REPORT NO.
69-5

STRUCTURES AND MATERIALS RESEARCH
DEPARTMENT OF CIVIL ENGINEERING

A COMPUTER PROGRAM FOR THE ANALYSIS OF THIN SHELLS

By
C. PHILIP JOHNSON
PETER G. SMITH

REPORT TO
U. S. NAVAL
CIVIL ENGINEERING LABORATORY

AUGUST, 1967

STRUCTURAL ENGINEERING LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY CALIFORNIA

A COMPUTER PROGRAM FOR THE ANALYSIS OF THIN SHELLS

by

C. Philip Johnson

Peter G. Smith

University of California
Department of Civil Engineering
Berkeley, California

January, 1969

ABSTRACT

The finite element method is used for the analysis of thin shells of arbitrary geometry. The shell surface is idealized by an assemblage of flat triangular elements, and the stiffness properties of each element are evaluated. The stiffness of the complete assemblage is obtained by the direct stiffness procedure.

The computer program is suitable for the static analysis of linear shells subjected to arbitrary surface pressure loadings and concentrated nodal forces. Displacement boundary conditions can also be specified.

The use of the program and a listing of the FORTRAN IV program for the CDC 6400 are given in the report. Also, a discussion of the analysis of a toroidal shell with meridional stiffeners is included as an example.

ACKNOWLEDGMENT

The work described in this report was carried out under Contract No. N62399-68-0035 between the University of California and the United States Naval Civil Engineering Laboratory at Port Hueneme, California. The project was conducted under the direction of Professors Ray W. Clough and Edward L. Wilson.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

The widespread use of thin shell structures has created a need for a systematic method of analysis which can adequately account for abritrary geometric form and boundary conditions. Classical thin shell theory yields differential equations of equilibrium or continuity which have been solved for only special geometric forms and certain boundary conditions. This is evidenced by the extensive study of spheres and circular cylinders using classical solution procedures.

A completely general approach for the solution of problems in continium mechanics was introduced in the late fifties and later became known as the finite element method [1]. This method was first applied to the solution of plane stress problems and subsequently was extended to the analysis of plates and shells [2]. The advantages of the finite element method are many. With the finite element solution procedure one can accommodate arbitrary geometry and boundary conditions together with variable thickness, variable material properties, discontinuities in the shell surface (cut-outs), and general loading.

The present report deals with the development of a general computer program for the static analysis of linear shells of arbitrary geometry and loading. The shell surface is discretized by an assemblage of flat triangular elements. In addition, a quadrilateral element, composed of four triangular elements, is possible. The material properties of the individual elements are assumed to be homogeneous, isotropic and linearly elastic. Automatic data generation options reduce and simplify the task of supplying input data to the program.

A toroidal shell with meridional stiffeners is analyzed in order to illustrate the use of the program. Consideration is given to the selection of a mesh which could adequately simulate the behavior of the shell. The selection of a nodal point numbering system is also discussed.

2. METHOD OF ANALYSIS

2.1 Discretization of Shell

The basic concept of the finite element method is the idealization of the continuum as an assemblage of discrete structural elements. In the analysis of thin shells, the finite element idealization consists of the geometric discretization and the displacement field discretization. The geometric discretization is due to the use of planar triangular elements to approximate the actual smoothly curved surface of the shell (Fig. 2.1). In addition, since the boundaries of these elements are straight, curved shell boundaries are also represented approximately. The displacement field discretization is caused by evaluating the stiffness properties of the individual elements from an assumed set of displacement shapes which only approximate the actual deformation of the shell.

The effects of the discretization error have been shown to diminish with mesh refinement using the analysis procedure described herein [2]. Hence, with adequate mesh refinement the finite element solution can be expected to be a satisfactory one.

2.2 Element Stiffnesses

Since shell behavior is characterized by both membrane action and bending action, it is essential to recognize both of these in evaluating the element stiffness properties.

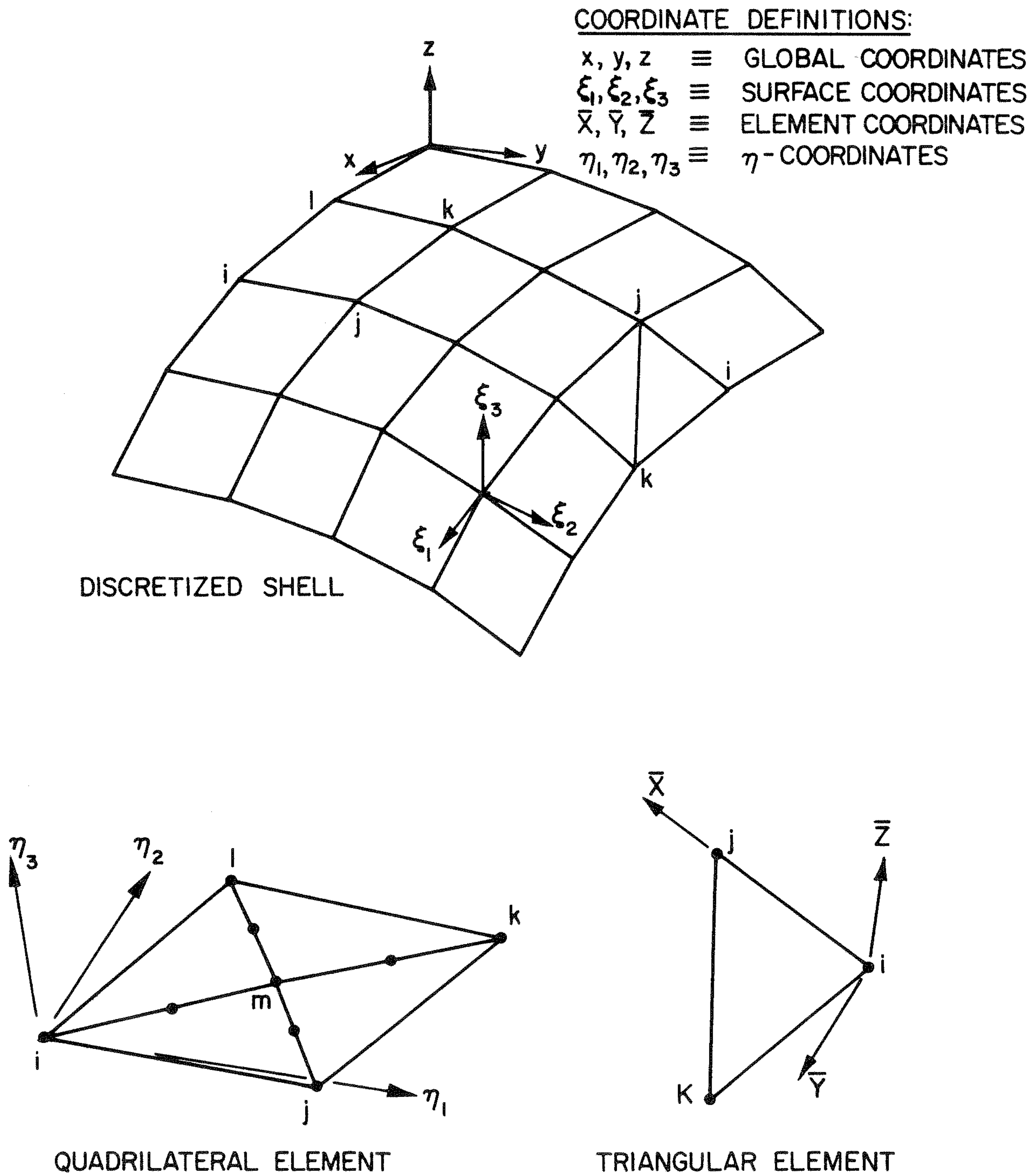


FIG. 2.1 DISCRETIZED SHELL WITH TYPICAL TRIANGULAR AND QUADRILATERAL ELEMENTS AND COORDINATE SYSTEMS

Two types of finite elements are available in the program, both of which include membrane and bending stiffnesses (Fig. 2.1):

a. Triangular element

Membrane stiffness ... Constant strain triangle [1]

Bending stiffness ... Fully compatible plate bending element after Hsieh, Clough and Tocher [3]

b. Non-planar Quadrilateral element

Membrane stiffness ... An assemblage of four linear strain triangles with linear displacements along exterior sides [4]

Bending stiffness ... An assemblage of four bending elements as per **a.** above

The superior stiffness properties of the quadrilateral versus the triangle motivates the general use of the quadrilateral.

The stiffness properties of the complete structure are obtained by superposition of the individual element stiffnesses using the direct stiffness procedure. This gives a system of linear equilibrium equations in terms of nodal point loads and displacements. The equilibrium equations are characterized by a symmetric positive definite matrix that can be expressed in a banded form. Hence, they may be solved for the unknown nodal displacements with a minimum of storage and computer time. A direct method, a modification of Gaussian elimination, was used to solve the equilibrium equations.

2.3 Coordinate Systems

A global coordinate system x, y, z (Fig 2.1) must be chosen for the shell which is to be analyzed. Although this choice is arbitrary, simplification of input nodal coordinates usually dictates the proper orientation for this coordinate system.

In addition, another set of coordinates ξ_1, ξ_2, ξ_3, called surface coordinates, must be selected. This coordinate system is characterized by the fact that ξ_3 is normal to the shell surface at each nodal point, while ξ_1 and ξ_2 are tangent to the shell surface at each nodal point. Surface generators may be conveniently used to describe this coordinate system. For example, in Fig. 2.2 the surface generators are taken as the straight line generators, and the required input at node I is the direction cosines of ξ_1 and $\bar{\xi}_2$ with respect to the global coordinates. The normal, ξ_3, is automatically constructed by a cross-product of ξ_1 and $\bar{\xi}_2$, and ξ_2 is then determined by the cross-product of ξ_3 and ξ_1 to insure a right-handed orthogonal system. Care should be taken to input ξ_1 and $\bar{\xi}_2$ so as to consistently have an outward normal, ξ_3.

Average plane coordinates (n-coordinates) are generated within the computer program for use with quadrilateral elements. The N_1-N_2 plane is the plane which "best fits" the coordinates of the exterior nodes (i,j,k,l), and is established by minimizing the sum of the squares of the normal distances from this plane to the exterior nodes. This plane is established automatically from the coordinates of the exterior nodes, and the N_1 axis is determined from a plane normal to the N_1-N_2 plane which contains side i-j of the quadrilateral element.

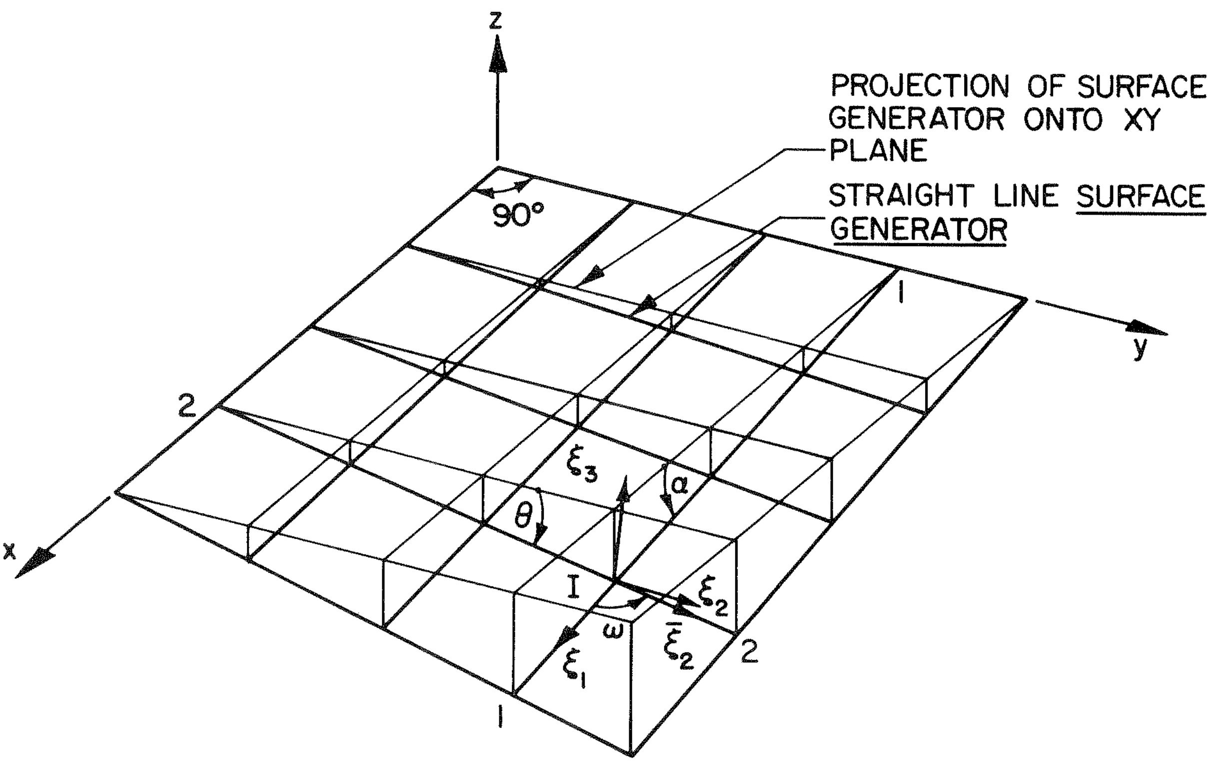


FIG. 2.2 SURFACE COORDINATES DEFINED USING STRAIGHT LINE SURFACE GENERATORS

Element stress resultants computed for a given element are expressed in this coordinate system. In addition, pressure loads are assumed to be positive if they act in the N_3 direction. As a result, element nodal point numbers i, j, k, l, should be numbered in a counter clockwise direction about the normal to the shell surface in order to ensure that the positive direction for pressure loads coincides with the direction of the outward normal to the shell surface.

A five degree of freedom nodal point displacement system for the assemblage is utilized. These five degrees of freedom consist of three linear translations and two rotations, and are defined as follows:

D1 $\equiv$ Translation in either global x-direction or surface ξ_1-dir.

D2 $\equiv$ Translation in either global y-direction or surface ξ_2-dir.

D3 $\equiv$ Translation in either global z-direction or surface ξ_3-dir.

D4 $\equiv$ Rotation about ξ_1 coordinate.

D5 $\equiv$ Rotation about ξ_2 coordinate.

It should be noted that all translations are either in global coordinates or surface coordinates.

Base coordinates are defined as the coordinates in which the five degrees of freedom at each nodal point of the assemblage are expressed. From the above description it is evident that the two options for base coordinates are:

a) Global coordinates for the three translations and surface coordinates for the two rotations.

b) Surface coordinates for both the three translations and two rotations.

The two rotation quantities are always referenced with respect to the surface coordinate system. These rotations are defined by vectors lying in the tangent plane to the shell surface. Hence, surface coordinate direction cosines at each node must be supplied as input data to the computer program in order to define the tangent plane.

The third rotation quantity, the rotation about the normal to the tangent plane, is considered small in comparison with the other two rotations and is neglected in the analysis procedure. The technique that is used to eliminate this rotation is equivalent to assigning a zero stiffness to it. It was assumed that this treatment of the rotation about ξ_3 would be negligible in the analysis, and results from numerous examples have verified this assumption [2].

If a coordinate system other than the one defining the tangent plane were used as base coordinates for rotations, three rotation quantities would have to be considered at each nodal point. This would present no problems provided all the elements meeting at a particular nodal point do not lie in the same plane. If that should be the case, then the stiffness matrix for the assemblage of elements would be singular and no solution of the equilibrium equations could be obtained.

3. COMPUTER PROGRAM USAGE

IDENTIFICATION

Analysis of Thin Shells

Programmed -- C. Philip Johnson, University of California, Berkeley

PURPOSE

The purpose of this finite element computer program is to provide a general capability for the analysis of thin shells of arbitrary geometry. The analysis includes the determination of nodal displacements as well as element and nodal stress resultants. Bending and membrane stiffnesses are considered in the analysis. Arbitrary loading and boundary conditions, as well as variable element thickness and material properties, may be accounted for.

3.1 Input Data

The first step in the finite element analysis of a thin shell is to select a global coordinate system x, y, z and a surface coordinate system ξ_1, ξ_2, ξ_3. Next a finite element mesh is obtained by subdividing the shell surface into a finite number of quadrilateral or triangular regions (elements). Although the exact proportions of the individual elements are arbitrary, care should be taken to ensure that the element proportions do not become overly exaggerated. Elements and nodal points are then numbered in two numerical sequences, each starting with one, and the input data is prepared as described on the following pages.

Abbreviations:

A = alphanumeric field

I = integer value (must be packed to the right of the field).

F = floating point number (must be punched with a decimal).

1. TITLE CARD (A) - alphanumeric information for problem identification.

2. CONTROL CARD - (4I5, 5X, I5)

Cols.	1- 5(I)	NUMEL	Number of elements (400 max)
	6-10(I)	NUPTS	Number of nodal points (400 max)
	11-15(I)	NUBPTS	Number of nodal points with displacement B.C. (100 max)
	16-20(I)	IBANDP	Nodal point half band width: Max element nodal difference + 1 (20 max)
	21-30(I)	IFLAG	Specifies base coordinates for translations; if IFLAG = 0, translations are in global coordinates. if IFLAG = 1, translations are in surface coordinates.

3. NODAL COORDINATE CARDS - (I4,6X,3F10.0)

One card per nodal point. Nodal coordinate cards must be input in numerical sequence corresponding to the nodal point numbering.

Cols.	1- 4(I)	Nodal point number.
	11-20(F)	Global x-coordinate.
	21-30(F)	Global y-coordinate.
	31-40(F)	Global z-coordinate.

4. SURFACE COORDINATE DIRECTION COSINE CARDS - (I4, 6X, 6F10.0)

One card per nodal point. Surface coordinate direction cosine cards need not be input in numberical sequence. A blank card must be used to terminate this data set.

Cols.	1- 4(I)	Nodal point number I.	or blank*
	11-20(F)	Component in global x-dir. of unit vector ξ_1	
	21-30(F)	Component in global y-dir. of unit vector ξ_1	

Cols. 31-40(F) Component in global z-dir. of unit vector ξ_1 } or blank*

41-50(F) Component in global x-dir. of unit vector $\bar{\xi}_2$

51-60(F) Component in global y-dir. of unit vector $\bar{\xi}_2$ } or blank**

61-70(F) Component in global z-dir. of unit vector $\bar{\xi}_2$

*If Cols. 11-40 are left blank, then the input cosines for ξ_1 are suppressed.

**If Cols. 41-70 are left blank, then the input cosines for $\bar{\xi}_2$ are suppressed.

Suppression of input cosines for either ξ_1 or $\bar{\xi}_2$ may be useful when coordinate generation options are used.

5. ELEMENT NODAL POINT NUMBER CARDS - (5I4)

One card per element. Element nodal point number cards must be input in numerical sequence corresponding to element numbering.

Cols. 1- 4(I) Element number

5- 8(I) Element nodal point I

9-12(I) Element nodal point J

13-16(I) Element nodal point K

*17-20(I) Element nodal point L

*A triangular element is assumed if Cols. 17-20 are left blank.

6. ELEMENT MATERIAL PROPERTY CARDS - (I4, 6X, 3F10.0)

One card per element. Element material property cards must be input in numerical sequence corresponding to element numbering. Material properties are assumed constant over each individual element.

Cols. 1- 4(I) Element number

11-20(F) Modulus of element

21-30(F) Thickness of element

31-40(F) Poisson's ratio of element

7. <u>BOUNDARY CONDITION CARDS</u> - (I4, 5I1, 1X, 5F10.0)

One card per nodal point having a specified displacement component (whether zero or non-zero). Boundary condition cards need not be input in numerical sequence. The five degrees of freedom are ordered as follows:

Cols. 1-4(I) Nodal point number

5(I)= 1 for specified value for D1; 0 otherwise

6(I)= 1 for specified value for D2; 0 otherwise

7(I)= 1 for specified value for D3; 0 otherwise

8(I)= 1 for specified value for D4; 0 otherwise

9(I)= 1 for specified value for D5; 0 otherwise

Cols. 11-20(F) *Specified value for D1

21-30(F) Specified value for D2

31-40(F) Specified value for D3

41-50(F) Specified value for D4

51-60(F) Specified value for D5

*Specified value may be non-zero.

8. <u>CONTROL CARD FOR LOADS</u> - (4I5,F10.0, I5)

A maximum of three independent load cases for a single problem may be specified on this card.

Col. 5(I) Number of independent load cases

6-10(I) Number of loaded nodes for load case 1

11-15(I) Number of loaded nodes for load case 2

16-20(I) Number of loaded nodes for load case 3

21-30(F) Uniform pressure load UPL (+ in direction of N_3, Fig 2.1)

31-35(I) Number of elements having a pressure different from UPL.

Loaded nodes are joints where concentrated point loads are applied. UPL is assumed to act on all elements of the assemblage if Col.'s 31-35 are left blank or zero. Only linear nodal force components resulting from the pressure loading are considered. The pressure loading will be applied in load Case 1 only. Hence, input nodal forces which are designated as load Case 1, will be superimposed on the pressure loading.

9. ELEMENT PRESSURE CARDS - (I5, F10.0)

One card per element having a normal pressure different than that specified as UPL on the control card for loads.

Cols. 1- 5(I) Element Number

6-15(F) Pressure normal to surface of element

Sign convention: Pressures are positive when they act in the same direction as N_3.

10. NODAL POINT LOAD CARDS - (I3,7X,5F10.0)

Input nodal point forces correspond in an energy sense to the nodal point displacement components, i.e., P_iD_i, i = 1,5. The number of input cards for each load case must be equal the specified number in (8) above. The input format for each load case is the following:

Cols. 1- 3(I) Nodal point number

11-20(F) Value of P_1 which corresponds to D1

21-30(F) Value of P_2 which corresponds to D2

31-40(F) Value of P_3 which corresponds to D3

41-50(F) Value of P_4 which corresponds to D4

51-60(F) Value of P_5 which corresponds to D5

These points need not be in nodal point numerical sequence, but all cards for each load case must be grouped together, in the sequence: Load Case 1,2,3.

3.2 Automatic Generation Options

The previous section provided a general input format by which all the information which is necessary to completely define a given problem is input via data cards. The generation options described below are intended to simplify and to reduce the amount of that required input.

1. NODAL COORDINATE GENERATION

Five geometry types are available. The first four types treat surface generators which occur frequently in shell structures: Straight lines, circular arcs, parabolas, and ellipses. Sequential numbering along individual generators is required in order to utilize these features. Two cards designated as Card 1 and Card 2, sequentially placed in the set of nodal coordinate cards (3) are required for the utilization of these types of coordinate generation described below.

The fifth type of generation described below is useful when the coordinates of a set of sequentially numbered points may, by constant increments, be defined from a previous set of sequentially numbered points. Only one card placed in the set of nodal coordinate cards is required to utilize this type of coordinate generation.

If any of these five types of generation are used, then two sets of direction cosines are generated. The procedure for the generation of these direction cosines is described below for each type of coordinate generation. In many cases, these generated direction cosines will correspond with the chosen surface coordinates ξ_1 and ξ_2, thereby eliminating the need of inputing these direction cosines; in other cases the generated direction cosines will have to be replaced by manually computed values.

a. Type 1 -- Straight line (see Fig. 3.1)

Card 1 -- Cols.	1- 4(I)	Node I of straight line
	8(I)	= 1
	11-20(F)	Global x-coordinate of point I.
	21-30(F)	Global y-coordinate of point I.
	31-40(F)	Global z-coordinate of point I.
Card 2 -- Cols.	1- 4(I)	Node J of straight line (J > I).
	11-20(F)	Global x-coordinate of point J
	21-30(F)	Global y-coordinate of point J
	31-40(F)	Global z-coordinate of point J

The straight line is subdivided into (J - I) equal parts and the intermediate global nodal coordinates are computed.

The two sets of direction cosines are computed by:

a1) Assuming ξ_1 is in the direction from point I to point J.

a2) Assuming ξ_2 lies in the x-y plane and is normal to the line obtained by projecting line I-J onto the x-y plane, i.e., $\xi_2 \equiv \bar{y}$ (Fig. 3.1), and by ensuring a right-handed system for $\bar{x},\bar{y},\bar{z}$. For example, if the numbering had required a reversed direction for $\bar{x}$ and hence ξ_1, then $\bar{y}$ and ξ_2 would be reversed to order to maintain a right-handed system without changing the direction of ξ_3. In the case where ξ_1 is parallel to z (either direction), then ξ_2 is assumed to be in the same direction as the global y-coordinate.

b. Type 2 -- Circular arc (see Fig. 3.2).

Card 1 -- Cols.	1- 4(I)	Node I of circular arc
	8(I)	= 2

Card 1 -- Cols.	11-20(F)	Global x-coordinate of point I
	21-30(F)	Global y-coordinate of point I
	31-40(F)	Global z-coordinate of point I
	41-50(F)	Global x-coordinate of point J
	51-60(F)	Global y-coordinate of point J
	61-70(F)	Global z-coordinate of point J
Card 2 -- Cols.	1- 4(I)	Node J of circular arc (J > I)
	11-20(F)	Global x-coordinate of point m
	21-30(F)	Global y-coordinate of point m
	31-40(F)	Global z-coordinate of point m

The circular arc is subdivided into (J - I) parts of equal arc length and the intermediate global nodal coordinates are computed. The local right-handed cartesian coordinate system $\bar{x}$, $\bar{y}$, $\bar{z}$ is constructed as follows:

1. $\bar{x}$ is positive from I to J where J > I.
2. $\bar{z}$ is positive from n, a point equidistant from I and J, to point m, the mid-point of the circular arc.
3. $\bar{y}$ is established by a cross-product of $\bar{x}$ and $\bar{z}$ (+ inward as shown in Fig. 3.2).

Surface coordinate direction cosines are computed by assuming:

1. ξ_1 lies in the plane $\bar{x}$ - $\bar{z}$ and is tangent to the circular arc at each node. It is directed along the arc going from I to J.
2. ξ_2 is assumed to be in the positive direction of $\bar{y}$. Note that if the nodal point numbers had increased from right to left in Fig. 3.2, $\bar{y}$ and hence ξ_2 would be positive outward.

The circular arc may have an arbitrary orientation w.r.t. the global coordinates.

c. Type 3 -- <u>Parabola</u> (see Fig. 3.3).

Card 1 -- Cols.	1- 4(I)	Node I of parabola
	8(I)	= 3
	11-20(F)	Global x-coordinate of origin point 0
	21-30(F)	Global y-coordinate of origin point 0
	31-40(F)	Global z-coordinate of origin point 0
	41-50(F)	Counterclockwise angle ω (in degrees) from x to $\bar{x}$.
Card 2 -- Cols.	1- 4(I)	Node J of parabola (J >I)
	11-20(F)	Local $\bar{x}$ coordinate of point I (negative for Fig. 3.3)
	21-30(F)	Local $\bar{x}$ coordinate of point J
	31-40(F)	Largest absolute value of $\bar{z}_i$ and $\bar{z}_j$

The horizontal distance between I and J is subdivided into J - I equal intervals and the intermediate global nodal coordinates are computed.

The local coordinate system $\bar{x},\bar{y},\bar{z}$ is constructed as follows:

1. $\bar{x}$ is positive in the direction from I to J
2. $\bar{z}$ is parallel and in the same direction as Z.
3. $\bar{y}$ forms a counterclockwise angle of $\omega + 90°$ from the global x-axis.

Surface coordinate direction cosines are computed by assuming:

1. ξ_1 lies in the plane $\bar{x}$ - $\bar{z}$ and is tangent to the parabola at each node. It is directed along the parabola going from I to J.
2. ξ_2 is parallel and in the same direction as $\bar{y}$.

d. Type 4 -- Ellipse (see Fig. 3.4)

Card 1 -- Cols.	1- 4(I)	Node I of ellipse
	8(I) = 4	
	11-20(F)	Global x-coordinate of origin point 0
	21-30(F)	Global y-coordinate of origin point 0
	31-40(F)	Global z-coordinate of origin point 0
	41-50(F)	Counter clockwise angle from x to $\bar{x}$
	51-60(F)	Distance, a, from 0 to ellipse along $\bar{z}$
	61-70(F)	Distance, b, from 0 to ellipse along $\bar{x}$
Card 2 -- Cols.	1- 4(I)	Node J of ellipse
	11-20(F)	Local $\bar{x}$-coordinate of point I
	21-30(F)	Local $\bar{x}$-coordinate of point J

The ellipse arc length between nodes I and J is subdivided into J-I equal arc lengths and the intermediate global nodal coordinates are computed.

The local coordinate system $\bar{x},\bar{y},\bar{z}$ is constructed as follows:

1. $\bar{x}$ is positive in the direction from I to J
2. $\bar{z}$ is parallel and in the same direction as z
3. $\bar{y}$ forms a counterclockwise angle of $\omega + 90°$ from the global x-axis

Surface coordinate direction cosines are computed by assuming:

1. ξ_1 lies in the plane $\bar{x}$ - $\bar{z}$ and is tangent to the ellipse at each node. It is directed along the parabola going from I to J.
2. ξ_2 is parallel and in the same direction as $\bar{y}$.

Restriction: $\bar{z}$ must be parallel to the global z axis and angle IOJ$\leq$180°.

e. Type 5 -- Incremental nodal coordinates - (2I4,2x,3F10.0)

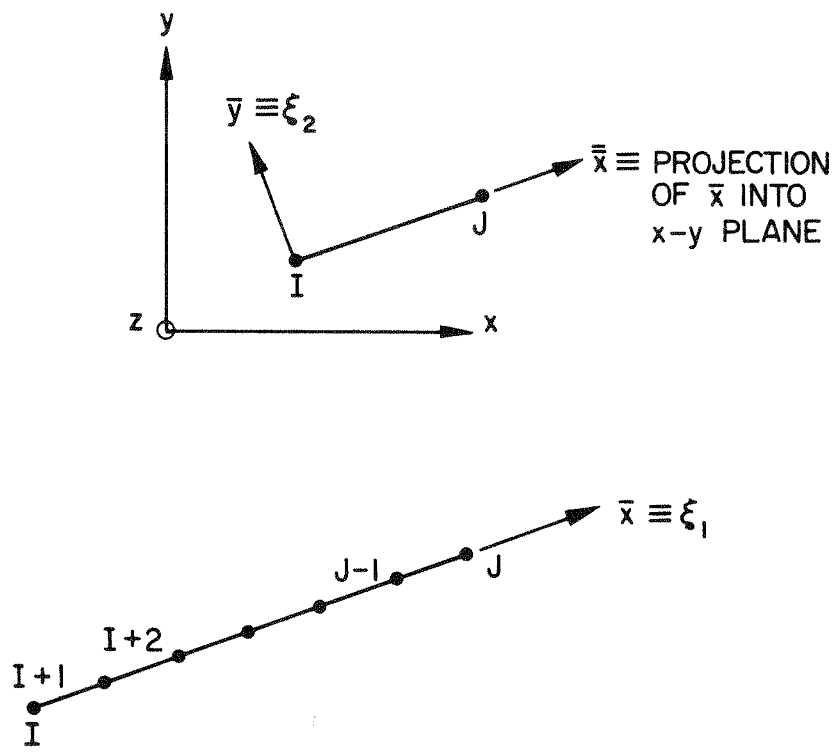
Cols.	1- 4(I)	Node I
	5- 8(I)	Node J preceded by a minus sign
	11-20(F)	Global x-increment ≡ XINC
	21-30(F)	Global y-increment ≡ YINC
	31-40(F)	Global z-increment ≡ ZINC

This card causes the generation of global coordinates of points I through J as a function of the preceding J - I + 1 nodes M and N where M = 2I - J - 1 and N = I - 1 as:

$$
\begin{array}{llll}
X_I = X_M + XINC & & Y_I = Y_M + YINC & & Z_I = Z_M + ZINC \\
X_{I+1} = X_{M+1} + XINC & & Y_{I+1} = Y_{M+1} + YINC & & Z_{I+1} = Z_{M+1} + ZINC \\
\cdot \quad \cdot \quad \cdot & & \cdot \quad \cdot \quad \cdot & & \cdot \quad \cdot \quad \cdot \\
\cdot \quad \cdot \quad \cdot & \text{and} & \cdot \quad \cdot \quad \cdot & \text{and} & \cdot \quad \cdot \quad \cdot \\
\cdot \quad \cdot \quad \cdot & & \cdot \quad \cdot \quad \cdot & & \cdot \quad \cdot \quad \cdot \\
X_{J-1} = X_{N-1} + XINC & & Y_{J-1} = Y_{N-1} + YINC & & Z_{J-1} = Z_{N-1} + ZINC \\
X_J = X_N + XINC & & Y_J = Y_N + YINC & & Z_J = Z_N + ZINC
\end{array}
$$

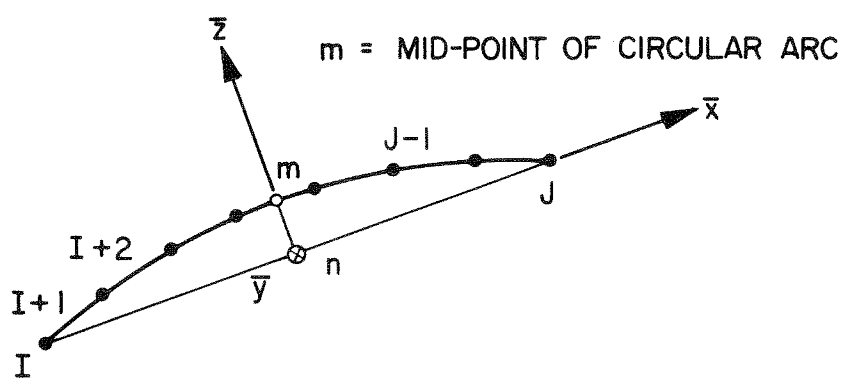
When this option is used, the orientation of the surface coordinates ξ_1 and ξ_2 for points I through J is assumed to be as follows:

$$
\begin{array}{lll}
(\xi_1)_I \equiv (\xi_1)_M & & (\xi_2)_I \equiv (\xi_2)_M \\
(\xi_2)_{I+1} \equiv (\xi_1)_{M+1} & & (\xi_2)_{I+1} \equiv (\xi_2)_{M+1} \\
\cdot \qquad \cdot & & \cdot \qquad \cdot \\
\cdot \qquad \cdot & \text{and} & \cdot \qquad \cdot \\
\cdot \qquad \cdot & & \cdot \qquad \cdot \\
(\xi_1)_{J-1} \equiv (\xi_1)_{N-1} & & (\xi_2)_{J-1} \equiv (\xi_2)_{N-1} \\
(\xi_1)_J \equiv (\xi_1)_N & & (\xi_2)_J \equiv (\xi_2)_N
\end{array}
$$



EQUAL INCREMENTS ALONG STRAIGHT LINE

FIG. 3.1 STRAIGHT LINE



EQUAL INCREMENTS ALONG CURVE

FIG. 3.2 CIRCULAR ARC

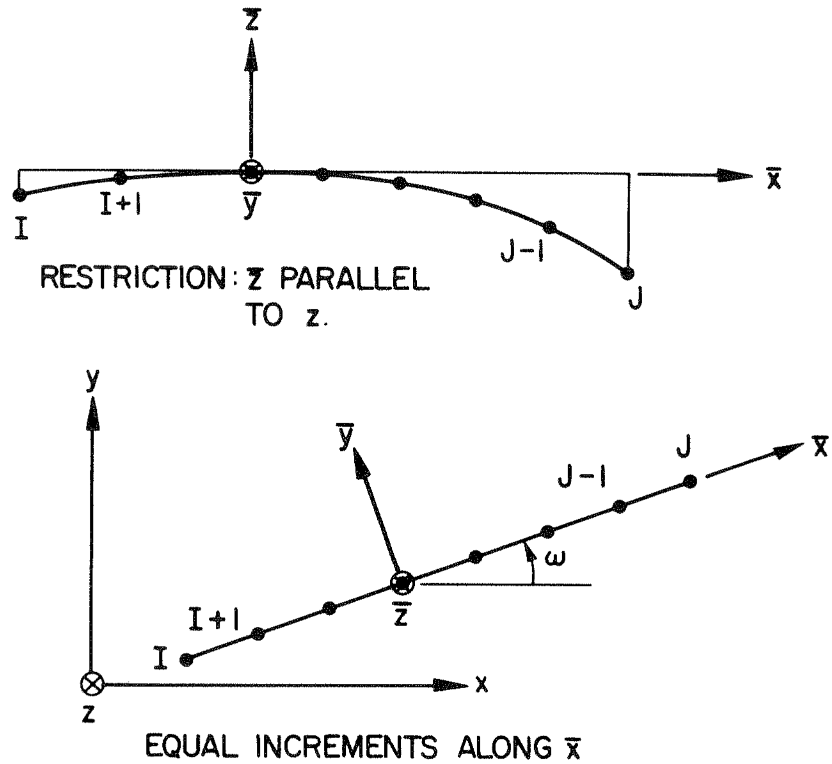


FIG. 3.3 PARABOLA

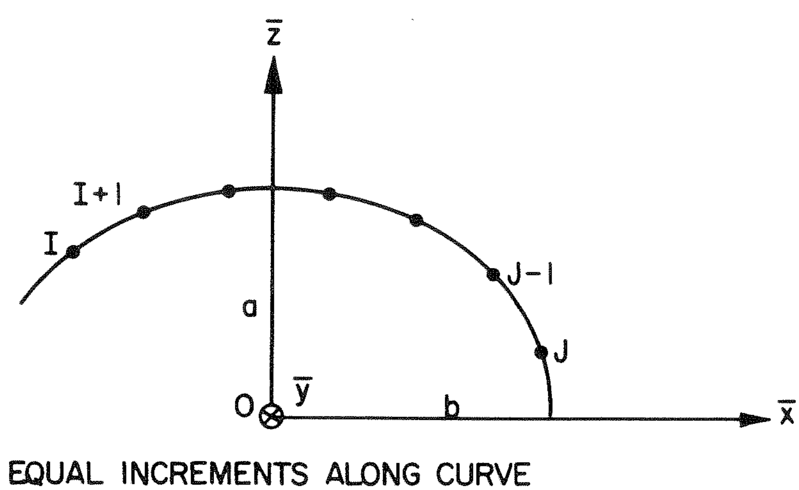


FIG. 3.4 ELLIPSE

2. SURFACE COORDINATE DIRECTION COSINE GENERATION

When nodal coordinate generation options are used assumed orientations are assigned to ξ_1 and $\bar{\xi}_2$. No distinction was made between ξ_2 and $\bar{\xi}_2$ in the previous section, Nodal Coordinate Generation, since $\bar{\xi}_2$ is constructed perpendicular to ξ_1.

If the orientations assigned for either ξ_1 or $\bar{\xi}_2$ differ from the desired directions or if nodal coordinate generation options do not apply for all nodes, surface coordinate direction cosines must be supplied as input data.

Surface coordinate direction cosines may be input for arbitrary nodal points as described in (4); however, the following option may considerably reduce this input. This option is exercized by preparing cards as described in (4), with the following modifications, and including them in the surface coordinate direction cosine cards.

Cols.	1- 4(I)	Nodal point number I
	11-70(F)	See (4) page 11
Cols.	73-76(I)	MOD
	77-80(I)	LIM

This option sets the direction cosines of points

$$I + MOD,\ I + 2*MOD,\ \ldots,\ LIM$$

equal to these specified for point I.

ξ_1 generation is suppressed by leaving Cols. 11-40 blank.

$\bar{\xi}_2$ generation is suppressed by leaving Cols. 41-70 blank.

3. ELEMENT NODAL POINT NUMBER GENERATION

a. Type1-- If M element cards are omitted in (5) these missing elements will be generated by increasing the nodal numbers I,J,K,L of each of the preceding elements by 1.

b. Type 2 -- If on the card for element N we specify

Cols. 21-24(I) MOD = number of elements in direction of numbering.

Cols. 25-28(I) NLAY = number of layers with similar element nodal numbering.

A regular mesh will be constructed for elements N through (N + MOD * NLAY - 1). The element number for the initial element and its nodal numbers will be specified on this card and the node numbers I,J,K,L should be ordered as per Fig. 3.5.

The nodal point numbering should run in the direction with the smallest number of elements in order to minimize the nodal point half band width (Max. element nodal difference + 1) of the assemblage. The element numbering should follow the general path of the nodal point numbering to insure successful formation of the assemblage stiffness.

A regular mesh is defined as a mesh having the same number of subdivisions in two directions throughout a portion of the idealization. An example of a regular mesh and its assumed nodal and element numbering is illustrated in Fig. 3.5. Element nodal points I,J,K,L are numbered counter clockwise with node I having the smallest number as illustrated for element 1.

Regular meshes should be used when possible. This simplifies the required input, thereby reducing possibilities of error in preparing the input data. Moreover, regular meshes usually result in minimum nodal point connectivity (max. element nodal point difference), hence reducing the computation time for a given problem.

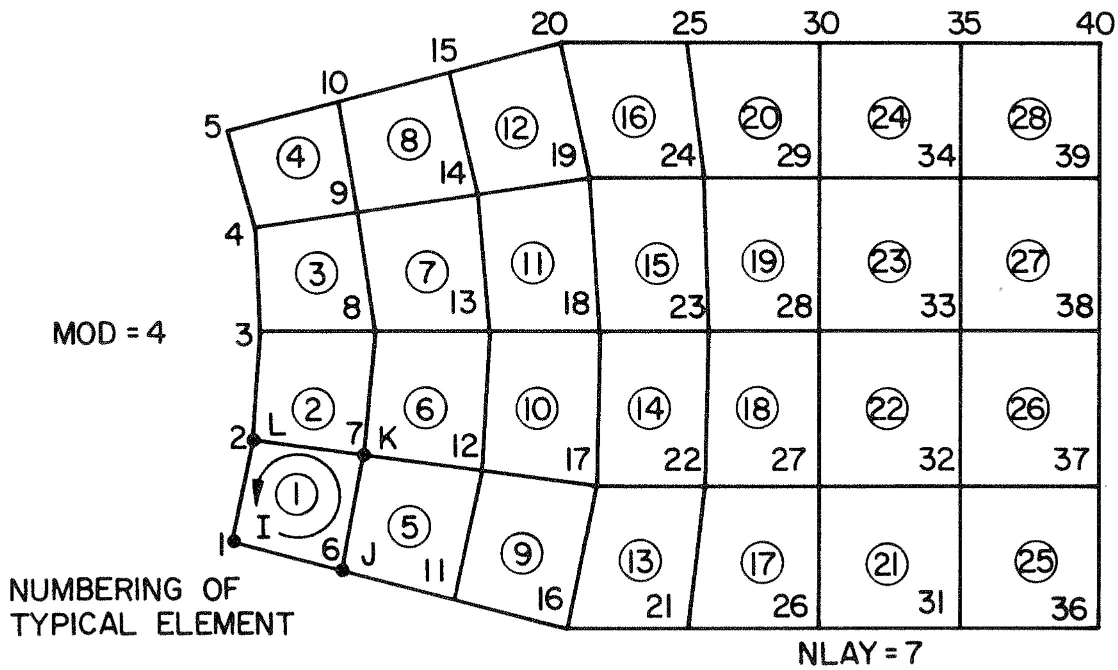


FIG. 3.5 REGULAR MESH WITH NODAL AND ELEMENT NUMBERING

4. ELEMENT MATERIAL PROPERTY GENERATION

If element material property cards N + 1, N + 2, ..., N + L - 1 are omitted, the modulus and poisson ratio of the missing L - 1 elements are assigned the values specified for element N, while the thickness is assumed to vary linearly between N and N + L.

5. BOUNDARY CONDITION GENERATION

If on a boundary condition card for node I we specify

Cols. 63-66(I) MOD > 0

67-70(I) LIM > I

the boundary conditions for points

I + MOD, I + 2MOD, ..., LIM

are set equal to those specified for point I.

6. LOAD GENERATION

Two options are available:

a. If on an element pressure card for element M (9) we specify

Cols. 63-66(I) MOD > 0

67-70(I) LIM > M

the pressure on elements

M + MOD, M + 2* MOD, ..., LIM

is set to that specified for element M.

b. If on a nodal point load card for node I (10) we specify

Cols. 63-66(I) MOD > 0

67-70(I) LIM > I

the nodal point loads for points

I + MOD, I + 2*MOD, ..., LIM

are set equal to those specified for point I.

3.3 Output Information

The following information is printed by the computer program:

1. Reprint of input data together with all data developed by automatic generation options.
2. Total applied nodal point forces for each load case. . . The forces P_i are in the base coordinate system and they correspond to displacement quantities D_i.
3. Nodal point displacements D_i in the base coordinate system for each load case.
4. Element stress resultants for each load case...these quantities are with respect to the average plane coordinate system (N - coordinates) defined in section 2.3 of this report. The sign convention for these quantities is identical to that illustrated in Fig. 3.6 for surface coordinates. These stress resultants can be assumed to be acting at the centroid of the quadrilateral element.
5. Averaged nodal stress resultants for each load case. . . the element stress resultants in all the elements surrounding a given node are averaged to obtain the nodal stress resultants. The nodal stress resultants are then printed with respect to the surface coordinate system ξ_i. These quantities are shown in Fig. 3.6 for a differential element in the surface of the shell.

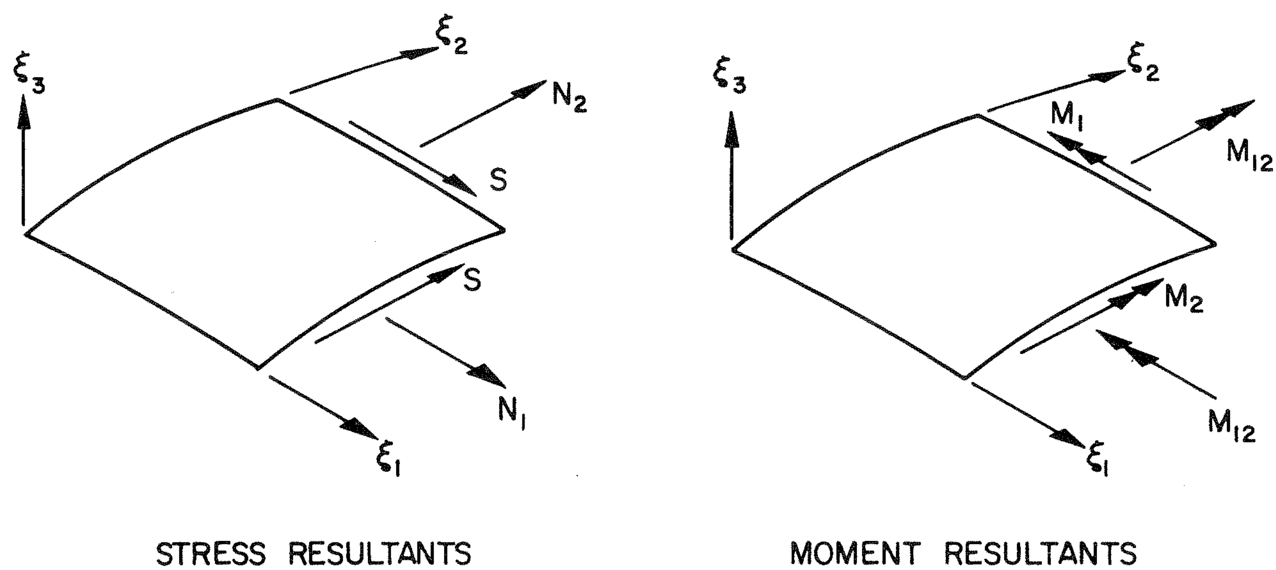


FIG. 3.6 SHELL STRESSES AND MOMENTS

4. PROGRAMMING INFORMATION

The program is written in FORTRAN IV. The program at various stages during its development has been successfully executed on the IBM 7094, IBM 7040, CDC 1604, CDC 6400, CDC 6600 and the Univac 1108 computers.

On-line input output is done via FORTRAN statements READ N, list, and PRINT N, list. FORTRAN logical units 1, 2, 3 and 4 are used for intermediate storage via FORTRAN statements WRITE (I) list and READ (I) list. The program consists of less than 2000 FORTRAN statements and is subdivided into a main program and 24 subroutines. Overlay features are not utilized.

In addition to the limitations regarding the number of nodal points, number of elements, etc., as discussed in the preceeding chapter, there is another limitation which can cause difficulties in some shell analysis problems. There is a limitation regarding the number of nodes contained in any one equilibrium equation of the complete structure stiffness matrix. No more than six (6) nodes can be involved in any equilibrium equation, I. Only the nodes larger than and including point I have to be considered in this count.

For example, in the mesh shown in Fig. 4.1, five (5) nodes are involved in the equilibrium equation for node 8. This number is within the limitation defined above, and, hence, the numbering scheme is acceptable.

On the other hand, the numbering scheme depicted in Fig. 4.2 is not acceptable since eight (8) nodes are involved in the equilibrium equation for node 5.

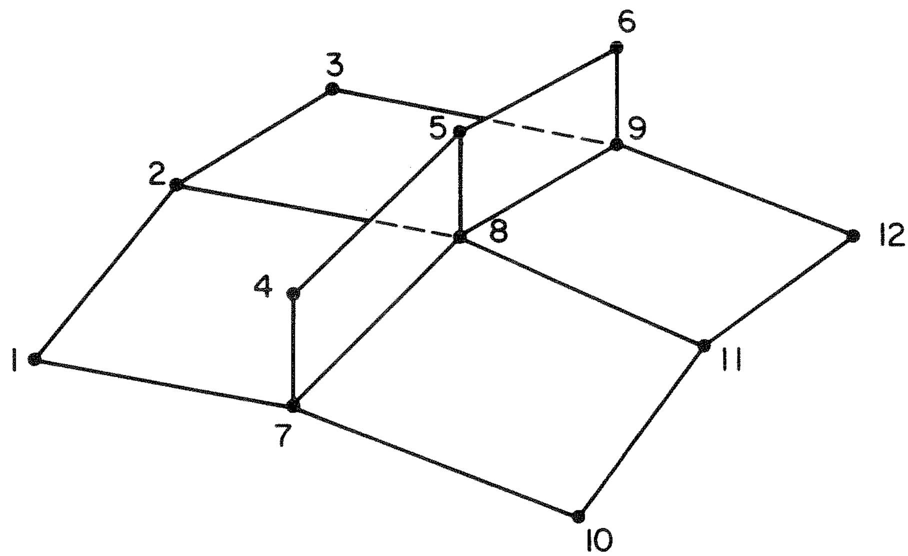


FIG. 4.1 ACCEPTABLE NODAL POINT NUMBERING SCHEME

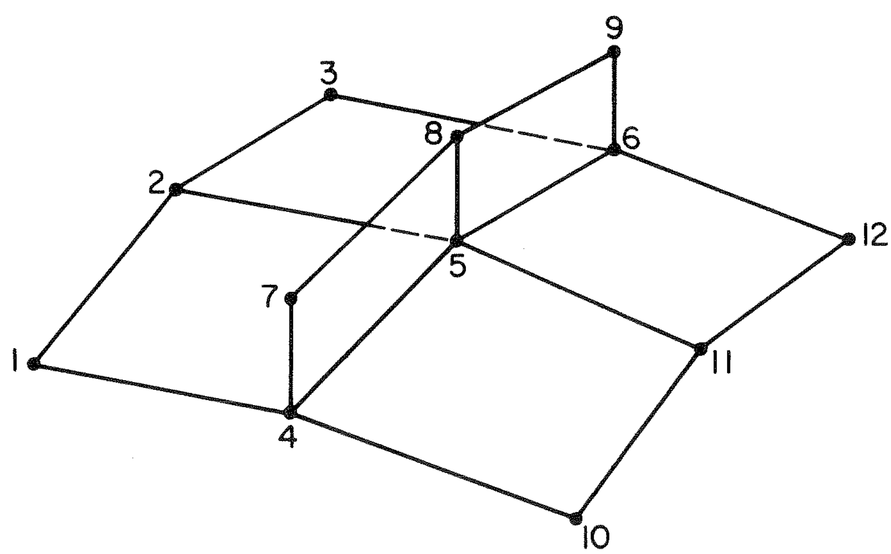


FIG. 4.2 UNACCEPTABLE NODAL POINT NUMBERING SCHEME

5. EXAMPLE -- TOROIDAL SHELL

Several examples with known exact solutions were solved to verify the method of analysis and to establish the validity of the computer program. Some of these examples were documented in another report [2]. In all cases, as the mesh was refined, the results approached the known values.

In order to illustrate the application of the program to a particular problem, the analysis of a toroidal shell with meridional stiffeners and subjected to a uniform external pressure was selected. The shell is pictured in Fig. 5.1.

In order to take advantage of symmetry, a 30° segment of the top half of the shell was selected for analysis. The mesh representation of the toroidal shell with the meridional stiffener is illustrated in Fig. 5.2. The nodal point numbering system shown was selected to take as much advantage of the automatic generation options as possible. Hence, input data to the program was kept to a minimum.

The control card contained the following information:

NUMEL	40
NUPTS	54
NUBPTS	26
IBANDP	20
IFLAG	1

Surface coordinates were selected as base coordinates for the translational degrees of freedom in order to facilitate the input of the displacement boundary conditions and to maintain the symmetry of the problem.

Ellipse nodal coordinate generation was employed to define the nodal coordinates (and surface coordinate direction cosines). Circle nodal coordinate generation could also have been used as an alternative. The directions for the generated surface coordinate vectors ξ_1 and ξ_2 are indicated in Fig. 5.2. These directions are satisfactory for all nodal points except those on the outer boundary of the meridional stiffener. It is possible to input direction cosines redefining the vectors ξ_1 and ξ_2 so that they lie in the surface of the stiffener for these points. However, since the rotation quantities associated with these directions are zero for this particular problem it makes little difference whether we redefine these quantities or not.

Element numbering was selected to follow the nodal point numbering. Three "regular meshes" were defined -- two 8 x 1 meshes and an 8 x 3 mesh. As a result only three input data cards were necessary to define the element nodal point numbering.

Since the same material properties were used throughout the entire shell, including the stiffener, only two cards were needed to define the thickness, elastic modulus and Poisson's ratio of all the elements. These quantities were input on cards for the first and last elements, and the computer program generated the material properties for all the intervening elements.

Special care had to be exercised when specifying the displacement boundary conditions for the problem. The symmetry of the problem was studied, and the appropriate displacement quantities were set to zero so that the analysis of the 30° segment selected would reflect the behavior of the entire structure. The selection of surface coordinates as base coordinates for translations made it possible to specify these skew displacement boundary conditions.

The control card for loads indicated that only one load case was to be considered. Since the loading consisted entirely of an external pressure (UPL = - 1 ksi), the number of loaded nodes was zero. Eight elements, comprising the stiffener, had a pressure different from UPL, and the pressure on these elements was set to zero with an element pressure card.

Nodal point load cards were not needed for this problem.

It should be noted that the dimensions for all quantities must be consistent. That is, all length and force quantities must be in the same units. This applies for the elastic modulus, E, as well as for the nodal coordinates x, y, z. For this example the kip is used as the unit for force and the inch is used for length. A listing of the data necessary to run this example can be found immediately following the program listing in the appendix.

Some of the output data for this example has been plotted in Fig. 5.3. Displacements are exaggerated since the scale used for deformation is different from that used to plot the toroidal shell. Nevertheless, an overall view of the behavior of the toroidal shell subject to the uniform external pressure can be seen. The plan view indicates that both the inside and outside diameters of the torus are reduced. The influence of the stiffener is depicted in the two section views. The distortion of the circular cross section of the torus is reduced significantly in the area of the shell immediately underneath the stiffener.

The toroidal shell was analyzed on the CDC 6400 at the University of California Computer Center at Berkeley. Approximately 32,000 words of central memory were needed for storage and the central processor time was 67 seconds.

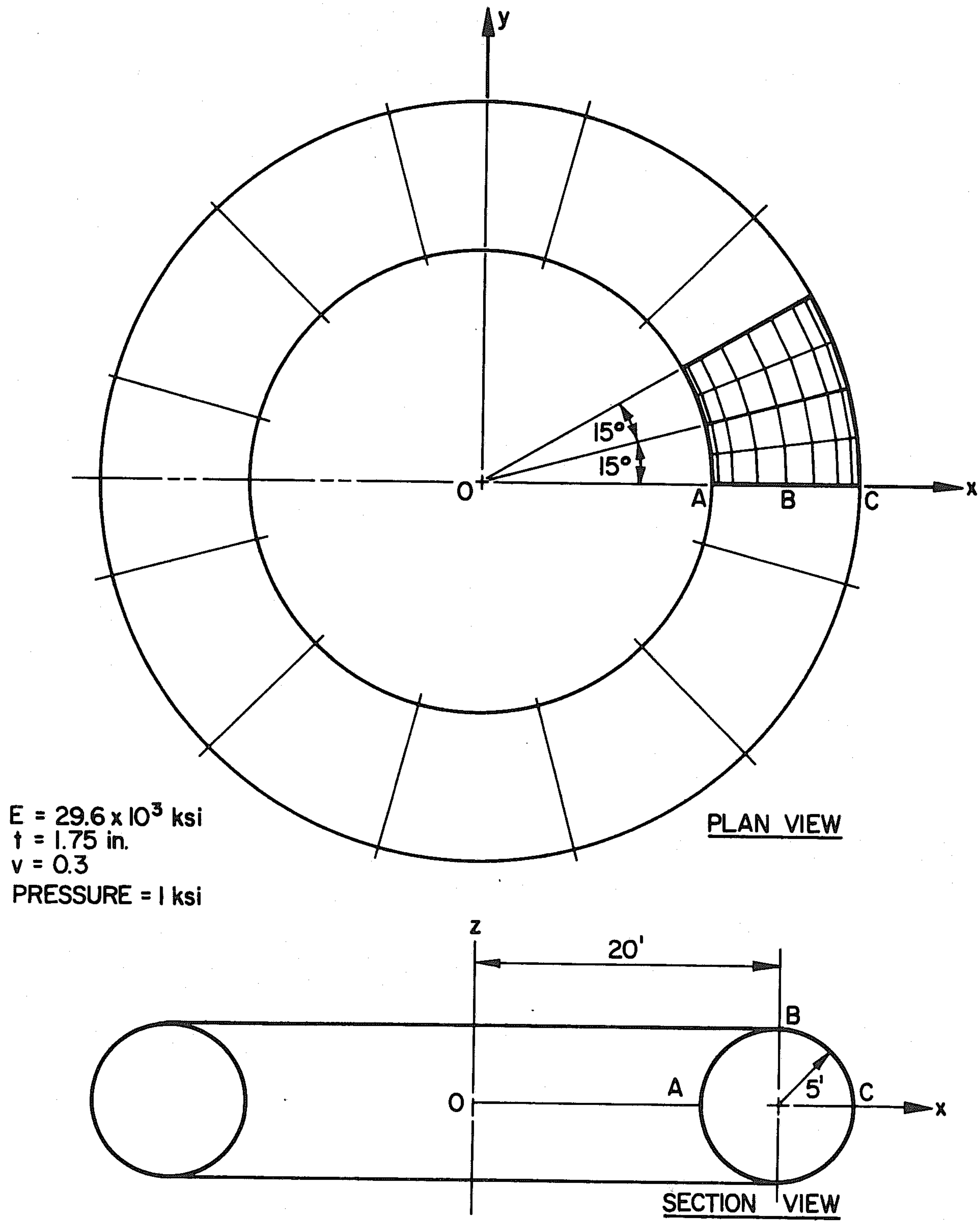


FIG. 5.1 TOROIDAL SHELL WITH MERIDIONAL STIFFENERS

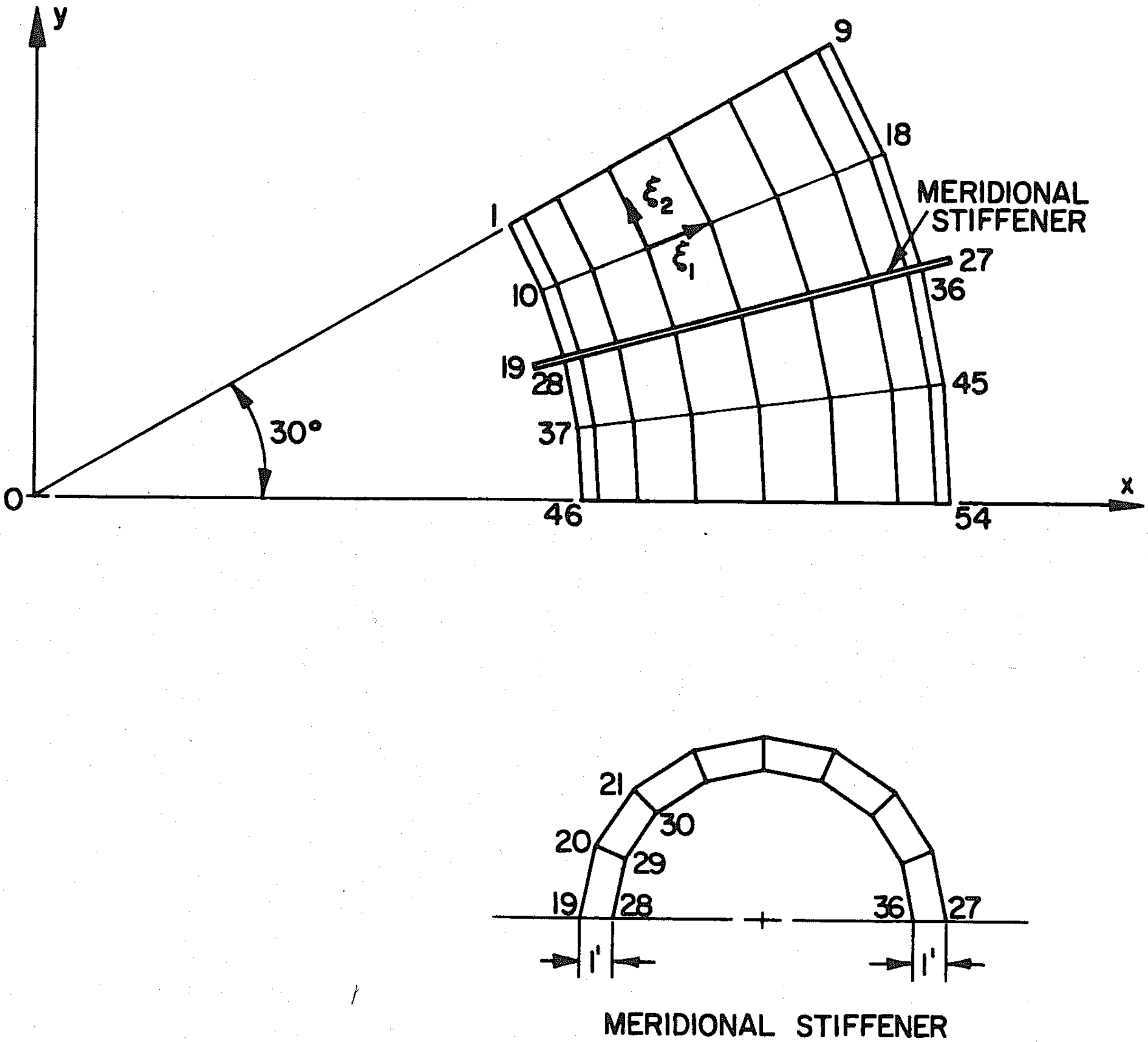
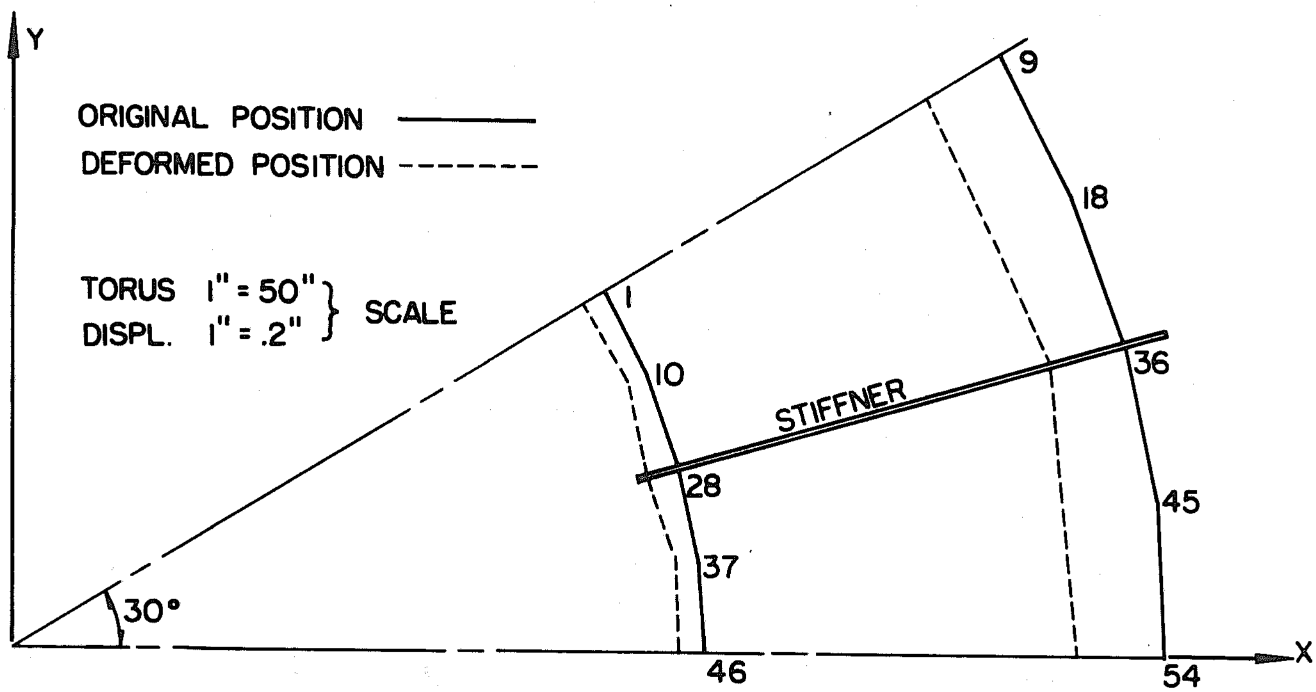
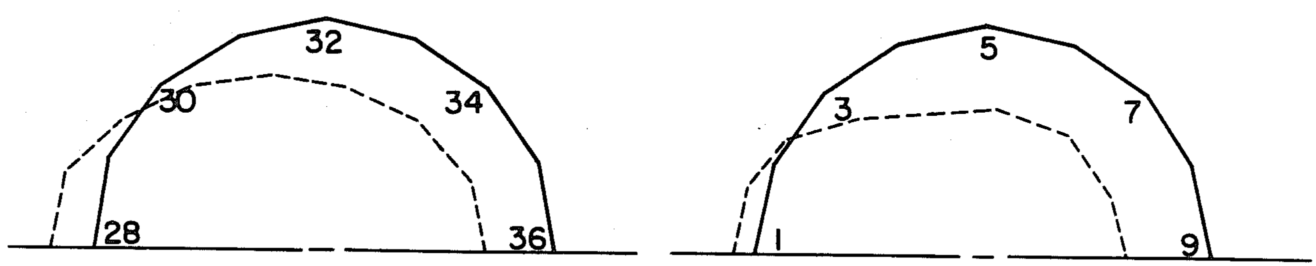


FIG. 5.2 MESH REPRESENTATION OF TORIODAL SHELL



PLAN VIEW



SECTION VIEWS

FIG. 5.3 DISPLACED POSITION FOR TORUS

REFERENCES

1. Turner, M.J., Clough, R.W., Martin, H.C., and Topp, L.J., "Stiffness and Deflection Analysis of Complex Structures", J. Aeron. Sci., V. 23, No. 9, 1956.

2. Johnson, C.P., "The analysis of Thin Shells by a Finite Element Procedure", Structural Engineering Laboratory Report, No. 67-22, University of California, Berkeley, 1967.

3. Clough, R.W and Tocher, J.L., "Finite Element Stiffness Matrices for the Analysis of Plate Bending", Proceedings , Conference on Matrix Methods in Structural Mechanics, Air Force Institute of Technology, Wright - Patterson Air Force Base, Ohio, October 1965.

4. Felippa, C.A., "Refined Finite Element Analysis of Linear and Nonlinear Two-Dimensional Structures," Structural Engineering Laboratory Report No. 66-22, University of California, Berkeley, 1966.

APPENDIX

COMPUTER PROGRAM LISTING

```
      PROGRAM SHELL(INPUT,OUTPUT,TAPE1,TAPE2,TAPE3,TAPE4)
      COMMON/CV/NUMEL,NUPTS,NUBPTS,IBANDP,MBAND,NBLOC,NDFRE,IFLAG,LVECT
      DIMENSION TITLE(12),C(5)
    1 READ  100,TITLE,NUMEL,NUPTS,NUBPTS,IBANDP,IFLAG
      IF (NUMEL.LE.0)  STOP
      PRINT 200,TITLE,NUMEL,NUPTS,NUBPTS,IBANDP,IFLAG
      IF (IFLAG.LE.0)  PRINT 300
      IF (IFLAG.GE.1)  PRINT 301
      NDFRE=5
      MBAND=IBANDP*NDFRE
      NBLOC =(NUPTS*NDFRE)/MBAND
      IF((MBAND*NBLOC-NUPTS*NDFRE).NE.0)  NBLOC=NBLOC+1
      REWIND 4
      REWIND 1
      REWIND 2
      REWIND 3
      CALL SECOND(C(1))
      CALL SETUP
      CALL SECOND(C(2))
      CALL SPARSE
      CALL SECOND(C(3))
      CALL SOLVE
      CALL SECOND(C(4))
      CALL STRESS
      CALL SECOND(C(5))
      PRINT 80
      DO 5 I=1,4
      T=C(I+1)-C(I)
    5 PRINT 90,T
      GOTO 1
  100 FORMAT (12A6/,4I5,6X,I4)
  200 FORMAT (1H6////12A6///
     1 40H NUMBER OF ELEMENTS                                I4//
     2 40H NUMBER OF NODES                                   I4//
     3 40H NUMBER OF NODES WITH DISPLACEMENT B.C.            I4//
     4 40H BANDWIDTH                                         I4//
     5 40H BASE COORDINATES FOR TRANSLATIONS                 I4)
  300 FORMAT (10X,8H(GLOBAL) )
  301 FORMAT (10X,9H(SURFACE))
   80 FORMAT (///19H SOLUTION TIME LOG   )
   90 FORMAT(F13.4)
      END
```

```
      SUBROUTINE SETUP
      COMMON/CV/NUMEL,NUPTS,NUBPTS,IBANDP,MBAND,NBLOC,NDFRE,IFLAG,LVECT
      COMMON IB(401),XQ(3,400),X(4),Y(4),Z(4),IQ(4,400),KQ(4),
     1 DIR(6,401),D1(3,4),D2(3,4),U(6),V(3),S(201),T(3,3),
     2 E1(3),E2(3),EMOD(400),THIK(400),XMOD(400)
      EQUIVALENCE
     1 (U(1),XI),(U(2),YI),(U(3),ZI),(U(4),XK),(U(5),YK),(U(6),ZK),
     2 (U(1),X0),(U(2),Y0),(U(3),Z0),(U(4), W),(U(5), A),(U(6), P),
     3 (V(1),XJ),(V(2),YJ),(V(3),ZJ),(V(1),BI),(V(2),BJ)
      LOGICAL TEST1,TEST2
      DO 80 I=1,401
      DIR(6,I)= 0.
   80 IB(I)=0
      PI=3.1415926535898 JJ=0
C.....GENERATE NODAL COORDINATES AND DIRECTION COSINES.
    1 READ  90,II,IGO,U
      IF(IGO.GT.0)  READ  90,JJ,I,V
      IF(IGO.LT.0)  GOTO  17
      IGO=IGO+1
      XINC=JJ-II
      GOTO ( 2, 4, 8,10,12),IGO
    2 DO 3 I=1,3
    3 XQ(I,II)=U(I)
      GOTO  19
C.....STRAIGHT LINE.
    4 XJ=XJ-XI
      YJ=YJ-YI
      ZJ=ZJ-ZI
      XL=SQRT(XJ**2+YJ**2+ZJ**2)
      XD=XJ/XINC
      YD=YJ/XINC
      ZD=ZJ/XINC
      DO 5 I=II,JJ
      XINC=I-II
      XQ(1,I)=XI+XD*XINC
      XQ(2,I)=YI+YD*XINC
    5 XQ(3,I)=ZI+ZD*XINC
      SQ=SQRT(XJ*XJ+YJ*YJ)
      DO 7 I=II,JJ
      DIR(1,I)=XJ/XL
      DIR(2,I)=YJ/XL
      DIR(3,I)=ZJ/XL
      IF(SQ.EQ..0)  GOTO 6
      DIR(4,I)=-YJ/SQ
      DIR(5,I)= XJ/SQ
      GOTO 7
    6 DIR(4,I)=0.
      DIR(5,I)=1.
    7 CONTINUE
      GOTO  19
C.....CIRCULAR CURVE.
    8 DIJ=SQRT((XI-XJ)**2+(YI-YJ)**2+(ZI-ZJ)**2)
      DIL=SQRT((XI-XK)**2+(YI-YK)**2+(ZI-ZK)**2)/2.
      DJL=SQRT(DIJ**2-DIL**2)
      DEL=PI        -2.*ATAN(DIL/DJL)
```

```
      XL=(XI+XK)/2.
      YL=(YI+YK)/2.
      ZL=(ZI+ZK)/2.
      R=   DIJ/SIN(DEL/2.)
      T(1,1)=(XK-XL)/DIL
      T(2,1)=(YK-YL)/DIL
      T(3,1)=(ZK-ZL)/DIL
      T(1,2)=(XJ-XL)/DJL
      T(2,2)=(YJ-YL)/DJL
      T(3,2)=(ZJ-ZL)/DJL
      T(1,3)= T(2,1)*T(3,2)-T(2,2)*T(3,1)
      T(2,3)=-T(1,1)*T(3,2)+T(1,2)*T(3,1)
      T(3,3)= T(1,1)*T(2,2)-T(1,2)*T(2,1)
      CONST=SQRT(T(1,3)**2+T(2,3)**2+T(3,3)**2)
      AINC=DEL/XINC
      DO 9 I=II,JJ
      XINC=I-II
      ANG=AINC*XINC
      DX=R*SIN(ANG)*COS(DEL-ANG)
      DY=R*SIN(ANG)*SIN(DEL-ANG)
      XQ(1,I)=XI+T(1,1)*DX+T(1,2)*DY
      XQ(2,I)=YI+T(2,1)*DX+T(2,2)*DY
      XQ(3,I)=ZI+T(3,1)*DX+T(3,2)*DY
      C=COS(DEL-2.*ANG)
      D=SIN(DEL-2.*ANG)
      DIR(1,I)=T(1,1)*C+T(1,2)*D
      DIR(2,I)=T(2,1)*C+T(2,2)*D
      DIR(3,I)=T(3,1)*C+T(3,2)*D
      DIR(4,I)=-T(1,3)/CONST
      DIR(5,I)=-T(2,3)/CONST
    9 DIR(6,I)=-T(3,3)/CONST
      GOTO  19
C.....PARABOLA.
   10 IF(ABS(BI).GT.ABS(BJ))  CONST=ZJ/BI**2
      IF(ABS(BJ).GE.ABS(BI))  CONST=ZJ/BJ**2
      DIJ=ABS(BI)+ABS(BJ)
      W=W*PI/180.
      CW=COS(W)
      SW=SIN(W)
      DINC=DIJ/XINC
      DO 11 I=II,JJ
      XINC=I-II
      DX=BI+DINC*XINC
      XQ(1,I)=XO+DX*CW
      XQ(2,I)=YO+DX*SW
      XQ(3,I)=ZO+CONST*DX*DX
      ANG=ATAN(2.*CONST*DX)
      DIR(1,I)=CW*COS(ANG)
      DIR(2,I)=SW*COS(ANG)
      DIR(3,I)= SIN(ANG)
      DIR(4,I)=-SW
   11 DIR(5,I)= CW
      GOTO  19
C.....ELLIPSE.
   12 A2=A*A
```

```
      B2=B*B
      AB=A/B
      BA=B/A
      ATB=A*B
      W=W*PI/180.
      SW=SIN(W)
      CW=COS(W)
      IFAC=200
      FAC=IFAC
      XK=B2/SQRT(A2+B2)
      Z1=AB*SQRT(B2-BI*BI)
      ZJ=AB*SQRT(B2-BJ*BJ)
      WI=-PI/2.
      WJ= PI/2.
      IF (Z1.GT..0) WI=ATAN(BI/Z1)
      IF(ZJ.GT..0) WJ=ATAN(BJ/ZJ)
      DW=(WJ-WI)/FAC
      WC=PI/2.  -WI-DW
      S(1)=0.
      DO 13 I=1,IFAC
      SWC=SIN(WC)
      CWC=COS(WC)
      R=   ATB/SQRT(B2*SWC*SWC+A2*CWC*CWC)
      DX=R*CWC-BI
      DZ=R*SWC-Z1
      S(I+1)=S(I)+SQRT(DX*DX+DZ*DZ)
      BI=BI+DX
      Z1=Z1+DZ
   13 WC=WC-DW
      DS=S(IFAC+1)/XINC
      ST=0.
      WC=PI/2.  -WI
   14 DO 15 K=1,IFAC
      J=K+1
      IF(S(J).GE.ST)  GOTO 16
   15 CONTINUE
   16 AINC=(J-2)*DW
      ANG=WC-AINC-((ST-S(J-1))/(S(J)-S(J-1)))*DW
      SS=SIN(ANG)
      CC=COS(ANG)
      R=   ATB/SQRT(B2*SS*SS+A2*CC*CC)
      XR=ABS(R*CC)
      ZR=ABS(R*SS)
      Q=SIGN(1.,CC)
      IF(XR.LE.XK)  ANGT=-Q*ATAN((AB*XR)/SQRT(B2-XR*XR))
      IF(XR.GT.XK)  ANGT=-Q*(PI/2.  -ATAN(BA*ZR/SQRT(A2-ZR*ZR)))
      SA=SIGN(1.,A)
      XQ(1,II)=X0+XR*CW*Q
      XQ(2,II)=Y0+XR*SW*Q
      XQ(3,II)=Z0+ZR*SA
      DIR(1,II)=COS(ANGT)*CW*SA
      DIR(2,II)=COS(ANGT)*SW*SA
      DIR(3,II)=SIN(ANGT)*SA
      DIR(4,II)=-SW
      DIR(5,II)= CW
```

```
      IF(II.EQ.JJ)  GOTO 19
      II=II+1
      ST=ST+DS
      GOTO  14
C.....REPEATED NODAL COORDINATES,AND E1 COSINES.
   17 JJ=IABS(IGO)
      I=JJ+1-II
      DO 18 J=II,JJ
      K=J-I
      DO 18 L=1,3
      DIR(L,J)=DIR(L,K)
      DIR(L+3,J)=DIR(L+3,K)
   18 XQ(L,J)=XQ(L,K)+U(L)
   19 IF((JJ.LT.NUPTS).AND.(II.LT.NUPTS))  GOTO 1
C.....INPUT DIRECTION COSINES FOR ARBITARY NODAL POINTS.
   51 CONTINUE
   53 READ  95, M,IGO,E1,E2,MOD,LIM
      IF( M.LE.0)  GOTO 57
      TEST1=ABS(E1(1))+ABS(E1(2))+ABS(E1(3)).GT..0
      TEST2=ABS(E2(1))+ABS(E2(2))+ABS(E2(3)).GT..0
      IF(MOD.LE.0)  LIM=M
      IF(MOD.LE.0)  MOD=1
      DO 52 L=M,LIM,MOD
      DO 52 K=1,3
      IF(TEST1)  DIR(K  , L)=E1(K)
   52 IF(TEST2)  DIR(K+3, L)=E2(K)
      GOTO 53
C.....INPUT MESH.
   57 CONTINUE
   59 READ  93,JJ, (IQ(I,JJ),I=1,4),MODL,NLAY
   62 II=JJ
      IF(MODL.GT.0)  GOTO 64
      IF(II.EQ.NUMEL)  GOTO 66
      READ  93,JJ,(IQ(I,JJ),I=1,4),MODL,NLAY
      IF(II+1.EQ.JJ)  GOTO 62
      JJ=JJ-2
      DO 63 J=II,JJ
      DO 63 K=1,4
   63 IQ(K,J+1)=IQ(K,J)+1
      JJ=JJ+2
      GOTO 62
   64 DO 65 I=1,NLAY
      LL=2
      DO 65 J=1,MODL
      IF (II.EQ.JJ)  GO TO 641
      DO 640 N=1,4
  640 IQ(N,II)=IQ(N,II-1)+LL
  641 LL=1
   65 II=II+1
      IF(II-1.LT.NUMEL)  GOTO 59
C.....INPUT MATERIAL PROPERTIES.
   66 CONTINUE
      READ  90,II,I,EMOD(II),THIK(II),XMOD(II)
      IF(II.EQ.NUMEL)  GOTO 73
      READ  90,JJ,I,EMOD(JJ),THIK(JJ),XMOD(JJ)
```

```
      IF(II+1.EQ.JJ)  GOTO 66
      XINC=JJ-II
      XINC=(THIK(JJ)-THIK(II))/XINC
      JJ=JJ-2
      DO 72 I=II,JJ
      EMOD(I+1)=EMOD(I)
      XMOD(I+1)=XMOD(I)
   72 THIK(I+1)=THIK(I)+XINC
      IF(JJ+2.LT.NUMEL)  GOTO 66
   73 PRINT 904
      DO 74 I=1,NUPTS
   74 PRINT 92,I,(XQ(J,I),J=1,3),(DIR(J,I),J=1,6)
      PRINT 905
      DO 75 I=1,NUMEL
      MAX=MAX0(IQ(1,I),IQ(2,I),IQ(3,I),IQ(4,I))
      MIN=MIN0(IQ(1,I),IQ(2,I),IQ(3,I),IQ(4,I))
      ND=MAX-MIN
   75 PRINT 94,I,(IQ(J,I),J=1,4),EMOD(I),THIK(I),XMOD(I),ND
      DO 83 M=1,NUMEL
      NODES=4
      IF(IQ(4,M).LE.0)  NODES=3
      KQ(4) = 0.
      DO 82 J=1,NODES
      K=IQ(J,M)
      KQ(J)=K
      DO 81 L=1,3
      D1(L,J)=DIR(L,K)
   81 D2(L,J)=DIR(L+3,K)
      X(J)=XQ(1,K)
      Y(J)=XQ(2,K)
      Z(J)=XQ(3,K)
   82 IB(K)=IB(K)+1
   83 WRITE(3) KQ,X,Y,Z,EMOD(M),THIK(M),XMOD(M),NODES,D1,D2
      REWIND 3
      RETURN
   90 FORMAT(2I4,2X,6F10.6)
   92 FORMAT( I5,2X,3F17.6,2X,6F10.6)
   93 FORMAT(7I4)
   94 FORMAT (5I5,3E17.6,11X,I3)
   95 FORMAT(2I4,2X,6F10.6,2X,2I4)
  904 FORMAT (85H1NODAL COORDINATES AND DIRECTION COSINES FOR SURFACE CO
     .ORDINATES E1 AND E2 BAR                                //
     1       5H NODE12X,1HX16X,1HY16X,1HZ11X,3HE1X7X,3HE1Y7X,3HE1Z7X,3HE2X
     27X,3HE2Y7X,3HE2Z)
  905 FORMAT(53H1ELEMENT NODAL POINT NUMBERS AND MATERIAL PROPERTIES  //
     1 9H ELEMENT,1HI4X,1HJ4X,1HK4X,1HL7X,7HMODULUS10X,9HTHICKNESS8X,
     2 9HPOISSON R 6X,15HNODE DIFFERENCE )
      END
```

```
      SUBROUTINE SPARSE
      COMMON IB(401),IQ(4),IBD(100,6),IBC(400),Q(6,25),EM,NU,THIK,
     1 PX1(107),X(4),Y(4),Z(4),TG(3,3,4),PX2(2020),D(25,6,25),BC(100,5),
     2 PX3(302),UPL,C(3,4),F(3,4),LOADS(3)
      DIMENSION P(500,5,3),R(5),DX(5,5),JX1(550),CQ1(4500),CQ2(8310)
      DIMENSION PR(400)
      COMMON/CV/NUMEL,NUPTS,NUBPTS,IBANDP,MBAND,NBLOC,NDFRE,IFLAG,LVECT
      EQUIVALENCE (IB,P),(IB,CQ2),(IBC,JX1),(D,CQ1)
      INTEGER Q
      REAL NU
C     D(NDFRE**2,6,25),IBD(NUBPTS,NDFRE+1,)BC(NUBPTS,NDFRE),IBC(NUPTS)
C     IB(NUPTS),P(NUPTS,NDFRE),R(NDFRE),IQ,X,Y,Z(NODES).
      DATA DX/1.,4*0.,0.,1.,3*0.,2*0.,1.,2*0.,3*0.,1.,0.,4*0.,1./
C.....INITIALIZE IBC,Q,D,BC,(P AND PM FOR SUB1 AND BLAY).
      DO 5 I=1,550
    5 JX1(I)=0
      DO 6 I=1,4500
    6 CQ1(I)=0.
      IK=1
      L=0
    7 L=L+1
      READ  80,(IBD(L,K),K=1,6),(BC(L,K),K=1,5),MOD,LIM
      IF(L.EQ.NUBPTS)  GOTO 74
      IF(MOD) 7,7,8
    8 K=IBD(L,1)+MOD
      DO 70 I=K,LIM,MOD
      L=L+1
      IBD(L,1)=I
      DO 70 J=1,5
      IBD(L,J+1)=IBD(L-1,J+1)
   70 BC(L,J)=BC(L-1,J)
      IF(L.LT.NUBPTS)  GOTO 7
   74 PRINT 90
      PRINT 91
      DO 79 L=1,NUBPTS
   79 PRINT 81,(IBD(L,K),K=1,6),(BC(L,K),K=1,5)
      READ 49,LVECT,LOADS,UPL,NPR
      PRINT 50,LVECT
      PRINT 51,(LOADS(I),I,I=1,LVECT)
      PRINT 52,NPR,UPL
C............NORMAL PRESSURE
      DO 100 I=1,NUMEL
  100 PR(I)=UPL
      IF (NPR.LE.0)  GO TO 200
      PRINT 53
      MM=0
  105 READ 55,M,PL,MOD,LIM
      IF (MOD.LE.0)  LIM=M
      DO 120 L=M,LIM,MOD
      MM=MM+1
      PR(L)=PL
  120 PRINT 54,L,PL
      IF (MM.LT.NPR)  GO TO 105
   49 FORMAT (4I5,F10.0,I5)
   50 FORMAT (21H1NUMBER OF LOAD CASES  I5//)
```

```
   51 FORMAT (I5,31H LOADED NODES FOR LOAD CASE NO.  I5)
   52 FORMAT (//I5,44H ELEMENTS WITH PRESSURE DIFFERENT FROM UPL =F15.5,
   53 FORMAT (//8H ELEMENT 6X,8HPRESSURE)
   54 FORMAT (I5,F15.5)
   55 FORMAT (I5,F10.0,47X,2I4)
  200 DO 9 I=1,NUBPTS
      J=IBD(I,1)
    9 IBC(J)=I
      DO 10 LZ=1,NUMEL
      READ (3) IQ,X,Y,Z,EM,THIK,NU,NODES,C,E
      DO 20 I=1,NODES
      TG(1,1,I)=C(1,I)
      TG(1,2,I)=C(2,I)
      TG(1,3,I)=C(3,I)
      TG(3,1,I)= C(2,I)*E(3,I)-C(3,I)*E(2,I)
      TG(3,2,I)=-C(1,I)*E(3,I)+C(3,I)*E(1,I)
      TG(3,3,I)= C(1,I)*E(2,I)-C(2,I)*E(1,I)
      CONST=SQRT(TG(3,1,I)**2+TG(3,2,I)**2+TG(3,3,I)**2)
      TG(3,1,I)=TG(3,1,I)/CONST
      TG(3,2,I)=TG(3,2,I)/CONST
      TG(3,3,I)=TG(3,3,I)/CONST
      TG(2,1,I)= TG(1,3,I)*TG(3,2,I)-TG(1,2,I)*TG(3,3,I)
      TG(2,2,I)=-TG(1,3,I)*TG(3,1,I)+TG(1,1,I)*TG(3,3,I)
   20 TG(2,3,I)= TG(1,2,I)*TG(3,1,I)-TG(1,1,I)*TG(3,2,I)
      UPL=PR(LZ)
      CALL QDSHEL
   10 CALL SUBSPR(IK,NODES)
      REWIND 4
      JK=NBLOC*IBANDP
      IF(IK.GT.JK)  GOTO 31
      K=1
      DO 30 I=IK,JK
   30 WRITE(2) K,I,((DX(J,L),L=1,NDFRE),J=1,NDFRE)
   31 REWIND 2
C.....LIMIT BELOW MUST EQUAL SIZE OF SIZE OF BLANK COMMON..............C
      DO 32 I=1,8310
   32 CQ2(I)=0.
      DO 11 L=1,NUPTS
   11 READ(4)  (P(L,J,3),J=1,NDFRE),(P(L,J,1),J=1,NDFRE)
      DO 40 I=1,NUPTS
      DO 40 J=1,NDFRE
      DO 40 K=1,2
   40 P(I,J,K)=P(I,J,3)+P(I,J,K)
      DO 73 I=1,LVECT
      II=1
      IF(LOADS(I).LE.0)  GOTO 73
   71 READ 1000,M,R,MOD,LIM
      IF(MOD.LE.0)  LIM=M
      IF(MOD.LE.0)  MOD=1
      DO 72 L=M,LIM,MOD
      II=II+1
      DO 72 K=1,5
   72 P(L,K,I)=P(L,K,I)+R(K)
      IF(II.LE.LOADS(I))  GOTO 71
   73 CONTINUE
```

```
      DO 75 I=1,LVECT
      PRINT 92,I
      PRINT 86
   75 PRINT 88,(L,(P(L,J,I),J=1,5),L=1,NUPTS)
   13 REWIND 4
      DO 14 N=1,NBLOC
      K=(N-1)*IBANDP+1
      L=K+IBANDP-1
   14 WRITE(4)(((P(I,J,IJ),J=1,NDFRE),I=K,L),IJ=1,LVECT)
      REWIND 4
      RETURN
   80 FORMAT(I4,5I1,1X,5F10.0,2X,2I4)
   81 FORMAT(I11,4X,5I3,5X,5F15.6,4X,2I5)
   85 FORMAT(I11,5X,5F15.6,2X,2I4)
   86 FORMAT(1H0 6X,4HNODE 9X,2HP1 13X,2HP2 13X,2HP3 13X,2HP4 13X,2HP5)
 1000 FORMAT(I3,7X,5F10.0,2X,2I4)
   88 FORMAT(I11,5X,5E15.6)
   90 FORMAT (55H1BOUNDARY CONTITIONS OF POINTS HAVING SPECIFIED DISPLS.
     . /7X,90HD1,D2,D3, ARE TRANSLATIONS IN BASE COORDINATES. D4,D5 ARE
     .ROTATIONS IN SURFACE COORDINATES./)
   91 FORMAT(1H0 6X,4HNODE 5X,14HD1 D2 D3 D4 D5 9X,2HD1 13X,2HD2 13X
     1,2HD3 13X,2HD4 13X,2HD5)
   92 FORMAT (34H1TOTAL APPLIED NODAL POINT FORCES  9HLOAD CASE  I4)
      END
```

```
      SUBROUTINE SUBSPR(IK,NODES)
      COMMON IB(401),IQ(4),IX1(1000),Q(6,25),PX1(167),S(37,37),PX2(642),
     1 D(25,6,25),PX3(625),PM(25,5),PX4(15),PT(37),PX5(25)
      COMMON/CV/NUMEL,NUPTS,NUBPTS,IBANDP,MBAND,NBLOC,NDFRE,IFLAG,LVECT
      INTEGER Q
      DO 4 II=1,NODES
      I=IQ(II)
      L=(II-1)*5
      K=I-IK+1
      PM(K,1)=PT(L+1)+PM(K,1)
      PM(K,2)=PT(L+2)+PM(K,2)
      PM(K,3)=PT(L+3)+PM(K,3)
      IB(I)=IB(I)-1
      DO 4 JJ=1,NODES
      J=IQ(JJ)-IK+1
      IF(IQ(JJ).GT.I)  GOTO 4
      DO 1 LL=1,7
      IF(Q(LL,J).EQ.0)  Q(LL,J)=I
      IF(Q(LL,J).EQ.I)  GOTO 2
    1 CONTINUE
    2 IS=(II-1)*NDFRE
      JS=(JJ-1)*NDFRE
      DO 3 IC=1,NDFRE
      N=IC+JS
      MN=(IC-1)*NDFRE
      DO 3 IR=1,NDFRE
      M=IR+IS
      MN=MN+1
    3 D(MN,LL,J)=D(MN,LL,J) + S(M,N)
    4 CONTINUE
   10 IF(IB(IK).GT.0)  GOTO 6
      CALL BLAYER(IK)
      DO 5 I=1,24
      DO 5 J=1,6
      Q(J,I)=Q(J,I+1)
      DO 5 K=1,25
    5 D(K,J,I)=D(K,J,I+1)
      IK=IK+1
      IF(IK.LE.NUPTS.AND.IB(IK).EQ.0)  GOTO 10
    6 RETURN
      END
```

```
      SUBROUTINE BLAYER(J)
      COMMON IX1(405),IBD(100,6),IBC(400),Q(6,25),PX1(2178),D(25,6,25),
     1 BC(100,5),P(25,5),PM(25,5),PX2(15),PT(37),PX3(25)
      COMMON/CV/NUMEL,NUPTS,NUBPTS,IBANDP,MBAND,NBLOC,NDFRE,IFLAG,LVECT
      INTEGER Q,QQ(6),X,Y,Z
      DIMENSION DD(5,5,6)
      EQUIVALENCE (Q,QQ),(D,DD)
    1 DO 10 II=1,6
      I=QQ(II)
      IF(I.EQ.0)  GOTO 10
      IT=II
      X=IBC(I)
      Y=IBC(J)
      Z=I-J+1
C.......................................................................
      IF(I.EQ.J.OR.X.EQ.0)  GOTO 4
C.....MODIFY LOAD VECTOR FOR B.C. ON UPPER BLOCKS........................
      DO 3 L=1,NDFRE
      IF(IBD(X,L+1).EQ.0)  GOTO 3
      DO 2 M=1,NDFRE
    2 P(1,M)=P(1,M) - DD(L,M,II)*BC(X,L)
    3 CONTINUE
C.......................................................................
    4 IF(Y.EQ.0)  GOTO 7
C.....MODIFY LOAD VECTOR FOR B.C. ON LOWER BLOCKS........................
      DO 6 L=1,NDFRE
      IF(IBD(Y,L+1).EQ.0)  GOTO 6
      DO 5 M=1,NDFRE
      P(Z,M)=P(Z,M)-DD(M,L,II)*BC(Y,L)
    5 DD(M,L,II)=0.
    6 CONTINUE
C.......................................................................
    7 IF(X.EQ.0)  GOTO 10
      DO 9 L=1,NDFRE
      IF(IBD(X,L+1).EQ.0)  GOTO 9
      DO 8 M=1,NDFRE
      DD(L,M,II)=0.
    8 IF(I.EQ.J.AND.L.EQ.M)  DD(M,L,II)=1.
    9 CONTINUE
C.......................................................................
   10 CONTINUE
      WRITE(2)
     1 IT,(QQ(I),I=1,IT),(((DD(I,L,K),I=1,NDFRE),L=1,NDFRE),K=1,IT)
      IF(Y.EQ.0)  GOTO 14
      DO 11 K=1,NDFRE
      IF(IBD(Y,K+1).EQ.1)  PM(1,K)=0.
   11 IF(IBD(Y,K+1).EQ.1)  P(1,K) = BC(Y,K)
   14 WRITE(4)((P(I,K),K=1,NDFRE),I=1,1),(PM(1,K),K=1,5)
      DO 12 I=1,24
      DO 12 K=1,NDFRE
      PM(I,K)=PM(I+1,K)
   12 P(I,K)=P(I+1,K)
      RETURN
      END
```

```
      SUBROUTINE NLOAD(NT)
      COMMON PX1(1555),PX1(1545),AD(3,4),BD(3,4),PX2(5109),PF(3,5),
     1PX3(37),UPL,PX4(24)
      A1=AD(1,NT)
      A2=AD(2,NT)
      A3=AD(3,NT)
      B1=BD(1,NT)
      B2=BD(2,NT)
      A=A3*B2/2.
      Y1=-A1*A3/(A1**2+B1**2)
      Y2=-(A2*A1+B2*B1)/(A2**2+B2**2)
      Y3=-A2*A3/A3**2
      PZ=A*UPL
      DO 1 I=1,3
      DO 1 J=1,3
    1 PF(I,J)=0.
      PF(3,1)=(10.-Y3+Y2)*PZ/30.
      PF(3,2)=(10.-Y1+Y3)*PZ/30.
      PE(3,3)=(10.-Y2+Y1)*PZ/30.
      RETURN
      END
```

```
      SUBROUTINE QDSHEL
      COMMON/CV/NUMEL,NUPTS,NUBPTS,IBANDP,MBAND,NBLOC,NDFRE,IFLAG,LVECT
      COMMON IX1(401),IQ(4),IX2(1150),EM,NU,THIK,AREA,B(3),A(3),
     1 ST(10,10),X(4),Y(4),Z(4),TG(3,3,4),TO(3,3),S(37,37),X1,Y1,Z1,
     2 X2,Y2,Z2,X3,Y3,Z3,AD(3,4),BD(3,4),T(3,3,4),TDIS(3,3,3),
     3 TROT(3,3,3),TD(3,36),TR(3,36),PX1(4803),PF(3,5),PT(37),PX2(25)
C.......................................................................
      DIMENSION SS(37,13),TX(3,3),IPERM(4),LOCB(15,4),LOCQ(3,5,4),
     1 LOCM(5)
      EQUIVALENCE (S(741),SS),(X1,TX)
      DATA  LOCQ /
     1   1, 2, 3, 6, 7, 8,21,22,23,28,29,35,26,27,34,
     2   6, 7, 8,11,12,13,21,22,23,30,31,36,28,29,35,
     3  11,12,13,16,17,18,21,22,23,32,33,37,30,31,36,
     4  16,17,18, 1, 2, 3,21,22,23,26,27,34,32,33,37  /
      DATA  LOCB / 1, 2, 3, 4, 5, 6, 7, 8, 9,10,21,22,23,24,25,
     1             6, 7, 8, 9,10,11,12,13,14,15,21,22,23,24,25,
     2            11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,
     3            16,17,18,19,20, 1, 2, 3, 4, 5,21,22,23,24,25 /
      DATA  LOCM/1,3,5,7,9/, IPERM /2,3,4,1/
      REAL NU,NORM
C........................................................................
C.....TRANSFORMATION MATRICES............................................
C.....T(3,3,4).......FROM Z TO ZBAR......................................
C.....TO(3,3)........FROM Z TO ZO........................................
C.....TG(3,3,4)......FROM Z TO Z'........................................
C.....TROT(3,3,3)....FROM Z' TO ZBAR.....................................
C.....TDIS(3,3,3)....FROM ZB TO ZBAR.....................................
C.....T(3,3,4).......FROM ZB TO ZO.......................................
C.....COMPUTE INTERNAL MID POINT COORDINATES IF QUAD.....................
      NTRI=4
      IF(IQ(4).LT.1)  NTRI=1
      IF(NTRI.EQ.1)  IADD=10
      IF(NTRI.EQ.4)  IADD=20
      DO 10 I=1,3
   10 LOCQ(I,3,1)=IADD+I
      DO 11 I=1,5
   11 LOCB(I+10,1)=IADD+I
      IF(NTRI.EQ.1)  NDIM=15
      IF(NTRI.EQ.4)  NDIM=37
      DO 160  I = 1,NDIM
      PT(I)=0.
      DO 160  J = 1,NDIM
  160 S(I,J) = 0.
      IF(THIK.LT..000001)  GOTO 1000
      XC = 0.25*(X(1)+X(2)+X(3)+X(4))
      YC = 0.25*(Y(1)+Y(2)+Y(3)+Y(4))
      ZC = 0.25*(Z(1)+Z(2)+Z(3)+Z(4))
      IF(NTRI.EQ.4)  GOTO 700
      XC=X(3)
      YC=Y(3)
      ZC=Z(3)
C.....COMPUTE ELEMENT DIRECTION COSINES, T(I,J,4)........................
  700 DO 130  N = 1,NTRI
      M = IPERM(N)
```

```
      X1  =  X(M)-X(N)
      Y1  =  Y(M)-Y(N)
      Z1  =  Z(M)-Z(N)
      X2  =  XC  -  X(N)
      Y2  =  YC  -  Y(N)
      Z2  =  ZC  -  Z(N)
      S11  =  X1*X1+Y1*Y1+Z1*Z1
      S12  =  X1*X2+Y1*Y2+Z1*Z2
      S22  =  X2*X2+Y2*Y2+Z2*Z2
      COS12  =  -S12/S11
      X2  =  X2  +  X1*COS12
      Y2  =  Y2  +  Y1*COS12
      Z2  =  Z2  +  Z1*COS12
      S1  =  SQRT(S11)
      S2  =  SQRT(X2*X2+Y2*Y2+Z2*Z2)
      T(1,1,N)  =  X1/S1
      T(1,2,N)  =  Y1/S1
      T(1,3,N)  =  Z1/S1
      T(2,1,N)  =  X2/S2
      T(2,2,N)  =  Y2/S2
      T(2,3,N)  =  Z2/S2
      T(3,1,N)  =  T(1,2,N)*T(2,3,N)  -  T(1,3,N)*T(2,2,N)
      T(3,2,N)  =  T(1,3,N)*T(2,1,N)  -  T(1,1,N)*T(2,3,N)
      T(3,3,N)  =  T(1,1,N)*T(2,2,N)  -  T(1,2,N)*T(2,1,N)
C.....COMPUTE A'S AND B'S..................................................
      AD(2,N)  =  S1*COS12
      AD(3,N)  =  S1
      AD(1,N)  =  -AD(3,N)-AD(2,N)
      BD(1,N)  =  -(S22+COS12*S12)/S2
      BD(2,N)  =  -BD(1,N)
  130 BD(3,N)  =  0.
C.....DIRECTION COSINES FOR MID POINT IF ELEMENT IS A TRI...............
      DO 900  I=1,3
      DO 900  J=1,3
      TO(I,J)=TG(I,J,3)
  900 TDIS(I,J,3)=  T(I,J,1)
      IF(NTRI.EQ.1)   GOTO 701
C.....COMPUTE DIRECTION COSINES OF N1,N2 PLANE.............................
      CALL GETNRM(4,X,Y,Z,A3,B3,C3,IERROR,XC,YC,ZC)
      TO(3,1)=A3
      TO(3,2)=B3
      TO(3,3)=C3
      XI  =      T(1,1,1)
      ETA  =     T(1,2,1)
      DSETA  =  T(1,3,1)
      A3X  =  A3*XI
      B3E  =  B3*ETA
      C3D  =  C3*DSETA
      X1  =  (B3**2+C3**2)*XI       -  A3*(C3D+B3E)
      Y1  =  (C3**2+A3**2)*ETA      -  B3*(A3X+C3D)
      Z1  =  (A3**2+B3**2)*DSETA  -  C3*(B3E+A3X)
      NORM  =  SQRT(X1**2+Y1**2+Z1**2)
      TO(1,1)  =  X1/NORM
      TO(1,2)  =  Y1/NORM
      TO(1,3)  =  Z1/NORM
```

```
      TO(2,1) = TO(1,3)*B3 - TO(1,2)*C3
      TO(2,2) = TO(1,1)*C3 - TO(1,3)*A3
      TO(2,3) = TO(1,2)*A3 - TO(1,1)*B3
C.....SUM OVER 4 TRIS. IF QUAD OR 1 TRI. IF ELEMENT IS A TRI...........
  701 DO 301  NT = 1,NTRI
      N1 = NT
      N2 = IPERM(N1)
C.....COMPUTE TRANSFORMATIONS FOR EACH POINT OF TRIANGLE...............
      DO 200  I = 1,3
      A(I) = AD(I,NT)
      B(I) = BD(I,NT)
      T1 = T(I,1,NT)
      T2 = T(I,2,NT)
      T3 = T(I,3,NT)
      DO 200  J = 1,3
      TROT(I,J,1) = T1*TG(J,1,N1) + T2*TG(J,2,N1) + T3*TG(J,3,N1)
      TROT(I,J,2) = T1*TG(J,1,N2) + T2*TG(J,2,N2) + T3*TG(J,3,N2)
      TROT(I,J,3) = T1*TO(J,1)    + T2*TO(J,2)    + T3*TO(J,3)
      IF(NTRI.EQ.4.OR.IFLAG.EQ.1)  TDIS(I,J,3)=TROT(I,J,3)
      IF (IFLAG.EQ.0)  GO TO 180
      TDIS(I,J,1) = TROT(I,J,1)
      TDIS(I,J,2) = TROT(I,J,2)
      GO TO 200
  180 TDIS(I,J,1) = T(I,J,NT)
      TDIS(I,J,2) = T(I,J,NT)
  200 CONTINUE
C.....STORE BASE TRANSFORMATION MATRICES...............................
      KK=(NT-1)*9
      DO 1 K=1,3
      L=KK+(K-1)*3
      DO 1 J=1,3
      JL=J+L
      DO 1 I=1,3
      TD(I,JL)=TDIS(I,J,K)
    1 TR(I,JL)=TROT(I,J,K)
C.....ADJUST TDIS FOR TRANSFORMING TO N COORDINATES....................
      IF(NTRI.EQ.1)  GOTO 3
      DO 2 I=1,3
      DO 2 J=1,3
      TDIS(I,J,1)=TDIS(I,J,3)
    2 TDIS(I,J,2)=TDIS(I,J,3)
C.....COMPUTE AREA OF TRIANGLE.........................................
    3 AREA = A(3)*B(2) - A(2)*B(3)
C.....FORM AND TRANSFORM MEMBRANE STIFFNESS TO BASE SYSTEM IF A TRI.....
C.....FORM AND TRANSFORM MEMBRANE STIFFNESS TO ZO   SYSTEM IF A QUAD....
      IF(NTRI.EQ.1)  CALL SCST
      IF(NTRI.EQ.4)  CALL SLST10
      IEND=3
      IF(NTRI.EQ.4)  IEND=5
      DO 27 II=1,IEND
      I=LOCM(II)
      IF(II.LT.4)  IL=II
      DO 27 JJ=1,II
      J=LOCM(JJ)
      IF(JJ.LT.4)  JL=JJ
```

```
      LS=1
      DO 27 K=1,3
      N=LOCQ(K,JJ,NT)
      H1=ST(I   ,J)*TDIS(1,K,JL)+ST(I   ,J+1)*TDIS(2,K,JL)
      H2=ST(I+1,J)*TDIS(1,K,JL)+ST(I+1,J+1)*TDIS(2,K,JL)
      IF(I.EQ.J)  LS=K
      DO 27 L=LS,3
      M=LOCQ(L,II,NT)
      S(M,N)=S(M,N)+TDIS(1,L,IL)*H1+TDIS(2,L,IL)*H2
   27 S(N,M)=S(M,N)
C.....FORM AND TRANSFORM PLATE STIFFNESS TO BASE SYSTEM IF A TRI........
C.....FORM AND TRANSFORM PLATE STIFFNESS TO ZO   SYSTEM IF A QUAD.......
      CALL SLCT9
      DO 300  II = 1,3
      K = 3*II - 2
      KK = 5*(II-1)
      DO 300  JJ = II,3
      L = 3*JJ - 2
      LL = 5*(JJ-1)
      DO 300  M = 1,5
      J = LL + M
      JS = LOCB(J,NT)
      IF (M.GT.3)  GO TO 270
      T3 = TDIS(3,M,JJ)
      H1 = ST(K,  L)*T3
      H2 = ST(K+1,L)*T3
      H3 = ST(K+2,L)*T3
      GO TO 280
  270 T1 = TROT(1,M-3,JJ)
      T2 = TROT(2,M-3,JJ)
      H1 = ST(K,  L+1)*T1 + ST(K,  L+2)*T2
      H2 = ST(K+1,L+1)*T1 + ST(K+1,L+2)*T2
      H3 = ST(K+2,L+1)*T1 + ST(K+2,L+2)*T2
  280 DO 300  N = 1,5
      I = KK + N
      IF (I.GT.J)  GO TO 300
      IS = LOCB(I,NT)
      IF (N.GT.3)  GO TO 290
      S(IS,JS) = S(IS,JS) + H1*TDIS(3,N,II)
      GO TO 295
  290 S(IS,JS) = S(IS,JS) + H2*TROT(1,N-3,II) + H3*TROT(2,N-3,II)
  295 S(JS,IS) = S(IS,JS)
  300 CONTINUE
      CALL NLOAD(NT)
      DO 800 I=1,5
      K=I
      IF(NTRI.EQ.4) K=3
      Q1=PF(1,I)
      Q2=PF(2,I)
      Q3=PF(3,I)
      DO 800 J=1,3
      L=LOCQ(J,I,NT)
  800 PT(L)=PT(L)+TDIS(1,J,K)*Q1+TDIS(2,J,K)*Q2+TDIS(3,J,K)*Q3
  301 CONTINUE
      IF(NTRI.EQ.1)  GOTO 1000
```

```
C.....ELIMINATE TRANSL. COMPONENTS NORMAL TO N1,N2 PLANE OF MIDSIDE NODES
      DO 34 I=1,37
      S(I, 3)=S(I, 3)+S(I,34)/2.
      S(I, 8)=S(I, 8)+S(I,35)/2.
      S(I,13)=S(I,13)+S(I,36)/2.
      S(I,18)=S(I,18)+S(I,37)/2.
   34 S(I,23)=S(I,23)+(S(I,34)+S(I,35)+S(I,36)+S(I,37))/2.
      DO 35 J=1,37
      S( 3,J)=S( 3,J)+S(34,J)/2.
      S( 8,J)=S( 8,J)+S(35,J)/2.
      S(13,J)=S(13,J)+S(36,J)/2.
      S(18,J)=S(18,J)+S(37,J)/2.
   35 S(23,J)=S(23,J)+(S(34,J)+S(35,J)+S(36,J)+S(37,J))/2.
      PT( 3)=PT( 3)+PT(34)/2.
      PT( 8)=PT( 8)+PT(35)/2.
      PT(13)=PT(13)+PT(36)/2.
      PT(18)=PT(18)+PT(37)/2.
      PT(23)=PT(23)+(PT(34)+PT(35)+PT(36)+PT(37))/2.
C.....CONDENSE INTERNAL DEGREES OF FREEDOM..............................
      DO 400  N = 1,13
      K = 33 - N
      L = K + 1
      PIVOT = S(L,L)
      DO 400  I = 1,K
      C = S(I,L)/PIVOT
      PT(I)=PT(I)-C*PT(L)
      S(I,L) = C
      DO 400  J = I,K
      S(I,J) = S(I,J) - C*S(L,J)
  400 S(J,I) = S(I,J)
C.....ESTABLISH TRANSFORMATION FROM BASE COORDS. TO N1,N2 PLANE.........
      IF(IFLAG.EQ.1)  GOTO 39
      DO 38 L=1,4
      DO 38 I=1,3
      DO 38 J=1,3
   38 T(I,J,L)=T0(I,J)
      GOTO 41
   39 DO 40 L=1,4
      DO 40 I=1,3
      DO 40 J=1,3
      T(I,J,L)=0.
      DO 40 K=1,3
   40 T(I,J,L)=T(I,J,L)+T0(I,K)*TG(J,K,L)
C.....TRANSFORM THE 4 EXTERIOR NODES TO THE BASE SYSTEM................
   41 K=0
      DO 43 II=1,16,5
      K=K+1
      DO 42 I=1,3
      DO 42 J=1,3
   42 TX(I,J)=T(I,J,K)
      Q1=PT(II   )
      Q2=PT(II+1)
      Q3=PT(II+2)
      PT(II   )=X1*Q1+Y1*Q2+Z1*Q3
      PT(II+1)=X2*Q1+Y2*Q2+Z2*Q3
```

```
      PT(II+2)=X3*Q1+Y3*Q2+Z3*Q3
      DO 43 I=II,20
      F=S(I,II   )
      G=S(I,II+1)
      H=S(I,II+2)
      S(I,II   )=F*X1+G*Y1+H*Z1
      S(I,II+1)=F*X2+G*Y2+H*Z2
   43 S(I,II+2)=F*X3+G*Y3+H*Z3
      K=0
      DO 45 II=3,18,5
      K=K+1
      DO 44 I=1,3
      DO 44 J=1,3
   44 TX(J,I)=T(I,J,K)
      DO 45 J=1,II
      F=S(II-2,J)
      G=S(II-1,J)
      H=S(II   ,J)
      S(II-2,J)=X1*F+X2*G+X3*H
      S(II-1,J)=Y1*F+Y2*G+Y3*H
   45 S(II   ,J)=Z1*F+Z2*G+Z3*H
      DO 46 J=1,20
      DO 46 I=J,20
   46 S(J,I)=S(I,J)
 1000 WRITE (1) IQ,EM,NU,THIK,AD,BD,TD,TR,T
      IF(IQ(4).GE.1)  WRITE (1)((SS(I,J),I=1,33),J=1,13),(PT(I),I=21,33)
 1001 RETURN
      END
```

```
      SUBROUTINE SCST
C
C     STIFFNESS SUBROUTINE FOR CONSTANT STRAIN TRIANGLE (CST).
C     LINEAR ELASTIC ISOTROPIC MATERIAL
C
      COMMON/CV/NUMEL,NUPTS,NUBPTS,IBANDP,MBAND,NBLOC,NDFRE,NODES,LVECT
      COMMON IX1(1555),EM,NU,THIK,AREA,B(3),A(3),ST(10,10),PX1(6645)
      REAL NU
      C1 = EM*THIK/(2.*AREA*(1.-NU**2))
      C2 = C1*NU
      C3 = 0.5*C1*(1.-NU)
      DO 200  I = 1,3
      K = I + I
      DO 200  J = I,3
      L = J + J
      AA = A(I)*A(J)
      BB = B(I)*B(J)
      AB = A(I)*B(J)
      BA = B(I)*A(J)
      ST(K-1,L-1) = C1*BB + C3*AA
      ST(K  ,L  ) = C1*AA + C3*BB
      ST(K-1,L  ) = C2*BA + C3*AB
  200 ST(K  ,L-1) = C2*AB + C3*BA
      DO 300  I = 2,6
      K = I - 1
      DO 300  J = 1,K
  300 ST(I,J) = ST(J,I)
      RETURN
      END
```

```
      SUBROUTINE SLST10
C
C     ELEMENT STIFFNESS SUBROUTINE
C     LINEAR STRAIN TRIANGLE WITH 5 NODAL POINTS (10 DEGREES OF FREEDOM)
C     LINEAR ELASTIC ISOTROPIC MATERIAL
C     EXTERNALLY CONSTRAINED SIDE IS 1-2
C
      COMMON IX1(1555),EM,NU,THIK,AREA,B(3),A(3),ST(10,10),PX1(1765),
     1 CX1(3),CX2(3),CX3(3),CY1(3),CY2(3),CY3(3),UV(3,5,2),PX2(4832)
      DIMENSION U(3,5),V(3,5),IPERM(3),BA(3,2)
      EQUIVALENCE (BA,B), (UV,U), (UV(16),V)
      REAL NU, NUH
      NUH = 0.5*(1.-NU)
      FAC = EM*THIK/(24.*AREA*(1.-NU**2))
      DO  150  N = 1,2
      DO 150  I = 1,2
      DO 140  J = 1,2
  140 UV(I,J,N) =  BA(J,N)
      UV(I,I,N) = UV(I,I,N) - 2.*BA(3,N)
      UV(I,3,N) =  -BA(3,N)
      UV(3,I,N) =  -BA(I,N)
      K = 6 - I
      UV(I,K,N) = 4.*BA(3,N)
      UV(3,K,N) = 4.*BA(I,N)
      UV(I,I+3,N) = 0.
  150 UV(3,3,N) = 3.*BA(3,N)
      DO 300  I = 1,5
      SUMU = U(1,I)+U(2,I)+U(3,I)
      SUMV = V(1,I)+V(2,I)+V(3,I)
      DO 200  L = 1,3
      X = (SUMU+U(L,I))*FAC
      Y = (SUMV+V(L,I))*FAC
      CX1(L) = X
      CY2(L) = Y
      CX2(L) = X*NU
      CY1(L) = Y*NU
      CX3(L) = Y*NUH
  200 CY3(L) = X*NUH
      K2 = 2*I
      K1 = K2 - 1
      DO 300  J = I,5
      L2 = 2*J
      L1 = L2 - 1
      X1 = 0.
      X2 = 0.
      X3 = 0.
      X4 = 0.
      DO 280  K = 1,3
      X = U(K,J)
      Y = V(K,J)
      X1 = X1 + CX1(K)*X + CX3(K)*Y
      X2 = X2 + CX2(K)*Y + CX3(K)*X
      X3 = X3 + CY1(K)*X + CY3(K)*Y
  280 X4 = X4 + CY2(K)*Y + CY3(K)*X
```

```
      ST(K1,L1) = X1
      ST(L1,K1) = X1
      ST(K1,L2) = X2
      ST(L2,K1) = X2
      ST(K2,L1) = X3
      ST(L1,K2) = X3
      ST(K2,L2) = X4
  300 ST(L2,K2) = X4
      RETURN
      END
```

```
      SUBROUTINE SLCT9
C
C     STIFFNESS SUBROUTINE FOR COMPATIBLE TRIANGULAR PLATE ELEMENT
C     WITH  9 D.F. (3 NODAL POINTS) - RIGHT HANDED COORDINATE SYSTEM -
C     LINEARLY ELASTIC ISOTROPIC MATERIAL
C
      COMMON/CV/NUMEL,NUPTS,NUBPTS,IBANDP,MBAND,NBLOC,NDFRE,NODES,LVECT
      COMMON  TX1(1555),EM,NU,TH   ,AREA,B(3),A(3),ST(10,10),PX1(1813),
     1 P(21,9),H(21),U(21),Q(3,6),TX(3),TY(3),PX2(4577)
      DIMENSION IPERM(3)
      REAL NU, NUH
      DATA IPERM/2,3,1/
      AREA = A(3)*B(2)-A(2)*B(3)
      FAC = EM*TH**3/(864.*AREA**3*(1.-NU**2))
      NUH = 2.*(1.-NU)
      DO 120  I = 1,3
      J = IPERM(I)
      X = A(I)**2+B(I)**2
      U(I) = -(A(I)*A(J)+B(I)*B(J))/X
      TX(I) =  2.*AREA*A(I)/X
      TY(I) = -2.*AREA*B(I)/X
      Q(1,I)    = B(I)*B(I)
      Q(2,I)    = A(I)*A(I)
      Q(3,I)    = B(I)*A(I)
      Q(1,I+3)  = 2.*B(I)*B(J)
      Q(2,I+3)  = 2.*A(I)*A(J)
  120 Q(3,I+3)  = A(I)*B(J)+A(J)*B(I)
      DO 200  I = 1,3
      J = IPERM(I)
      K = IPERM(J)
      II = 3*I
      JJ = 3*J
      KK = 3*K
      A1 = A(I)
      A2 = A(J)
      A3 = A(K)
      B1 = B(I)
      B2 = B(J)
      B3 = B(K)
      U1 = U(I)
      U2 = U(J)
      U3 = U(K)
      W1 = 1.-U1
      W2 = 1.-U2
      W3 = 1.-U3
      B1D = 2.*B1
      B2D = 2.*B2
      B3D = 2.*B3
      A1D = 2.*A1
      A2D = 2.*A2
      A3D = 2.*A3
      C21 = B1-B3*U3           + TX(K)
      C22 = -B1D+B2*W2+B3*U3  + TX(J)-TX(K)
      C31 = A1-A3*U3           + TY(K)
```

```
      C32 = -A1D+A2*W2+A3*U3 + TY(J)-TY(K)
      C51 = B3*W3-B2         + TX(K)
      C52 = B2D-B3*W3-B1*U1  + TX(I)-TX(K)
      C61 = A3*W3-A2         + TY(K)
      C62 = A2D-A3*W3-A1*U1  + TY(I)-TY(K)
      C81 = B3-B2D-B2*U2     + TX(J)
      C82 = B1D-B3+B1*W1     + TX(I)
      C91 = A3-A2D-A2*U2     + TY(J)
      C92 = A1D-A3+A1*W1     + TY(I)
      DO 200  N = 1,3
      L = 6*(I-1) + N
      Q11 = Q(N,I)
      Q22 = Q(N,J)
      Q33 = Q(N,K)
      Q12 = Q(N,I+3)
      Q23 = Q(N,J+3)
      Q31 = Q(N,K+3)
      Q2333 = Q23-Q33
      Q3133 = Q31-Q33
      P(L    ,II-2) = 6.*(-Q11+W2*Q33+U3*Q2333)
      P(L    ,II-1) = C21*Q23+C22*Q33-B3D*Q12+B2D*Q31
      P(L    ,II  ) = C31*Q23+C32*Q33-A3D*Q12+A2D*Q31
      P(L    ,JJ-2) = 6.*(Q22+W3*Q2333)
      P(L    ,JJ-1) = C51*Q2333+B3D*Q22
      P(L    ,JJ  ) = C61*Q2333+A3D*Q22
      P(L    ,KK-2) = 6.*(1.+U2)*Q33
      P(L    ,KK-1) = C81*Q33
      P(L    ,KK  ) = C91*Q33
      P(L+3  ,II-2) = 6.*(Q11+U3*Q3133)
      P(L+3  ,II-1) = C21*Q3133-B3D*Q11
      P(L+3  ,II  ) = C31*Q3133-A3D*Q11
      P(L+3  ,JJ-2) = 6.*(-Q22+U1*Q33+W3*Q3133)
      P(L+3  ,JJ-1) = C51*Q31+C52*Q33+B3D*Q12-B1D*Q23
      P(L+3  ,JJ  ) = C61*Q31+C62*Q33+A3D*Q12-A1D*Q23
      P(L+3  ,KK-2) = 6.*(1.+W1)*Q33
      P(L+3  ,KK-1) = C82*Q33
      P(L+3  ,KK  ) = C92*Q33
      P(N+18,II-2) = 2.*(Q11+U3*Q12+W2*Q31)
      P(N+18,KK-1) = ((B1D-B2D)*Q33+C82*Q23+C81*Q31)/3.
  200 P(N+18,KK  ) = ((A1D-A2D)*Q33+C92*Q23+C91*Q31)/3.
      DO 300  J = 1,9
      DO 240  L = 19,21
      H(L) = 3.*P(L,J)
      DO 240  M = 1,13,6
      N = M + L - 19
      COMM = P(N,J) + P(N+3,J) + P(L,J)
      H(N)   = P(N,J)   + COMM
      H(N+3) = P(N+3,J) + COMM
  240 H(L)   = H(L)     + COMM
      DO 260  N = 1,19,3
      U(N)   = H(N) + NU*H(N+1)
      U(N+1) = H(N+1) + NU*H(N)
  260 U(N+2) = NUH*H(N+2)
      DO 300  I = 1,J
      X = 0.
```

```
      DO 280  N = 1,21
  280 X = X + U(N)*P(N,I)
      ST(I,J) = X*FAC
  300 ST(J,I) = ST(I,J)
      RETURN
      END
```

```
      SUBROUTINE GETNRM (NSP,X,Y,Z,A3,B3,C3,IERROR,XC,YC,ZC)
      DIMENSION X(1), Y(1), Z(1)

      THIS SUBROUTINE COMPUTES THE 3 DIRECTION COSINES (A3,B3,C3)
      OF THE NORMAL TO THE PLANE P WHICH IS THE LEAST SQUARE FIT FOR
      'NSP' SPACE POINTS WHOSE GLOBAL COORDINATES ARE
              X(I), Y(I), Z(I),   I=1 .... NSP
      (A3,B3,C3) ARE REFERRED TO THE (X,Y,Z)  COORDINATE SYSTEM.

      THE 3 DIRECTION COSINES ARE THE ELEMENTS OF THE NORMALIZED
      EIGENVECTOR CORRESPONDING TO THE SMALLEST EIGENVALUE OF THE
      (3 X 3) NORMAL MATRIX CONSTRUCTED WITH THE COORDINATES OF THE
      'NSP' POINTS, AND ARE OBTAINED BY INVERSE ITERATION.

      IF 'NSP' IS LESS THAN 3, THE COMPUTATION IS SKIPPED AND THE
      ERROR FLAG 'IERROR' IS SET TO 1.

      'PRECS' IS THE PRECISION DESIRED FOR THE DIRECTION COSINES AND
      SHOULD BE SET TO 10.**(-N+1), WHERE 'N' IS THE NUMBER OF
      SIGNIFICANT DECIMAL DIGITS CARRIED OUT BY THE COMPUTER IN
      FLOATING POINT ARITHMETIC.

      REAL M12, M13, M23
      IERROR = 0
      IF (NSP.LT.3)  GO TO 200
      PRECS = 1.0E-07
      XX = 0.
      YY = 0.
      ZZ = 0.
      XY = 0.
      YZ = 0.
      ZX = 0.
      DO 100  I = 1,NSP
      X1=X(I)-XC
      Y1=Y(I)-YC
      Z1=Z(I)-ZC
      XX = XX+X1**2
      YY = YY+Y1**2
      ZZ = ZZ+Z1**2
      XY = XY+X1*Y1
      YZ = YZ+Y1*Z1
  100 ZX = ZX+Z1*X1
      NIT = 0
      EPS = 1.0
      A3 = 1.0
      B3 = 1.0
      C3 = 1.0
      S=XX+YY+ZZ
      EV=-S*PRECS
  120 U11 = XX - EV
      U12 = XY
      U13 = ZX
      M12 = XY/U11
      M13 = ZX/U11
```

```
      U22 = YY - EV - M12*U12
      U23 = YZ - M12*U13
      M23 = (YZ - M13*U12)/U22
      U33 = ZZ - EV - M13*U13 - M23*U23
      IF (U33.GT.0.)  GO TO 140
      EV = EV - 0.001/S
      GO TO 120
  140 IF (NIT.EQ.0)  GO TO 180
  160 B3 = B3 - M12*A3
      C3 = C3 - M13*A3 - M23*B3
  180 C3 = C3/U33
      B3 = (B3 - U23*C3)/U22
      A3 = (A3 - U12*B3 - U13*C3)/U11
      S = SQRT(A3**2+B3**2+C3**2)
      A3 = A3/S
      B3 = B3/S
      C3 = C3/S
      NIT = NIT + 1
      IF (NIT.LT.2)  GO TO 190
      EPS = AMAX1 (ABS(A3-AP),ABS(B3-BP),ABS(C3-CP))
      IF (EPS.LT.PRECS.OR.NIT.GT.25)   GO TO 250
  190 AP = A3
      BP = B3
      CP = C3
      IF (EPS.GT.0.005)  GO TO 160
      EV = EV + 0.999/S
      GO TO 120
  200 IERROR = 1
  250 RETURN
      END
```

```
$IBFTC SOLV     DECK
      SUBROUTINE SOLVE
      COMMON M,N,P(100, 3),B(100, 3),A(100,100)
      COMMON/CV/NUMEL,NUPTS,NUBPTS,IBANDP,MBAND,NBLOC,NDFRE,NODES,LVECT
      DIMENSION PV(450,5,3)
      EQUIVALENCE (P,PV)
      M=MBAND
      N=NBLOC
      M1=M-1
      DO 5 I=1,M
      DO 5 J=1,M
    5 A(I,J)=0.
      CALL CHOL(M1)
      REWIND 4
      CALL BPASS(M1)
      REWIND 4
      DO 1 K=1,N
      K1=(N-K)*IBANDP+1
      K2=K1+IBANDP-1
    1 READ(4)(((PV(I,J,L),J=1,NDFRE),I=K1,K2),L=1,LVECT)
      DO 2 LV=1,LVECT
      PRINT 10,LV
      PRINT 11
      DO 2 K=1,NUPTS
    2 PRINT 4,K,(((PV(I,J,L),J=1,NDFRE),I=K,K),L=LV,LV)
    4 FORMAT(I11,5F20.6)
      REWIND 4
      REWIND 1
   10 FORMAT(28H1DISPLACEMENTS FOR LOAD CASE I5)
   11 FORMAT(1H0 6X,4HNODE 9X,2HD118X,2HD218X,2HD318X, 2HD418X,2HD5)
      RETURN
      END
```

```
      SUBROUTINE FORMK(II,M1)
      COMMON M,N,P(100, 3),B(100, 3),A(100,100),D(5,5,6),Q(6)
      COMMON/CV/NUMEL,NUPTS,NUBPTS,IBANDP,MBAND,NBLOC,NDFRE,NODES,LVECT
      INTEGER Q
      DO 1 I=1,M1
      K=I+1
      DO 1 J=K,MBAND
    1 A(I,J)=0.
      DO 2 J=1,MBAND
    2 A(MBAND,J)=0.
      JJ=(II-1)*IBANDP
      DO 7 IK=1,IBANDP
      READ(2) IT,(Q(I),I=1,IT),(((D(I,J,K),I=1,NDFRE),J=1,NDFRE),K=1,IT)
      DO 7 K=1,IT
      IX=(Q(K)-JJ-1)*NDFRE
      JX=(IK-1)*NDFRE
      I1=NDFRE
      IF(IX.GE.MBAND)  IX=IX-MBAND
      DO 7 L=1,NDFRE
      IF(IX.EQ.JX)  I1=L
      LX=IX+L
      DO 7 I=1,I1
      MX=JX+I
    7 A(LX,MX)=D(L,I,K)+A(LX,MX)
      RETURN
      END
```

```
      SUBROUTINE CHOL(M1)
      COMMON M,N,P(100, 3),B(100, 3),A(100,100)
      COMMON/CV/NUMEL,NUPTS,NUBPTS,IBANDP,MBAND,NBLOC,NDFRE,NODES,LVECT
      DO 11 ICHOL=1,N
      CALL FORMK(ICHOL,M1)
      DO 5 J=1,M
      L1=J-1
      DO 5 I=J,M
      SUM = 0.
    2 IF(J.EQ.1)  GOTO 4
      DO 3 L=1,L1
    3 SUM = SUM+A(J,L)*A(I,L)
    4 IF(I.EQ.J)  A(I,J) = SQRT(A(I,J)-SUM)
    5 IF(I.NE.J)  A(I,J) = (A(I,J)-SUM)/A(J,J)
      BEGIN DECOMPOSITION OF LOWER TRIANGLE CASE 1
      IF(ICHOL.EQ.N)  GOTO 8
      DO 7 I=1,M1
      K1=I+1
      DO 7 J=K1,M
      K2=J-1
      SUM=0.0
      IF(K1.GT.K2)  GOTO 7
      DO 6 K=K1,K2
    6 SUM = SUM+A(J,K)*A(I,K)
    7 A(I,J) = (A(I,J)-SUM)/A(J,J)
    8 CALL FPASS(ICHOL,M1)
      WRITE(1)((A(I,J),J=1,M),I=1,M),((B(K,L),K=1,M),L=1,LVECT)
      IF(ICHOL.EQ.N)  GOTO 11
      DO 10 J=1,M1
      DO 10 I=J,M1
      K1=I+1
      SUM=0.
      DO 9 K=K1,M
    9 SUM=SUM+A(J,K)*A(I,K)
   10 A(I,J)=-SUM
   11 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SWITCH
      COMMON M,N,P(100, 3),B(100, 3),A(100,100)
      COMMON/CV/NUMEL,NUPTS,NUBPTS,IBANDP,MBAND,NBLOC,NDFRE,NODES,LVECT
      MD=M/2
      MJ=M+1
      DO 1 I=1,MD
      LI=M-I+1
      DO 3 LV=1,LVECT
      C=P(I,LV)
      P(I,LV)=P(LI,LV)
    3 P(LI,LV)=C
      MJ=MJ-1
      DO 1 J=I,MJ
      LJ=M-J+1
      C=A(J,I)
      A(J,I)=A(LI,LJ)
      A(LI,LJ)=C
    1 CONTINUE
      MJ=M+1
      DO 2 I=1,MD
      LI=M-I+1
      MJ=MJ-1
      JJ=I+1
      DO 2 J=JJ,MJ
      LJ=M-J+1
      C=A(I,J)
      A(I,J) = A(LJ,LI)
      A(LJ,LI) = C
    2 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE FPASS(ICHOL,M1)
      COMMON M,N,P(100, 3),B(100, 3),A(100,100)
      COMMON/CV/NUMEL,NUPTS,NUBPTS,IBANDP,MBAND,NBLOC,NDFRE,NODES,LVECT
      IF(ICHOL.EQ.1)  READ(4)((P(I,J),I=1,M),J=1,LVECT)
      DO 20 LV=1,LVECT
   20 B(1,LV)=P(1,LV)/A(1,1)
      DO 2 LV=1,LVECT
      DO 2 I=2,M
      K=I-1
      DO 1 J=1,K
    1 P(I,LV)=P(I,LV)-A(I,J)*B(J,LV)
    2 B(I,LV)=P(I,LV)/A(I,I)
      IF(ICHOL.EQ.N)  GOTO 4
      READ(4) ((P(I,J),I=1,M),J=1,LVECT)
      DO 3 LV=1,LVECT
      DO 3 I=1,M1
      L=I+1
      DO 3 J=L,M
    3 P(I,LV)=P(I,LV)-A(I,J)*B(J,LV)
    4 RETURN
      END
```

```
      SUBROUTINE BPASS(M1)
      COMMON M,N,P(100, 3),B(100, 3),A(100,100)
      COMMON/CV/NUMEL,NUPTS,NUBPTS,IBANDP,MBAND,NBLOC,NDFRE,NODES,LVECT
      DO 7 II=1,N
      BACKSPACE 1
      IF(II.GT.1)  BACKSPACE 1
      READ(1)((A(I,J),J=1,M),I=1,M),((P(K,L),K=1,M),L=1,LVECT)
      CALL SWITCH
      IF(II.EQ.1)  GOTO 3
      DO 2 LV=1,LVECT
      DO 2 I=1,M1
      L=I+1
      DO 2 J=L,M
    2 P(I,LV)=P(I,LV)-A(I,J)*B(J,LV)
    3 DO 10 LV=1,LVECT
   10 B(1,LV)=P(1,LV)/A(1,1)
      DO 5 LV=1,LVECT
      DO 5 I=2,M
      K=I-1
      DO 4 J=1,K
    4 P(I,LV)=P(I,LV)-A(I,J)*B(J,LV)
    5 B(I,LV)=P(I,LV)/A(I,I)
      DO 6 LV=1,LVECT
      DO 6 I=1,M
      IZ=M-I+1
    6 P(I,LV)=B(IZ,LV)
    7 WRITE(4)((P(I,J),I=1,M),J=1,LVECT)
      REWIND 4
      RETURN
      END
```

```
      SUBROUTINE STRESS
C.....................................................................................
      COMMON/CV/NUMEL,NUPTS,NUBPTS,IBANDP,MBAND,NBLOC,NDFRE,NODES,LVECT
      COMMON IX1(2),BX(450,5,3),       UM(5,4),VM(5,4),BP(9,4),TD(3,36),
     1 TR(3,36),AD(3,4),BD(3,4),S(33,13),PP(400,6),ZZ(3,3,4),ZT(3,3,4),
     2 ZQ(3,5),GA(3,3,4),G1(3,3,4),G3(3,4),AP(21),GQ(3,5),
     3 IQ(4),IC(400),NTRI,NQUAD
      COMMON AJ,AK,BJ,BK,EM,NU,TH,P(21,9),TQ(3,3,4)
      COMMON/CQ6/ PT(13)
      REAL NU
C.....................................................................................
      DO 7 LV=1,LVECT
      REWIND 1
C.....INITIALIZE PP,THE MATRIX STORING AVERAGED MOMENTS AND STESSES........
      DO 1 I=1,NUPTS
      IC(I)=0
      DO 2 K=1,5
    2 BX(I,K,1)=BX(I,K,LV)
      DO 1 J=1,6
    1 PP(I,J)=0.
C.....SUM OVER NUMBER OF ELEMENTS.............................................
      PRINT 90,LV
      PRINT 93
      DO 6 JQ=1,NUMEL
      READ (1)  IQ,EM,NU,TH,AD,BD,TD,TR,TQ
      NODES=3
      IF(IQ(4).GE.1) NODES=4
      DO 8 I=1,NODES
      K=IQ(I)
    8 IC(K)=IC(K)+1
      IF(IQ(4).GE.1)  READ (1)  S,PT
      NQUAD=1
      IF(IQ(4).GE.1)  NQUAD=4
C.....COMPUTE INTERIOR NODAL PT. DISPL. FOR QUAD. AND NODAL PT. DISPL. FOR
C.....TRI. IN ELEMENT COORDS..................................................
      CALL GDISPL
C.....COMPUTE AND TRANSFORM ELEMENT STRESSES AND MOMENTS.................
      DO 3 NTRI=1,NQUAD
      CALL MEMB10
    3 CALL MOMENT
C.....PRINT ELEMENT STRESSES AND MOMENTS IF ELEMENT IS A TRIANGLE.........
      IF(NQUAD.EQ.1)  CALL PRTRI
C.....AVERAGE AND PRINT STRESSES AND MOMENTS IF ELEMENT IS A QUAD.........
    6 IF (NQUAD.EQ.4)  CALL PQUAD (JQ)
C.....AVERAGE AND PRINT NODAL PT. MOMENTS AND STRESSES...................
      PRINT 91,LV
      PRINT 92
      DO 5 I=1,NUPTS
      XP=IC(I)
      DO 4 J=1,6
    4 PP(I,J)=PP(I,J)/XP
    5 PRINT 110,I,PP(I,2),PP(I,1),(PP(I,L),L=3,6)
    7 CONTINUE
      RETURN
```

```
 90 FORMAT (1H16X 39HELEMENT STRESS RESULTANTS FOR LOAD CASE , I3,
   . 5X 26H(WRT AVERAGE PLANE COORDS) )
 91 FORMAT (1H16X46HAVERAGED NODAL STRESS RESULTANTS FOR LOAD CASE I3,
   . 5X 20H(WRT SURFACE COORDS) )
 92 FORMAT(1H0 7X,4HNODE 6X,2HM1 10X,2HM2 10X,3HM12 9X,2HN1 10X,2HN2
  1 10X,1HS)
 93 FORMAT (1H0,10X,7HELEMENT,6X,2HM1,10X,2HM2 10X,3HM12 9X,2HN1 10X,
   . 2HN2 10X,1HS)
110 FORMAT(7H       ,I5,3X,6F12.4)
    END
```

```
      SUBROUTINE GDISPL
C.......................................................................
      COMMON  IX1(2),B(450,5),PX1(4500),UM(5,4),VM(5,4),BP(9,4),TD(3,36),
     1 TR(3,36),AD(3,4),BD(3,4),S(33,13),PP(400,6),ZZ(3,3,4),ZT(3,3,4),
     2 ZQ(3,5),GA(3,3,4),G1(3,3,4),G3(3,4),AP(21),GQ(3,5),
     3 IQ(4),IC(400),NTRI,NQUAD
      COMMON  PX2(196),TQ(3,3,4),P(39),R2(13),D(5,5,4)
      COMMON/CQ6/ PT(13)
C.......................................................................
      IP=IQ(1)
      JP=IQ(2)
      KP=IQ(3)
      IF(NQUAD.EQ.4)  GOTO 10
      DO 11 I=1,5
      D(I,1,1)=B(IP,I)
      D(I,2,1)=B(JP,I)
   11 D(I,3,1)=B(KP,I)
      GOTO 12
   10 LP=IQ(4)
C.....GROUP DISPL. OF CORNER NODES IN P.................................
      DO 1 M=1,3
      P(M   )=TQ(M,1,1)*B(IP,1)+TQ(M,2,1)*B(IP,2)+TQ(M,3,1)*B(IP,3)
      P(M+5 )=TQ(M,1,2)*B(JP,1)+TQ(M,2,2)*B(JP,2)+TQ(M,3,2)*B(JP,3)
      P(M+10)=TQ(M,1,3)*B(KP,1)+TQ(M,2,3)*B(KP,2)+TQ(M,3,3)*B(KP,3)
    1 P(M+15)=TQ(M,1,4)*B(LP,1)+TQ(M,2,4)*B(LP,2)+TQ(M,3,4)*B(LP,3)
      DO 4 M=4,5
      P(M   )=B(IP,M)
      P(M+5 )=B(JP,M)
      P(M+10)=B(KP,M)
    4 P(M+15)=B(LP,M)
C.....COMPUTE DISPL. AT INTERIOR NODES..................................
      DO 3 I=1,13
      L=19+I
      R2(I)=PT(I)/S(L+1,I)
      DO 2 K=1,L
    2 R2(I)=R2(I)-S(K,I)*P(K)
    3 P(I+20)=R2(I)
C.....STORE ALL THREE DISPL. COMPONENTS AT MID SIDE NODES IN P..........
      DO 5 I=1,4
      J=(I-1)*2
      K=(I-1)*3
      P(K+26)=R2(J+6)
    5 P(K+27)=R2(J+7)
      P(28)=(P( 3)+P(23))/2.
      P(31)=(P( 8)+P(23))/2.
      P(34)=(P(13)+P(23))/2.
      P(37)=(P(18)+P(23))/2.
C.....STORE DISPL. COMPONENTS FOR EACH TRI. IN D........................
      DO 6 I=1,5
      D(I,1,1)=B(IP,I)
      D(I,2,1)=B(JP,I)
      D(I,4,1)=P(I+28)
      D(I,5,1)=P(I+25)
      D(I,3,1)=P(I+20)
```

```
      D(I,1,2)=B(JP,I)
      D(I,2,2)=B(KP,I)
      D(I,4,2)=P(I+31)
      D(I,5,2)=P(I+28)
      D(I,3,2)=P(I+20)
      D(I,1,3)=B(KP,I)
      D(I,2,3)=B(LP,I)
      D(I,4,3)=P(I+34)
      D(I,5,3)=P(I+31)
      D(I,3,3)=P(I+20)
      D(I,1,4)=B(LP,I)
      D(I,2,4)=B(IP,I)
      D(I,4,4)=P(I+25)
      D(I,5,4)=P(I+34)
    6 D(I,3,4)=P(I+20)
C.....TRANSFORM NODAL PT. DISPL. TO ELEMENT COORDS.....................
   12 DO 8 K=1,NQUAD
      KK=(K-1)*9
      DO 7 J=1,5
      IF(J.LT.4)  L=KK+(J-1)*3
      UM(J,K)=TD(1,L+1)*D(1,J,K)+TD(1,L+2)*D(2,J,K)+TD(1,L+3)*D(3,J,K)
      VM(J,K)=TD(2,L+1)*D(1,J,K)+TD(2,L+2)*D(2,J,K)+TD(2,L+3)*D(3,J,K)
      IF(J.GT.3)  GOTO 7
      I=(J-1)*3
      BP(I+1,K)=TD(3,L+1)*D(1,J,K)+TD(3,L+2)*D(2,J,K)+TD(3,L+3)*D(3,J,K)
      BP(I+2,K)=TR(1,L+1)*D(4,J,K)+TR(1,L+2)*D(5,J,K)
      BP(I+3,K)=TR(2,L+1)*D(4,J,K)+TR(2,L+2)*D(5,J,K)
    7 CONTINUE
    8 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MEMB10
C.......................................................................
      COMMON IX1(2),B(450,5),PX1(4500),UM(5,4),VM(5,4),BP(9,4),TD(3,36),
     1 TR(3,36),AD(3,4),BD(3,4),S(33,13),PP(400,6),ZZ(3,3,4),ZT(3,3,4),
     2 ZQ(3,5),GA(3,3,4),G1(3,3,4),G3(3,4),AP(21),GQ(3,5),
     3 IQ(4),IC(400),NTRI,NQUAD
      COMMON AJ,AK,BJ,BK,FM,NU,TH,P(21,9)
      COMMON PX2(188),U(5),V(5),X(3),Y(3),Z(3)
      REAL NU
C.......................................................................
C.....GROUP MEMBRANE DISPL. IN U AND V..................................
      DO 1 I=1,5
      U(I)=UM(I,NTRI)
    1 V(I)=VM(I,NTRI)
C.....MODIFY STRAIN DISPL. MATRIX IF ELEMENTT IS A CST..................
      IF(IQ(4).GT.0)  GOTO 2
      U(4)=(U(2)+U(3))/2.
      V(4)=(V(2)+V(3))/2.
      U(5)=(U(1)+U(3))/2.
      V(5)=(V(1)+V(3))/2.
C.....COMPUTE STRAINS IN X DIR..........................................
    2 F=BD(1,NTRI)
      X(1)=(U(1)-U(2))*F
      X(2)=(U(1)-U(2))*F
      X(3)=(-U(1)+U(2)+4.*(-U(4)+U(5)))*F
      A=AD(1,NTRI)
      C=AD(2,NTRI)
      D=AD(3,NTRI)
C.....COMPUTE STRAINS IN Y DIR..........................................
      Y(1)=(A-2.*D)*V(1)+C*V(2)-D*V(3)+4.*D*V(5)
      Y(2)=A*V(1)+(C-2.*D)*V(2)-D*V(3)+4.*D*V(4)
      Y(3)=-A*V(1)-C*V(2)+3.*D*V(3)+4.*C*V(4) +4.*A*V(5)
C.....COMPUTE SHEAR STRAINS.............................................
      Z(1)=(V(1)-V(2))*F+(A-2.*D)*U(1)+C*U(2)-D*U(3)+4.*D*U(5)
      Z(2)=(V(1)-V(2))*F+A*U(1)+(C-2.*D)*U(2)-D*U(3)+4.*D*U(4)
      Z(3)=(-V(1)+V(2)+4.*(-V(4)+V(5)))*F
     1   -A*U(1)-C*U(2)+3.*D*U(3)+4.*C*U(4)+4.*A*U(5)
C.....COMPUTE STRESSES..................................................
      D=-FM*TH/(D*F*(1.-NU**2))
      DO 3 I=1,3
      ZZ(1,I,NTRI)=X(I)*D+Y(I)*D*NU
C.....TRANSFORM STRESSES TO SURFACE COORDS..............................
      ZZ(2,I,NTRI)=X(I)*D*NU+Y(I)*D
    3 ZZ(3,I,NTRI)=.5*Z(I)*(1.-NU)*D
      K=NTRI
      KK=(NTRI-1)*9
      DO 4 J=1,3
      L=KK+(J-1)*3
      C=TR(1,L+1)
      R=TR(1,L+2)
      CS=SQRT(C**2+R**2)
      C=C/CS
      R=R/CS
      F=ZZ(1,J,K)
```

```
      G=ZZ(2,J,K)
      H=ZZ(3,J,K)
      ZT(1,J,K)=F*C**2+G*R**2-H*2.*R*C
      ZT(2,J,K)=F*R**2+G*C**2+H*2.*R*C
    4 ZT(3,J,K)=F*R*C-G*R*C+H*(C**2-R**2)
      RETURN
      END
```

```
      SUBROUTINE MOMENT
C.......................................................................
      COMMON IX1(2),B(450,5),PX1(4500),UM(5,4),VM(5,4),BP(9,4),TD(3,36),
     1 TR(3,36),AD(3,4),BD(3,4),S(33,13),PP(400,6),ZZ(3,3,4),ZT(3,3,4),
     2 ZQ(3,5),GA(3,3,4),G1(3,3,4),G3(3,4),AP(21),GQ(3,5),
     3 IQ(4),IC(400),NTRI,NQUAD
      COMMON AJ,AK,BJ,BK,EM,NU,TH,P(21,9),PX2(207),GG(3,2,3),BQ(9)
      REAL NU
C.......................................................................
C.....PLACE NODAL PT. DISPL. IN BQ......................................
      DO 1 I=1,9
    1 BQ(I)=BP(I,NTRI)
      DO 20 I=1,7,3
   20 BQ(I)=-BQ(I)
C.....COMPUTE MOMENTS FOR SUBLEMENTS AND STORE IN AP.....................
      AJ=AD(3,NTRI)
      AK=AD(1,NTRI)+AJ
      BJ=BD(3,NTRI)
      BK=BD(2,NTRI)
      CALL MOMT
      CALL MATMU3 ( P,BQ,AP,21,9)
C.....GROUP MOMENTS FOR SUB ELEMENTS IN GG...............................
      DO 2 J=1,3
      K=(J-1)*6
      DO 2 L=1,2
      M=(L-1)*3+K+1
      GG(J,L,1)=AP(M)
      GG(J,L,2)=AP(M+1)
    2 GG(J,L,3)=AP(M+2)
C.....DETERMINE AVERAGE MOMENTS FROM SUB ELEMENTS AT I,J,K..............
      K=NTRI
      DO 3 J=1,3
      GA(J,1,K)=(GG(1,1,J)+GG(3,2,J))/2.
      GA(J,2,K)=(GG(1,2,J)+GG(2,1,J))/2.
    3 GA(J,3,K)=(GG(2,2,J)+GG(3,1,J))/2.
C.....TRANSFORM AVERAGE MOMENTS TO SURFACE COORDS.........................
      KK=(NTRI-1)*9
      DO 4 J=1,3
      L=KK+(J-1)*3
      C=TR(1,L+1)
      R=TR(1,L+2)
      CS=SQRT(C**2+R**2)
      C=C/CS
      R=R/CS
      X=GA(1,J,K)
      Y=GA(2,J,K)
      Z=GA(3,J,K)
      G1(1,J,K)=X*C**2+Y*R**2-Z*2.*R*C
      G1(2,J,K)=X*R**2+Y*C**2+Z*2.*R*C
    4 G1(3,J,K)=X*R*C -Y*R*C+Z*(C**2-R**2)
      G3(1,K)=AP(19)*C**2+AP(20)*R**2-AP(21)*2.*R*C
      G3(2,K)=AP(19)*R**2+AP(20)*C**2+AP(21)*2.*R*C
      G3(3,K)=AP(19)*R*C-AP(20)*R*C+AP(21)*(C**2-R**2)
      RETURN
      END
```

```
      SUBROUTINE PRTRI
C.........................................................................
      COMMON IX1(2),B(450,5),PX1(4500),UM(5,4),VM(5,4),BP(9,4),TD(3,36),
     1 TR(3,36),AD(3,4),BD(3,4),S(33,13),PP(400,6),ZZ(3,3,4),ZT(3,3,4),
     2 ZQ(3,5),GA(3,3,4),G1(3,3,4),G3(3,4),AP(21),GQ(3,5),
     3 IQ(4),IC(400),NTRI,NQUAD
C.........................................................................
C.....ADD NODAL PT. MOMENTS AND STRESSES IN PP.............................
      DO 1 J=1,3
      K=IQ(J)
      DO 1 L=1,3
      M=L+3
      PP(K,L)=PP(K,L)+G1(L,J,1)
    1 PP(K,M)=PP(K,M)+ZT(L,J,1)
C.....PRINT MEMBRANE STRESSES IN ELEMENT COORDS............................
      PRINT 10,(((ZZ(I,J,K),K=1,1),J=1,1),I=1,3)
C.....PRINT MOMENTS AT CENTROID IN ELEMENT AND SURFACE COORDS..............
      PRINT 12,(AP(I),I=19,21),((G3(I,J),J=1,1),I=1,3)
      RETURN
   10 FORMAT(3E12.4)
   12 FORMAT(6E12.4)
      END
```

```
      SUBROUTINE PQUAD (JQ)
C.........................................................................
      COMMON IX1(2),B(450,5),PX1(4500),UM(5,4),VM(5,4),BP(9,4),TD(3,36),
     1 TR(3,36),AD(3,4),BD(3,4),S(33,13),PP(400,6),ZZ(3,3,4),ZT(3,3,4),
     2 ZQ(3,5),GA(3,3,4),G1(3,3,4),G3(3,4),AP(21),GQ(3,5),
     3 IQ(4),IC(400),NTRI,NQUAD
C.........................................................................
C.....COMPUTE AVERAGE MOMENTS AND STRESSES FOR QUAD.......................
      DO 1 I=1,3
      GQ(I,1)=(G1(I,1,1)+G1(I,2,4))/2.
      GQ(I,2)=(G1(I,2,1)+G1(I,1,2))/2.
      GQ(I,3)=(G1(I,2,2)+G1(I,1,3))/2.
      GQ(I,4)=(G1(I,2,3)+G1(I,1,4))/2.
      GQ(I,5)=(G1(I,3,1)+G1(I,3,2)+G1(I,3,3)+G1(I,3,4))/4.
      ZQ(I,1)=(ZT(I,1,1)+ZT(I,2,4))/2.
      ZQ(I,2)=(ZT(I,2,1)+ZT(I,1,2))/2.
      ZQ(I,3)=(ZT(I,2,2)+ZT(I,1,3))/2.
      ZQ(I,4)=(ZT(I,2,3)+ZT(I,1,4))/2.
    1 ZQ(I,5)=(ZT(I,3,1)+ZT(I,3,2)+ZT(I,3,3)+ZT(I,3,4))/4.
C.....ADD NODAL PT. MOMENTS AND STRESSES IN PP............................
      DO 2 J=1,4
      K=IQ(J)
      DO 2 L=1,3
      M=L+3
      PP(K,L)=PP(K,L)+GQ(L,J)
    2 PP(K,M)=PP(K,M)+ZQ(L,J)
C.....PRINT AVERAGE MOMENTS AND STRESSES FOR QUAD.........................
      M=JQ
      PRINT 10,M,GQ(2,5),GQ(1,5),GQ(3,5),(ZQ(I,5),I=1,3)
      RETURN
   10 FORMAT (10X,I5,6X,6E12.4)
      END
```

```
      SUBROUTINE MOMT
      COMMON IX1(2),PX1(10102),IX2(406)
      COMMON AJ,AK,BJ,BK,EM,NU,TH,P(21,9),PX2(234),H(9,21),PO(21,3),
     1 G(3,9),A(3),B(3),BS(3),AS(3),HG(3),U(3),W(3),CN(3),SN(3)
      DIMENSION R(21,9),IPERM(3)
      DATA IPERM/2,3,1/
      REAL NU,NUT
      EQUIVALENCE (P,R)

      INITIALIZE

      A(1)=AK-AJ
      A(2)=-AK
      A(3)=AJ
      B(1) = BJ-BK
      B(2) = BK
      B(3) = -BJ
      AREA = AJ*BK-BJ*AK
      DO 110  I = 1,3
      J = IPERM(I)
      K = IPERM(J)
      X = A(I)**2+B(I)**2
      W(I) = -(A(I)*A(K)+B(I)*B(K))/X
      U(I) = 1.-W(I)
      X = SQRT(X)
      HG(I) =-AREA/X
      CN(I) =  A(I)/X
  110 SN(I) = -B(I)/X
      C13 = 1./3.
      C16 = 1./6.
      C227 = 2./27.
      AREA12 =  6.*AREA
      AREA6 = 3.*AREA
      AREA36 = 18.*AREA
      DO 120  I = 1,21
      DO 115  J = 1,3
  115 PO(I,J) = 0.
      DO 120  J = 1,9
  120 P(I,J) = 0.

      FORMATION OF CORNER CURVATURE-NODAL DISPLACEMENT MATRIX P

      DO 150  I = 1,3
      J = IPERM(I)
      K = IPERM(J)
      M3 = 3*I
      M2 = M3-1
      M1 = M2-1
      N3 = 3*J
      N2 = N3-1
      N1 = N2-1
      B1 = B(I)
      B2 = B(J)
      B3 = B(K)
```

```
      A1 = A(I)
      A2 = A(J)
      A3 = A(K)
      W2 = W(J)
      W3 = W(K)
      U2 = U(J)
      U3 = U(K)
      B32 = B3-B2
      A32 = A3-A2
      BW = B3*W3
      BU = B2*U2
      AW = A3*W3
      AU = A2*U2
      H2 = 2.*HG(J)
      H3 = 2.*HG(K)
      C2 = CN(J)
      C3 = CN(K)
      S2 = SN(J)
      S3 = SN(K)
      G(1,M1) = C13 + C227*(W2-W3)
      G(2,M1) = -(5.*A1+AW+AU)/AREA6
      G(3,M1) =  (5.*B1+BW+BU)/AREA6
      G(1,M2) = (4.*B32+BU-BW+H2*C2+H3*C3)/81.
      G(1,M3) = (4.*A32+AU-AW+H2*S2+H3*S3)/81.
      X = A2*B2*(1.+W2) - A3*B3*(1.+U3)
      G(2,M2) = -(X+4.*A1*B32)/AREA36 + C16 - (C2*C2+C3*C3)/9.
      G(3,M3) =  (X+4.*B1*A32)/AREA36 + C16 - (S2*S2+S3*S3)/9.
      X = (C2*S2+C3*S3)/9.
      G(2,M3) = (-2.*A1*A32+C13*(AU*A2-AW*A3))/AREA12 - X
      G(3,M2) =  (2.*B1*B32+C13*(BW*B3-BU*B2))/AREA12 - X
      AS(3) = A(K)
      AS(1) = (A(I)-A(K))/3.
      AS(2) = (A(J)-A(K))/3.
      BS(3) = B(K)
      BS(1) = (B(I)-B(K))/3.
      BS(2) = (B(J)-B(K))/3.
      W3 = (1.+W(K))/3.
      U3 = 1.-W3
      DO 140  L = 1,3
      H(L,1) = BS(L)**2
      H(L,2) = AS(L)**2
      H(L,3) = 2.*AS(L)*BS(L)
      M = IPERM(L)
      N = L + 3
      H(N,1) = 2.*BS(L)*BS(M)
      H(N,2) = 2.*AS(L)*AS(M)
  140 H(N,3) =(AS(L)*BS(M)+AS(M)*BS(L))*2.
      B3 = 2.*BS(3)
      A3 = 2.*AS(3)
      B2 = 2.*BS(2)
      A2 = 2.*AS(2)
      B1 = 2.*BS(1)
      A1 = 2.*AS(1)
      HTC = C13*H3*C3
      HTS=C13*H3*S3
```

```
      BU=BS(3)*U3-BS(1)+HTC
      AU=AS(3)*U3-AS(1)+HTS
      BW=BS(2)-BS(3)*W3+HTC
      AW=AS(2)-AS(3)*W3+HTS
      L=6*(I-1)
      DO 150  J = 1,3
      L1 = L+J
      L2 = L1+3
      L3 = J+18
      H1 =  H(1,J)
      H2 =  H(2,J)
      H3 =  H(3,J)
      H12 = H(4,J)
      H23 = H(5,J)
      H31 = H(6,J)
      P(L1,M1) = 6.*(H1+H12+U3*H23+H31)
      P(L1,M2) = B3*H12+BU*H23-B2*H31
      P(L1,M3) = A3*H12+AU*H23-A2*H31
      P(L1,N1) = 6.*(H2+W3*H23)
      P(L1,N2) = -B3*H2+BW*H23
      P(L1,N3) = -A3*H2+AW*H23
      PO(L1,1) = 6.*H3
      PO(L1,2) = B2*H3
      PO(L1,3) = A2*H3
      P(L2,M1) = 6.*(H1+U3*H31)
      P(L2,M2) = B3*H1+BU*H31
      P(L2,M3) = A3*H1+AU*H31
      P(L2,N1) = 6.*(H2+H12+H23+W3*H31)
      P(L2,N2) = -B3*H12+B1*H23+BW*H31
      P(L2,N3) = -A3*H12+A1*H23+AW*H31
      PO(L2,1) = 6.*H3
      PO(L2,2) = -B1*H3
      PO(L2,3) = -A1*H3
      IF (I.GT.1)  GO TO 150
      P(L3,M1) = 6.*(H1+U3*H12)
      P(L3,M2) = -B2*H1+BU*H12
      P(L3,M3) = -A2*H1+AU*H12
      P(L3,N1) = 6.*(H2+W3*H12)
      P(L3,N2) = B1*H2+BW*H12
      P(L3,N3) = A1*H2+AW*H12
      PO(L3,1) = 6.*(H3+H23+H31)
      PO(L3,2) = -B1*H23+B2*H31
      PO(L3,3) = -A1*H23+A2*H31
  150 CONTINUE

      CONDENSATION OF CENTROID DISPLACEMENTS, MULTIPLICATION BY
      MOMENT-CURVATURE LAW AND STRAIN ENERGY INTEGRATION

      FAC = 9./AREA**2
      NUT = 12.*(1.-NU**2)
      D = EM*TH**3/NUT
      GX = D*(1.-NU)/2.
      DO 250  J = 1,9
      DO 180  I = 1,21
      DO 170  K = 1,3
```

```
170 P(I,J) = P(I,J) + PO(I,K)*G(K,J)
180 P(I,J) = P(I,J)*FAC
    DO 250  M1= 1,19,3
    M2 = M1 + 1
    M3 = M2 + 1
    B1 = P(M1,J)
    B2 = P(M2,J)
    R(M1,J) = D*(B1+NU*B2)
    R(M2,J) = D*(B2+NU*B1)
250 R(M3,J) = GX*P(M3,J)
    RETURN
    END
```

```
      SUBROUTINE MATMU3(A,B,C,I,J)
      DIMENSION A(I,J),B(J),C(I)
      DO 1 K=1,I
      C(K)=0.0
      DO 1 L=1,J
      C(K)=C(K)+A(K,L)*B(L)
    1 CONTINUE
      RETURN
      END
```

```
   *  TOROIDAL SHELL  *  MERIDIONAL STIFFENER
 40    54   26   20    5     1
 1    4   207.8472  120.                    30.      60.      60.
 9          60.     60.
10    4   221.7312  91.8432                 22.5     60.      60.
18          60.     60.
19    4   231.8232  62.1168                 15.      72.      72.
27          72.      72.
28    4   231.8232  62.1168                 15.      60.      60.
36          60.      60.
37    4   237.9456  31.3272                 7.5      60.      60.
45          60.     60.
46    4   240.                                       60.      60.
54          60.     60.

 1    1  10  11   2   8   1
 9   10  28  29  11   8   1
17   19  28  29  20   8   3
 1        29600.    1.75      .3
40        29600.    1.75      .3
 111011                                                           8   9
4611011                                                           8  54
 201010                                                           1   8
4701010                                                           1  53
1010011                                                           9  37
1810011                                                           9  45
  1                    -1.      8
 17                                                               1  24
```