 233      STR(L3+4,K2) = SA(K,1)
TSHL 234      STR(L3+5,K2) = SA(K,3)
TSHL 235      STR(L3+5,K3) = SA(K,2)
TSHL 236      STR(L3+6,K1) = SA(K,3)
TSHL 237      402 STR(L3+6,K3) = SA(K,1)
TSHL 238      C
TSHL 239      405 L2=L2+3
TSHL 240      C
TSHL 241      C      STATIC CONDENSATION
TSHL 242      C
TSHL 243      DO 710 M=1,15
TSHL 244      MN=64-M
TSHL 245      MO=MN-1
TSHL 246      C
TSHL 247      C      STIFFNESS MATRIX - S
TSHL 248      C
TSHL 249      SP=S(MN,MN)
TSHL 250      DO 650 I=1,MO
TSHL 251      650 S(MN,I)=S(I,MN)/SP
TSHL 252      DO 700 K=1,MO
TSHL 253      SP=S(MN,K)
TSHL 254      DO 700 J=1,K
TSHL 255      700 S(J,K)=S(J,K) - SP*S(J,MN)
TSHL 256      C
TSHL 257      C      DERIVATIVE MATRIX - STR
TSHL 258      C
TSHL 259      DO 710 J=1,42
TSHL 260      SP=STR(J,MN)
TSHL 261      IF(SP.EQ.0.) GO TO 710
TSHL 262      DO 705 K=1,MO
TSHL 263      705 STR(J,K)=STR(J,K) - SP*S(MN,K)
TSHL 264      710 CONTINUE
TSHL 265      DO 760 I=1,48
TSHL 266      DO 760 J=1,48
TSHL 267      760 S(I,J)=S(I,J)
TSHL 268      C
TSHL 269      C      SAVE ELASTIC PROPERTIES
TSHL 270      C
TSHL 271      DETT(8)=CC1
TSHL 272      DETT(9)=CC2
TSHL 273      DETT(10)=CC3
TSHL 274      DETT(11)=FAC
TSHL 275      DO 510 I=1,4
TSHL 276      510 DETT(11+I)=CC4*TLF(I)
TSHL 277      70 CONTINUE
TSHL 278      C
TSHL 279      C      DISTRIBUTED LOAD
TSHL 280      C
TSHL 281      DO 410 J=1,63
TSHL 282      DO 410 I=1,4
TSHL 283      410 RF(J,I)=0.
TSHL 284      C
TSHL 285      440 CALL LDCAL(KTYPE,PR,YREF,NFACE)
TSHL 286      C
TSHL 287      C      SELF WGT.
TSHL 288      C
TSHL 289      450 DO 460 II=1,16
TSHL 290      K=3*II
TSHL 291      J=K-1
TSHL 292      I=J-1
TSHL 293      DO 460 L=1,4

```

```

TSHL 294      RF(I,L) = RF(I,L)*PLF(L) + DL(II)*XLFL(L)
TSHL 295      RF(J,L) = RF(J,L)*PLF(L) + DL(II)*YLFL(L)
TSHL 296      460 RF(K,L) = RF(K,L)*PLF(L) + DL(II)*ZLFL(L)
TSHL 297      C
TSHL 298      C      THERMAL LOADS
TSHL 299      C
TSHL 300      DO 470 I=1,48
TSHL 301      GT=TT(I)*FACT
TSHL 302      DO 470 J=1,4
TSHL 303      470 RF(I,J)=RF(I,J) + GT*TLF(J)
TSHL 304      C
TSHL 305      C      MASS ARRAY
TSHL 306      C
TSHL 307      L=0
TSHL 308      DO 465 I=1,16
TSHL 309      DO 465 J=1,3
TSHL 310      L=L+1
TSHL 311      465 XM(L)=DL(I)/GRAV
TSHL 312      C
TSHL 313      IJ=0
TSHL 314      DO 550 I=1,16
TSHL 315      II=NP(I)
TSHL 316      DO 550 J=1,3
TSHL 317      IJ=IJ+1
TSHL 318      550 LM(IJ)=ID(II,J)
TSHL 319      NS=42
TSHL 320      ND=48
TSHL 321      NDM=63
TSHL 322      C
TSHL 323      CALL CALBAN (MBAND,NCIF,LM,XM,S,RF,ND,NDM)
TSHL 324      WRITE (1) ND,NS,(DETT(I),I=1,15),(LM(I),I=1,ND),((STR(I,J),I=1,NS)
TSHL 325      1,J=1,ND)
TSHL 326      C
TSHL 327      C      CHECK IF LAST ELEMENT
TSHL 328      C
TSHL 329      IF(N3DEL-NEL) 50,600,590
TSHL 330      590 IF(ML) 30,30,40
TSHL 331      C
TSHL 332      C
TSHL 333      600 RETURN
TSHL 334      C
TSHL 335      C
TSHL 336      1000 FORMAT (4I5,4I2,2X,F10.2/1615)
TSHL 337      1001 FORMAT (I5,4F10.0)
TSHL 338      1002 FORMAT (2I5,2F10.2,I5)
TSHL 339      1003 FORMAT (2F10.2/(4F10.2))
TSHL 340      2000 FORMAT (I6,X,8I5/7X,8I5,I9,I12,8X,A1,3X,4I5,F8.1/)
TSHL 341      2001 FORMAT (X,I5,4E15.4)
TSHL 342      2002 FORMAT (I5,I9,2F13.3,I12)
TSHL 343      2003 FORMAT (3I11,...,STRESS FREE TEMPERATURE = F10.3////
TSHL 344      . 38H LOAD FACTORS FOR 4 ELEMENT LOAD CASES //
TSHL 345      . 46X 17ELEMENT LOAD CASE /
TSHL 346      . 36X 1HA 9X 1HB 9X 1HC 9X 1HD /
TSHL 347      . 30H PRESSURE LOAD FACTORS . . . 4F10.3/
TSHL 348      . 30H THERMAL LOAD FACTORS . . . 4F10.3//
TSHL 349      . 30H PERCENT GRAVITY IN +X DIRN. 4F10.3/
TSHL 350      . 30H PERCENT GRAVITY IN +Y DIRN. 4F10.3/
TSHL 351      . 30H PERCENT GRAVITY IN +Z DIRN. 4F10.3/ )
TSHL 352      1300 FORMAT (9HIMATERIAL 10X 1HE 12X 2HNU 10X 3HRHO 11X 7HALPHA-T /
TSHL 353      . 8H NUMBER //)
TSHL 354      1301 FORMAT (30H1...16 NODE SOLID ELEMENT DATA ///
TSHL 355      . 8H ELEMENT 10X 15HCONNECTED NODES 17X 28HINTEGRATION MATERIAL I
TSHL 356      .NPUT 7X 13HELEMENT LOADS 5X 7HELEMENT /
TSHL 357      . 8H NUMBER 3X,36H1 2 3 4 5 6 7 8 6X,5HORDER
TSHL 358      . 7X,3HNC. 6X 3HTAG 7X 16H1 2 3 4 4X 5HTEMP. /
TSHL 359      . 11X,36H9 10 11 12 13 14 15 16 // )
TSHL 360      1302 FORMAT (//////26H ELEMENT DISTRIBUTED LOADS //
TSHL 361      . 52H NUMBER KTYPE PR YREF FACE )
TSHL 362      4003 FORMAT (36HELEMENT CARD ERROR, ELEMENT NUMBER= 16)
TSHL 363      C
TSHL 364      END

```

```

TSHL 365      SUBROUTINE FUNCT (KK,D)
TSHL 366      C
TSHL 367      COMMON /JUNK/ R ,S ,T ,DET,MLD(4),KLD(4),MULT(4),A(3,3),
TSHL 368      IP(3,21),B(3,3),XZ(16,3),Q(19)
TSHL 369      COMMON/GASS/XK(4,4),WGT(4,4),IPERM(3)
TSHL 370      DIMENSION C(21,3),BB(3)
TSHL 371      C
TSHL 372      R2=2.*(1.-(R**2))
TSHL 373      S2=2.*(1.-(S**2))
TSHL 374      PP=-.125*(1.+R)
TSHL 375      RN=-.125*(1.-R)
TSHL 376      SP=1.+S
TSHL 377      SN=1.-S
TSHL 378      TP=1.+T
TSHL 379      TN=1.-T
TSHL 380      RPSP= R+S-1.
TSHL 381      RPSN= R-S-1.
TSHL 382      RNSP=-R+S-1.
TSHL 383      RNSN=-R-S-1.
TSHL 384      XPP= 2.*R+S
TSHL 385      XPN= 2.*R-S
TSHL 386      XNP=-2.*R+S
TSHL 387      XNN=-2.*R-S
TSHL 388      YPP= R+(2.*S)
TSHL 389      YPN= R-(2.*S)
TSHL 390      YNP=-R+(2.*S)
TSHL 391      YNN=-R-(2.*S)
TSHL 392      XX=-.125
TSHL 393      IF (KK.EQ.2) GO TO 100
TSHL 394      C
TSHL 395      C      SHAPE FUNCTIONS
TSHL 396      C
TSHL 397      Q( 1)=RP*SP*TP*RPSP
TSHL 398      Q( 2)=RN*SN*TP*RNPS
TSHL 399      Q( 3)=RN*SN*TP*RNNS
TSHL 400      Q( 4)=RP*SN*TP*RPNS
TSHL 401      Q( 5)=RP*SP*TN*RPSP
TSHL 402      Q( 6)=RN*SN*TN*RNPS
TSHL 403      Q( 7)=RN*SN*TN*RNNS
TSHL 404      Q( 8)=RP*SN*TN*RPNS
TSHL 405      Q( 9)=R2*SP*TP*XX
TSHL 406      Q(10)=RN*S2*TP
TSHL 407      Q(11)=R2*SN*TP*XX
TSHL 408      Q(12)=RP*S2*TP
TSHL 409      Q(13)=R2*SP*TN*XX
TSHL 410      Q(14)=RN*S2*TN
TSHL 411      Q(15)=R2*SN*TN*XX
TSHL 412      Q(16)=RP*S2*TN
TSHL 413      C
TSHL 414      C      DERIVATIVES OF SHAPE FUNCTIONS
TSHL 415      C
TSHL 416      100 XR=-.5*R
TSHL 417      XS=-.4*S
TSHL 418      P(1,1)= XX*SP*TP*XPP
TSHL 419      P(1,2)=-XX*SP*TP*XNP
TSHL 420      P(1,3)=-XX*SN*TP*XNN
TSHL 421      P(1,4)= XX*SN*TP*XPN
TSHL 422      P(1,5)= XX*SP*TN*XPP
TSHL 423      P(1,6)=-XX*SP*TN*XNP
TSHL 424      P(1,7)=-XX*SN*TN*XNN
TSHL 425      P(1,8)= XX*SN*TN*XPN
TSHL 426      P(1,9)= XR*SP*TP
TSHL 427      P(1,10)=-XX*S2*TP
TSHL 428      P(1,11)= XR*SN*TP
TSHL 429      P(1,12)= XX*S2*TP
TSHL 430      P(1,13)= XR*SP*TN
TSHL 431      P(1,14)=-XX*S2*TN
TSHL 432      P(1,15)= XR*SN*TN
TSHL 433      P(1,16)= XX*S2*TN
TSHL 434      P(1,17)=1.-(3.*R*R)
TSHL 435      P(1,18)=0.
TSHL 436      P(1,19)=0.
TSHL 437      P(1,20)=S*(1.0-(3.0*R*R))
TSHL 438      P(1,21)=S*(1.0-(S*S))
TSHL 439      C

```

```

TSHL 440 P(2,1)= RP*TP*YPP
TSHL 441 P(2,2)= RN*TP*YNP
TSHL 442 P(2,3)=-RN*TP*YNN
TSHL 443 P(2,4)=-RP*TP*YPN
TSHL 444 P(2,5)= RP*TN*YPP
TSHL 445 P(2,6)= RN*TN*YNP
TSHL 446 P(2,7)=-RN*TN*YNN
TSHL 447 P(2,8)=-RP*TN*YPN
TSHL 448 P(2,9)= R2*TP*XX
TSHL 449 P(2,10)= XS*TP*RN
TSHL 450 P(2,11)=-R2*TP*XX
TSHL 451 P(2,12)= XS*TP*RP
TSHL 452 P(2,13)= R2*TN*XX
TSHL 453 P(2,14)= XS*TN*RN
TSHL 454 P(2,15)=-R2*TN*XX
TSHL 455 P(2,16)= XS*TN*RP
TSHL 456 P(2,17)=0.
TSHL 457 P(2,18)=1.-{3.*S*S)
TSHL 458 P(2,19)=0.
TSHL 459 P(2,20)=R*(1.0-(R*R))
TSHL 460 P(2,21)=R*(1.0-(3.0*S*S))
TSHL 461 C
TSHL 462 P(3,1)= RP*SP*RPSP
TSHL 463 P(3,2)= RN*SP*RNNSP
TSHL 464 P(3,3)= RN*SN*RNNSN
TSHL 465 P(3,4)= RP*SN*RPSPN
TSHL 466 P(3,5)=-RP*SP*RPSP
TSHL 467 P(3,6)=-RN*SP*RNNSP
TSHL 468 P(3,7)=-RN*SN*RNNSN
TSHL 469 P(3,8)=-RP*SN*RPSPN
TSHL 470 P(3,9)= R2*SP*XX
TSHL 471 P(3,10)= RN*S2
TSHL 472 P(3,11)= R2*SN*XX
TSHL 473 P(3,12)= RP*S2
TSHL 474 P(3,13)=-R2*SP*XX
TSHL 475 P(3,14)=-RN*S2
TSHL 476 P(3,15)=-R2*SN*XX
TSHL 477 P(3,16)=-RP*S2
TSHL 478 P(3,17)=0.
TSHL 479 P(3,18)=0.
TSHL 480 P(3,19)=-2.*T
TSHL 481 P(3,20)=0.0
TSHL 482 P(3,21)=0.0
TSHL 483 C
TSHL 484 C JACOBIAN MATRIX A
TSHL 485 C
TSHL 486 DD 200 I=1,3
TSHL 487 DD 200 J=1,3
TSHL 488 C=0.
TSHL 489 DD 150 K=1,16
TSHL 490 C=C*P(I,K)*XZ(K,J)
TSHL 491 200 A(I,J)=C
TSHL 492 IF (KK.EQ.3) GO TO 600
TSHL 493 C
TSHL 494 C INVERT JACOBIAN
TSHL 495 C
TSHL 496 DD 300 I=1,3
TSHL 497 J=IPERM(I)
TSHL 498 K=IPERM(J)
TSHL 499 B(I,1)=A(J,J)*A(K,K)-A(K,J)*A(J,K)
TSHL 500 B(I,J)=A(K,J)*A(I,K)-A(I,J)*A(K,K)
TSHL 501 300 B(J,1)=A(J,K)*A(K,I)-A(J,I)*A(K,K)
TSHL 502 DET=A(I,1)*B(1,1)+A(1,2)*B(2,1)+A(1,3)*B(3,1)
TSHL 503 C
TSHL 504 C MATRIX OF X-Y-Z DERIVATIVES
TSHL 505 C
TSHL 506 DD 400 I=1,3
TSHL 507 DD 400 J=1,22
TSHL 508 C=D.
TSHL 509 DD 350 K=1,3
TSHL 510 350 C=C+B(I,K)*P(K,J)
TSHL 511 400 D(J,I)=C
TSHL 512 C
TSHL 513 600 RETURN
TSHL 514 C

```

```

TSHL 515 END

TSHL 516 SUBROUTINE LOCAL(KTYPEE,PRR,YREF,NFACE)
TSHL 517 C
TSHL 518 COMMON/EM/LM(48),ND,NDM,RF(63,4),XM(48),SA(21,3),DL(16)
TSHL 519 COMMON/JUNK/ETA(3),DET,MLD(4),KLD(4),MULT(4),A(3,3),
TSHL 520 IP(3,21),B(3,3),XX(16,3),Q(19)
TSHL 521 COMMON/GASS/DUM(12),XX(4),DDUM(12),WGT(4),IPERM(3)
TSHL 522 C
TSHL 523 DIMENSION KTYPEE(1),PRR(1),YREF(1),NFACE(1),KCRD(6),FVAL(6),
TSHL 524 LKFACE(6,8)
TSHL 525 C
TSHL 526 DATA KFACE / 1, 2, 2, 3, 2, 6,
TSHL 527 1 5, 6, 6, 7,10,14,
TSHL 528 2 16,14,13,15, 3, 7,
TSHL 529 3 8, 7, 5, 8,11,15,
TSHL 530 4 4, 3, 1, 4, 4, 8,
TSHL 531 5 12,10, 9,11,12,16,
TSHL 532 6 0, 0, 0, 0, 1, 5,
TSHL 533 7 0, 0, 0, 0, 9,13/
TSHL 534 DATA KCRD / 1,1,2,2,3,3/
TSHL 535 DATA FVAL / 1.,-1.,1.,-1.,1.,-1./
TSHL 536 C
TSHL 537 C
TSHL 538 DO 700 KK=1,4
TSHL 539 NNN=KLD(KK)
TSHL 540 IF(NNN) 700,700,10
TSHL 541 10 KTYPE=KTYPEE(NNN)
TSHL 542 PR=PRR(NNN)
TSHL 543 YREF=YREF(NNN)
TSHL 544 KF=NFACE(NNN)
TSHL 545 C
TSHL 546 C INTEGRATE OVER THE SURFACE
TSHL 547 C
TSHL 548 ML = KCRD(KF)
TSHL 549 MM = IPERM(ML)
TSHL 550 MN = IPERM(MM)
TSHL 551 ETA(ML) = FVAL(KF)
TSHL 552 DO 300 LX = 1,4
TSHL 553 ETA(MM) = XK(LX)
TSHL 554 DO 300 LY = 1,4
TSHL 555 ETA(MN) = XK(LY)
TSHL 556 CALL FUNCT(3,SA)
TSHL 557 C
TSHL 558 C COMPUTE DIRECTION COSINES OF NORMAL TO SURFACE
TSHL 559 C
TSHL 560 A1 = (A(MN,2)*A(MN,3)-A(MN,3)*A(MN,21))
TSHL 561 A2 = (A(MN,3)*A(MN,1)-A(MN,1)*A(MN,31))
TSHL 562 A3 = (A(MN,1)*A(MN,2)-A(MN,2)*A(MN,11))
TSHL 563 AA=SQRT(A1**2+A2**2+A3**2)
TSHL 564 A1 = A1/AA
TSHL 565 A2 = A2/AA
TSHL 566 A3 = A3/AA
TSHL 567 C
TSHL 568 C COMPUTE FIRST FUND. FORM (SIN / )
TSHL 569 C
TSHL 570 AA = 0.
TSHL 571 BB = 0.
TSHL 572 CC = 0.
TSHL 573 DD 200 I = 1,3
TSHL 574 AA=AA+A(MN,I)**2
TSHL 575 CC=CC+A(MN,I)**2
TSHL 576 200 BB = BB + A(MN,I)*A(MN,1)
TSHL 577 C=SQRT (AA*CC - BB*BB)
TSHL 578 C
TSHL 579 C COMPUTE PRESSURE,LOAD COMPONENTS, STORE IN R
TSHL 580 C
TSHL 581 IF (KTYPE.EQ.2) GO TO 170
TSHL 582 FORCE = PR

```

D-38

```

TSHL 583      GO TO 185
TSHL 584      170 YY = 0.
TSHL 585      DO 180 I = 1,16
TSHL 586      180 YY = YY + Q(I)*XX(I,2)
TSHL 587      YY = YY - YREF
TSHL 588      FORCE = -PR*YY
TSHL 589      IF (YY.GT.0.) FORCE = 0.
TSHL 590      185 CONTINUE
TSHL 591      TS=FORCE*WGT(LX)*WGT(LY)*C
TSHL 592      C
TSHL 593      DO 190 I = 1,8
TSHL 594      N = KFACE(KF,I)
TSHL 595      IF (N.EQ.0) GO TO 190
TSHL 596      QQ=TS*Q(N)
TSHL 597      K=3*N
TSHL 598      RF(K-2,KK) = RF(K-2,KK) + QQ*A1
TSHL 599      RF(K-1,KK) = RF(K-1,KK) + QQ*A2
TSHL 600      RF(K ,KK) = RF(K ,KK) + QQ*A3
TSHL 601      190 CONTINUE
TSHL 602      C
TSHL 603      300 CONTINUE
TSHL 604      C
TSHL 605      700 CONTINUE
TSHL 606      C
TSHL 607      RETURN
TSHL 608      END

```

```

TSHL 609      SUBROUTINE ST3D16(D,STR,NEQ,NUME)
TSHL 610      C
TSHL 611      COMMON /ELPAR/ NPAR(14),NUMNP,MBAND,NELTYP,N1,N2,N3,N4,N5,MTOT,MEQ
TSHL 612      COMMON/EM/ND,NS,LM(48),ST(42,48),DETT(15)
TSHL 613      COMMON/JUNK/LT,LH,MM,L,K,SIG(6),STN(6),E(3,3),C1,C2,C3,I,J,IL,JL,
TSHL 614      LJ2,J3,I1,I2,I3,SS,DET,KX,KY,KZ,DO,SX,FAC
TSHL 615      C
TSHL 616      DIMENSION D(NEQ,1),STR(4,1)
TSHL 617      C
TSHL 618      WRITE (6,2005)
TSHL 619      WRITE (6,3025)
TSHL 620      DO 800 MM=1,NUME
TSHL 621      READ (1) NO,NS,(DETT(I),I=1,15),(LM(I),I=1,ND),((ST(I,J),I=1,NS),
TSHL 622      LJ=1,ND)
TSHL 623      C1=DETT(8)
TSHL 624      C2=DETT(9)
TSHL 625      C3=DETT(10)
TSHL 626      FAC=DETT(11)
TSHL 627      DO 501 I=1,3
TSHL 628      501 E(I,1)=C1
TSHL 629      E(I,2) = C2
TSHL 630      E(I,3) = C2
TSHL 631      E(2,3) = C2
TSHL 632      E(2,1) = C2
TSHL 633      E(3,1) = C2
TSHL 634      E(3,2) = C2
TSHL 635      LL=0
TSHL 636      DO 700 L=LT,LH
TSHL 637      WRITE (6,2000)
TSHL 638      LL=LL+1
TSHL 639      C
TSHL 640      DO 540 IL=1,7
TSHL 641      DO 503 I=1,3
TSHL 642      J=I+3
TSHL 643      SS=0.
TSHL 644      DO 502 KK=1,4
TSHL 645      502 SS=SS-DETT(11+KK)*STR(KK,L)
TSHL 646      STN(I)=SS
TSHL 647      503 STN(J)=0.
TSHL 648      DET=DETT(1L)
TSHL 649      C
TSHL 650      JL=6*(IL-1)

```

```

TSHL 651      DO 510 I=1,48
TSHL 652      I1=LM(I)
TSHL 653      IF(I1) 510,510,505
TSHL 654      505 DD=D(I1,LL)/DET
TSHL 655      DO 508 J=1,6
TSHL 656      508 STN(J)=STN(J) + ST(J1+J,I)*DD
TSHL 657      510 CONTINUE
TSHL 658      C
TSHL 659      SX=C3*FAC
TSHL 660      DO 515 I=1,3
TSHL 661      SS=0.
TSHL 662      DO 514 J=1,3
TSHL 663      514 SS = SS + E(I,J)*STN(J)*FAC
TSHL 664      SIG(I)=SS
TSHL 665      J=I+3
TSHL 666      515 SIG(J)=STN(J)*SX
TSHL 667      C
TSHL 668      I=IL-1
TSHL 669      WRITE (6,3005) MM,L,I,SIG
TSHL 670      C
TSHL 671      540 CONTINUE
TSHL 672      590 CONTINUE
TSHL 673      700 CONTINUE
TSHL 674      800 CONTINUE
TSHL 675      C
TSHL 676      RETURN
TSHL 677      C
TSHL 678      2000 FORMAT (//)
TSHL 679      2005 FORMAT (36H1....16-NODE SOLID ELEMENT STRESSES //
TSHL 680      . 31X,67H SIG-XX      SIG-YY      SIG-ZZ      SIG-XY      SIG-YZ
TSHL 681      .  SIG-ZX )
TSHL 682      3005 FORMAT (/I6,I9,I8,5X,1P6E12.3)
TSHL 683      3025 FORMAT (/724H ELEMENT LOAD NO. FACE )
TSHL 684      C
TSHL 685      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 commercial 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.

- E2 -

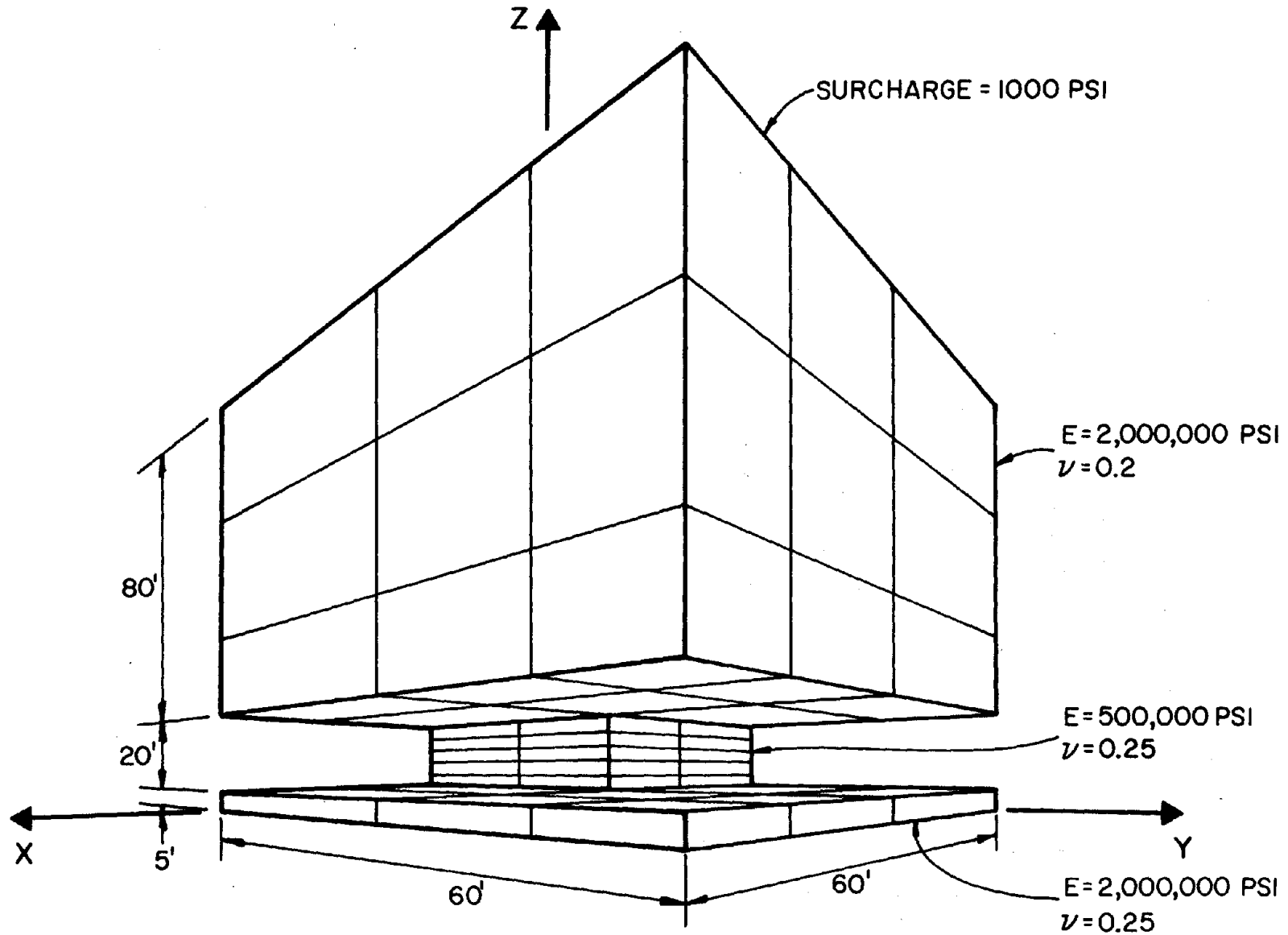


FIGURE E.1 TYPICAL MINE STRUCTURE

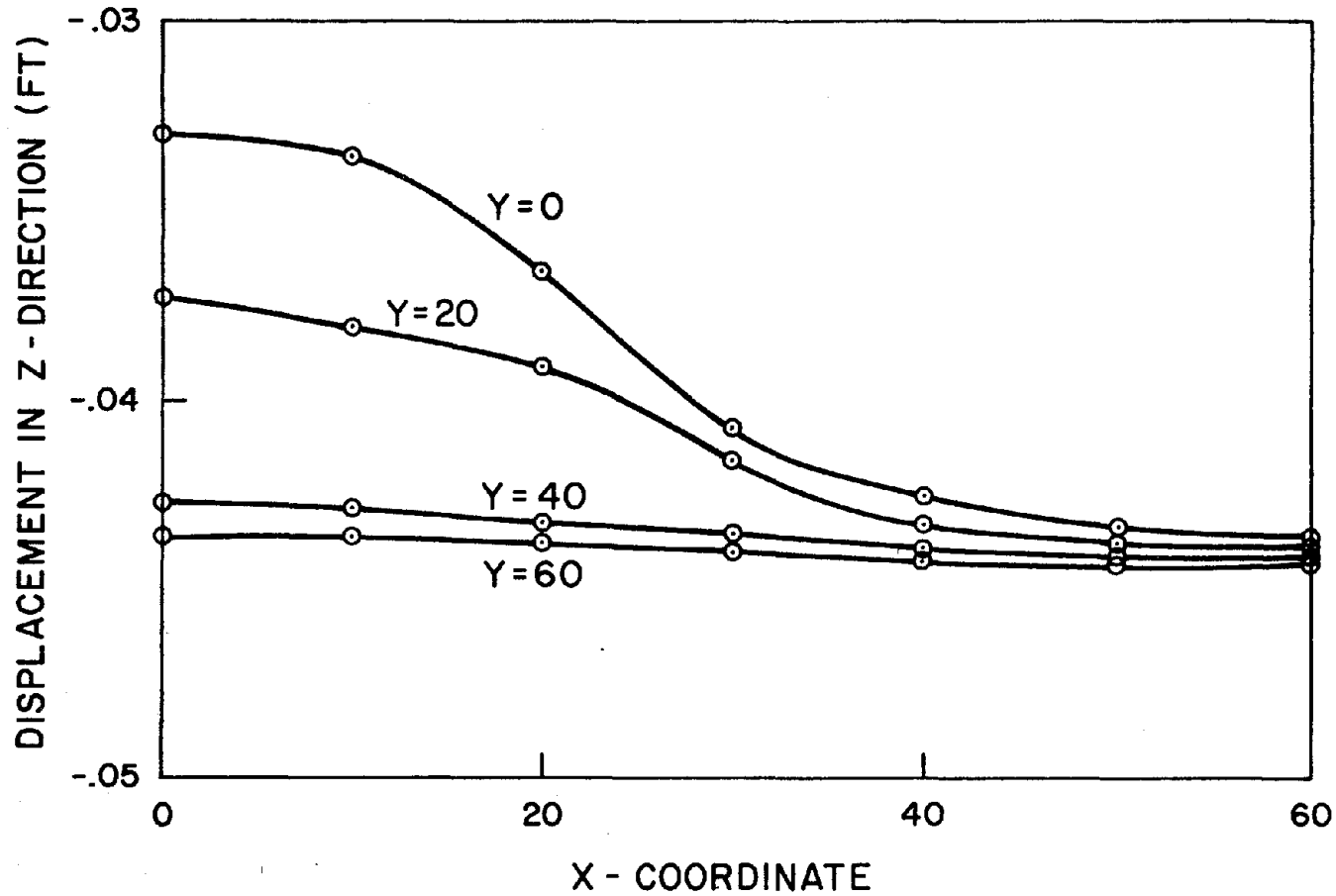


FIGURE E.2 VERTICAL DISPLACEMENTS OF THE CEILING OF THE MINE ROOM (Z=25 FT)

- E4 -

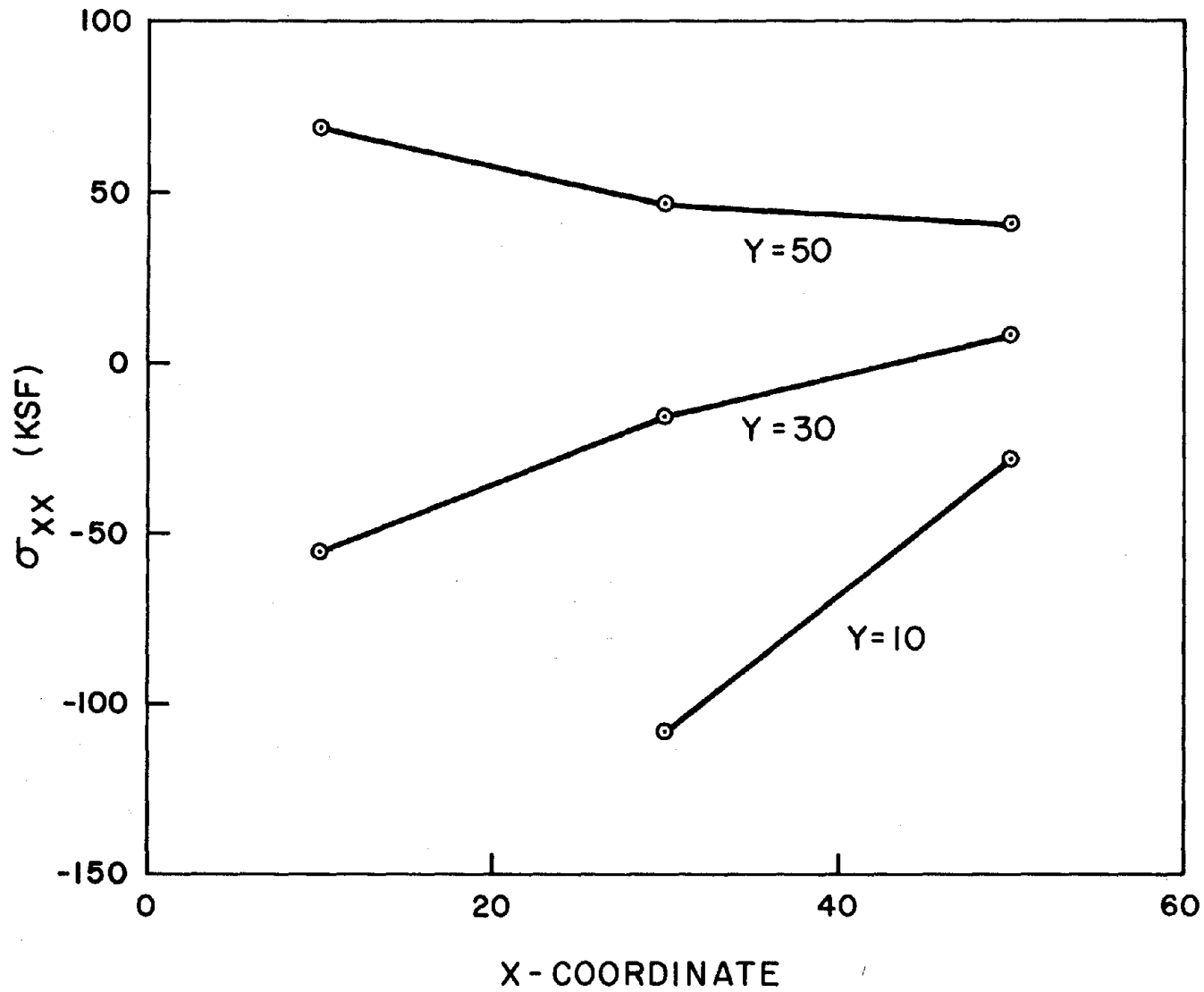


FIGURE E.3 STRESSES IN THE CEILING OF THE MINE ROOM (Z=25 FT)

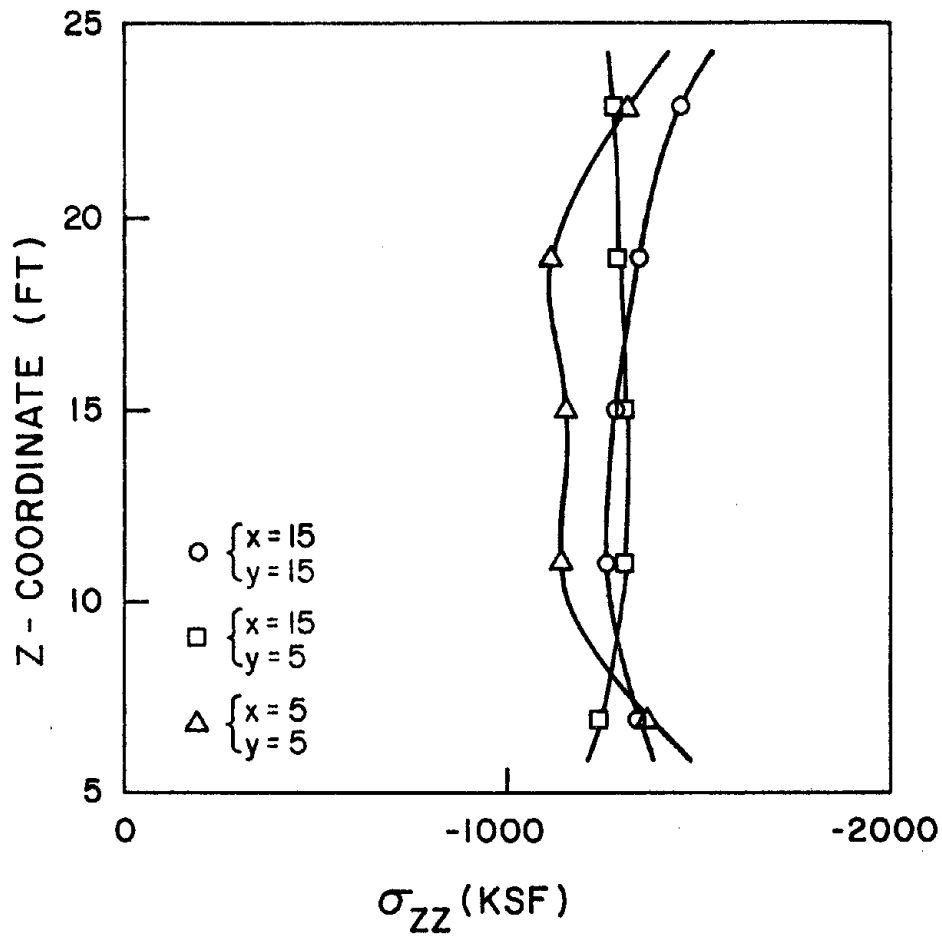


FIGURE E.4 STRESSES IN THE MINE PILAR

