1969-12

A curvilinear shell infinite element.

Kiess, Dean William

Monterey, California. U.S. Naval Postgraduate School

https://hdl.handle.net/10945/11909

A CURVILINEAR SHELL FINITE ELEMENT

by

Dean William Kiess

United States Naval Postgraduate School

THESIS

A CURVILINEAR SHELL FINITE ELEMENT

by

Dean William Kiess

December 1969

A Curvilinear Shell Finite Element

by

Dean William Kiess
Lieutenant, United States Navy
B.S., Pennsylvania State University, 1962

Submitted in partial fulfillment of the
requirements for the degrees of

MECHANICAL ENGINEER

and

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL

December 1969

ABSTRACT

A doubly curved element for a shell of revolution which has arbitrarily curved meridians is developed and analyzed. Meridional curvature is calculated using a highly accurate polynomial approximation. The displacement functions selected satisfy interelement compatibility and contain all the lower modes of a complete set of straining functions. Non-straining modes corresponding to rigid body motions are introduced into the final stiffness matrix for any conceivable rigid body motions.

The direct stiffness method was used to construct a stress and strain analysis program and the results of the analysis of a few problems are compared to classical solutions to establish the integrity of the element.

TABLE OF CONTENTS

I. INTRODUCTION ---------- 7

A. GENERAL DESCRIPTION ---------- 7

B. HISTORICAL BACKGROUND ---------- 7

C. OBJECTIVE ---------- 9

II. THE FINITE ELEMENT METHOD ---------- 10

A. GENERAL ---------- 10

B. STRUCTURAL IDEALIZATION ---------- 10

C. EVALUATION OF ELEMENT STIFFNESS PROPERTIES ---- 11

D. EVALUATION OF THE FORCE-DISPLACEMENT BEHAVIOR OF THE IDEALIZED STRUCTURE ---------- 11

E. EVALUATION OF ELEMENT STRESS AND STRAIN ------- 11

III. CONSTRUCTION OF THE STIFFNESS MATRIX ---------- 12

A. GENERAL ---------- 12

B. GEOMETRIC DERIVATION ---------- 12

C. DISPLACEMENT FUNCTIONS ---------- 15

1. Application to the Curvilinear Shell Finite Element ---------- 16

D. RIGID BODY MODES ---------- 16

1. Application to the Curvilinear Shell Finite Element ---------- 17

E. CLASSICAL SHELL THEORY ---------- 19

F. NODAL COORDINATES ---------- 22

G. STIFFNESS MATRIX ---------- 22

H. DEFINITION OF MATRICES ---------- 24

I. LOAD VECTORS ---------------------------------- 25
1. Concentrated Load Vectors ------------------- 25
2. Pressure Load Vectors ----------------------- 25
IV. NUMERICAL CONSIDERATIONS AND PROGRAM INFORMATION - 35
A. GENERAL -- 35
B. QUADRATURE ------------------------------------- 35
C. LIMITING CASES --------------------------------- 36
D. CASE WHERE ϕ EQUALS ZERO ------------------ 36
E. INITIAL CONSTRUCTION OF THE STIFFNESS MATRIX ---- 36
F. DIRECT STIFFNESS PROGRAM ----------------------- 37
V. DISCUSSION -------------------------------------- 39
A. INTEGRITY OF THE CURVILINEAR SHELL FINITE ELEMENT - 39
B. NUMERICAL SOLUTION OF CLASSICAL PROBLEMS ------- 39
1. Pinched Cylinder ------------------------------ 39
2. Spherical Cap --------------------------------- 41
C. CONCLUSIONS ------------------------------------ 43
D. RECOMMENDATIONS -------------------------------- 44
APPENDIX A NUMERICAL EXAMPLE OF THE STIFFNESS MATRIX --- 45
APPENDIX B COMPUTER OUTPUT ------------------------ 52
APPENDIX C COMPUTER PROGRAM ----------------------- 57
BIBLIOGRAPHY --------------------------------------- 83
INITIAL DISTRIBUTION LIST -------------------------- 84
FORM DD 1473 --------------------------------------- 85

ACKNOWLEDGEMENTS

The author wishes to express his appreciation to his advisor, Professor Gilles Cantin of the Naval Postgraduate School, for his constructive supervision and enthusiastic encouragement throughout this work.

The Naval Postgraduate School Computer Center provided facilities and technical advice for computer work.

I. INTRODUCTION

A. GENERAL DESCRIPTION

The concept of dividing complex structures into an assemblage of individual structural components or elements is axiomatic to the structural engineer. However, the ability to execute a breaking down and reassembling process has been dependent upon the availability of analytical tools. Not until the present decade were developments in high-speed digital computers sufficient to make possible the fundamental approach to problems of structural analysis known as the finite element method.

The finite element method is essentially the analysis of an idealized structural system composed of a finite number of elements interconnected at a finite number of joints or nodal points. It is the finite characteristic of structural connectivity which distinguishes the finite element problem from problems of continuum mechanics.

B. HISTORICAL BACKGROUND

The finite element method of analysis is applicable to one, two and three dimensional structures. Background discussion will be limited to the two dimensional case, although recently there has been considerable research activity on three dimensional structures.

Various shapes of finite elements have been employed in plane stress and plane strain analysis. Early research was focused on the development of flat rectangular and triangular finite elements. An abundance of papers, technical reports, and theses were produced; a complete bibliography would be beyond the scope of this paper. It is sufficient to say however, that requirements for efficient flat plate finite elements have been formulated and tested. Considerable success was reported in Refs. 1 and 2 using these elements to approximate shells of free form.

The development of curved finite elements, a considerably more difficult problem, has progressed at a much slower pace. Some early success was achieved in the development of truncated conical ring elements. However, problem formulation using these elements has not been generalized. Truncated conical elements have produced excellent results for particular problems but have also resulted in unsatisfactory analysis of similar problems.

A ring element with arbitrarily curved meridians developed by R. E. Jones and D. R. Strome [Ref. 3] produced acceptable results for the case of shells of revolution under pressure loads.

In 1968, G. Cantin [Ref. 4] successfully developed a curved finite element for cylindrical shells. This curved element was developed using 6 nodal coordinates per joint (u_i, v_i, w_i, $w_{i,\xi}$, $w_{i,\eta}$, $w_{i,\xi\eta}$) and implicitly contained a complete description of rigid body motions. This element was tested and found to produce excellent results for a broad class of cylindrical shell and flat plate problems for both membrane and bending behavior. (The flat element is obtained by letting $R \rightarrow \infty$.)

C. OBJECTIVE

The objective of the author's research is to develop the stiffness matrix of a four nodal point finite element for a shell of revolution which has arbitrarily curved meridians, hereafter to be called a curvilinear shell finite element.

II. THE FINITE ELEMENT METHOD

A. GENERAL

Finite element analysis of an elastic continuum logically separates into four phases: (1) structural idealization, (2) evaluation of the element stiffness properties, (3) the evaluation of the force-displacement behavior of the overall element assemblage, and (4) the evaluation of element stress and strain.

B. STRUCTURAL IDEALIZATION

Structural idealization is the division of the actual elastic continuum into an assemblage of geometrically compatible structural elements which are interconnected at a discrete number of nodal points, through which the forces are transmitted.

The discretization of the continuum is a physical process, subject only to conditions of geometric compatibility, rather than an approximation of a mathematical nature. The idealized structure, then, is constructed of elements which possess the material properties of the actual continuum.

A basic reduction is necessary to make the idealized structure mathematically tractable. This reduction, based on an a priori selection of possible deformations of the elastic continuum, constrains the elements to deform in only a certain number of predictable shapes. These constraints called displacement functions uniquely define displacements and the state of strain within the element in terms of nodal displacements. These functions are the keystone of a successful idealization.

C. EVALUATION OF ELEMENT STIFFNESS PROPERTIES

The evaluation of the stiffness properties of the element is the critical phase of the finite element method. As the evaluation of the stiffness properties of a curvilinear shell finite element is the object of the author's research, procedures will be discussed in following sections. For the present, it is sufficient to say that the element stiffness matrix relates the nodal forces and nodal displacements. This relation takes the form

$$\{F\} = [k]\{\delta\} \tag{2.1}$$

where

$\{F\}$ = nodal force vector

$\{\delta\}$ = nodal point displacement vector

$[k]$ = element stiffness matrix

D. EVALUATION OF THE FORCE-DISPLACEMENT BEHAVIOR OF THE IDEALIZED STRUCTURE

The force-displacement behavior of the overall element assemblage is obtained by the systematic superposition of individual element stiffness matrices and load vectors. This technique, called the "direct stiffness method," is well documented in the literature.

E. EVALUATION OF ELEMENT STRESS AND STRAIN

When nodal displacements are known, element strain is directly obtained using classical shell theory strain-displacement relationships. Element stress is then calculated using constitutive laws.

III. CONSTRUCTION OF THE STIFFNESS MATRIX

A. GENERAL

The stiffness matrix for a particular finite elment is the relationship between a set of generalized displacements and the corresponding generalized forces.

Standard matrix notation, square brackets denoting rectangular matrices, curly brackets representing column matrices, superscript (-1) denoting inversion, and superscript T denoting transposition, will be used.

B. GEOMETRIC DERIVATION

The geometric description of the curvilinear shell finite element must contain sufficient parameters to guarantee that an assemblage of elements will form a smooth surface whose slope is continuous and whose dimensions at element boundaries match those of the actual continuum. Furthermore, the geometric description must be sufficiently general to describe shells of revolution which have meridional inflection or reverse curvature as well as shells of degenerate geometry such as cylindrical shells and flat plates. An extension of the descriptive procedure used in Ref. 3 was found to satisfy the above requirements.

Geometrical symbols for the curvilinear shell finite element are defined in Figure 1. Nondimensional meridional arc length is denoted by

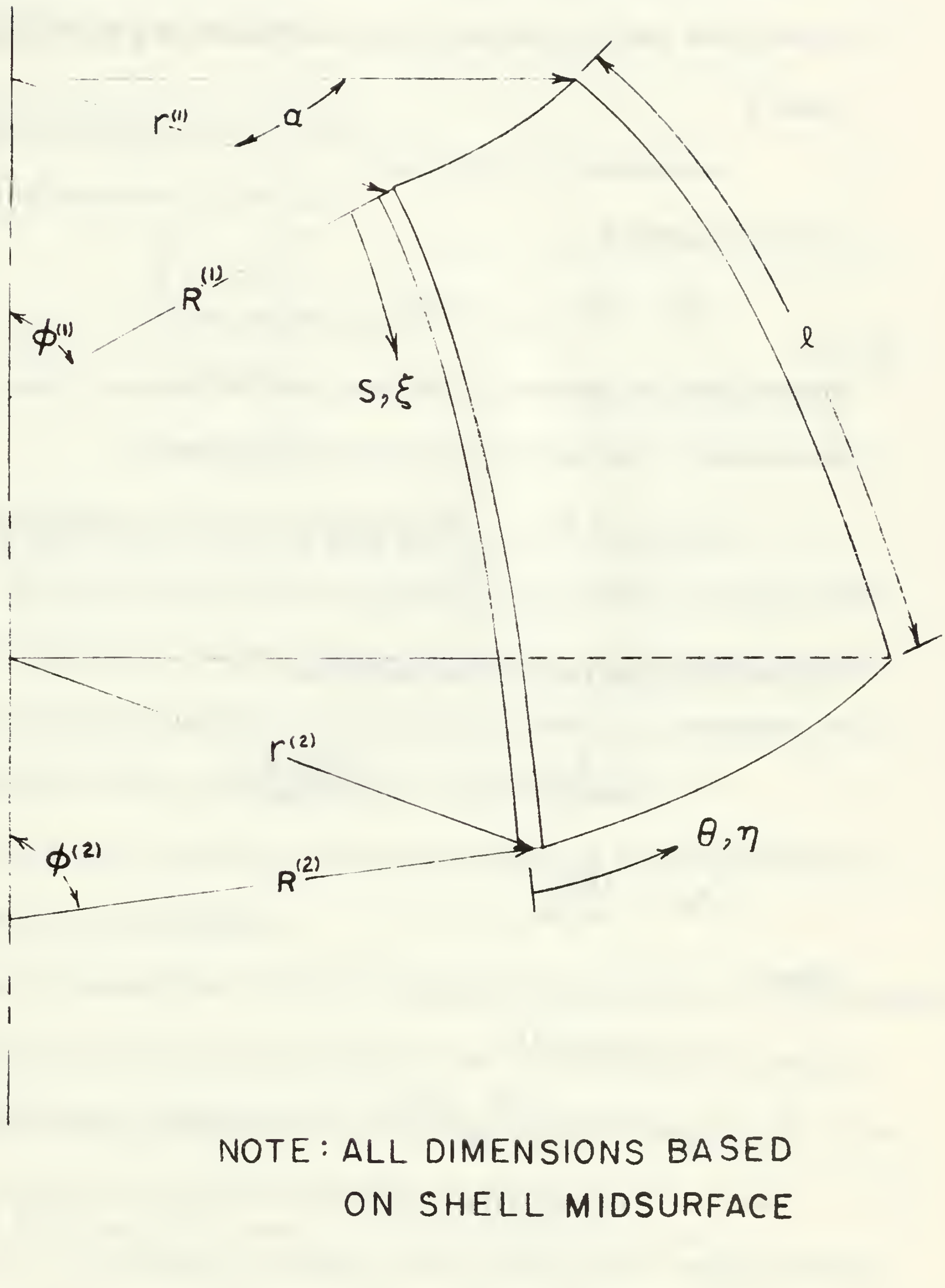


Fig. 1 DEFINITION OF GEOMETRICAL SYMBOLS

$$\xi = \frac{s}{\ell} \tag{3.1}$$

and nondimensional parallel arc length is denoted by

$$\eta = \frac{\theta}{\alpha} . \tag{3.2}$$

Through the use of a second order polynomial in ξ for the cosine of the angle ϕ,

$$\cos \phi = c_o + c_1\xi + c_2\xi^2 \tag{3.3}$$

and the values of

$r^{(1)}$, $r^{(2)}$, $\phi^{(1)}$, $\phi^{(2)}$, ℓ, α (see Figure 1)

expressions for completely defining element geometry were directly formulated. The radius of the element is given by

$$r(\xi) = r^{(1)} + \ell \int_0^{\xi} \cos \phi \, d\xi = r(0) + \ell \, (c_o \xi + \tfrac{1}{2} c_1 \xi^2 + \tfrac{1}{3} c_2 \xi^3) \tag{3.4}$$

and the principal radii of curvature are

$$R_1 = \frac{\ell}{d\phi/d\xi} = - \frac{\ell \sin \phi}{c_1 + 2c_2\xi} \tag{3.5}$$

$$R_2 = \frac{r(\xi)}{\sin \phi} \tag{3.6}$$

where

$$c_o = \cos \phi^{(1)} \tag{3.7}$$

$$c_1 = 2 \, [3 \left(\frac{r^{(2)} - r^{(1)}}{\ell} \right) - 2 \cos \phi^{(1)} - \cos \phi^{(2)}] \tag{3.8}$$

$$c_2 = 3 \, [-2 \left(\frac{r^{(2)} - r^{(1)}}{\ell} \right) + \cos \phi^{(1)} + \cos \phi^{(2)}] . \tag{3.9}$$

It is worthy of note that all values required for these expressions are found from engineering drawings or models.

For particular cases, such as conical shells and cylindrical shells where geometry is defined by analytical expressions, geometric approximation is not required.

C. DISPLACEMENT FUNCTIONS

The displacement functions

$$\{\tilde{u}\} = \begin{Bmatrix} u(\xi,\eta) \\ v(\xi,\eta) \\ w(\xi,\eta) \end{Bmatrix} \tag{3.10}$$

must satisfy three critical requirements.

1. They must insure that continuity of the structure can be automatically maintained across common boundaries of adjacent elements.

2. They must insure that all the lower modes of a complete set of straining modes will be present in the stiffness matrix.

3. They must contain an exact description of all possible rigid body motions of the element.

It can be shown [Ref. 5] that satisfaction of the first two requirements will guarantee that the strain energy of the "assemblage of elements" approximation will represent, for specified loads, a lower bound to the strain energy of the actual continuum. Satisfaction of the third requirement is necessary to expose hidden constraints opposing rigid body motion.

1. Application to the Curvilinear Shell Finite Element

The polynomial expansion:

$$u(\xi,\eta) = a_1\xi\eta + a_2\xi + a_3\eta + a_4 \qquad (3.11u)$$

$$v(\xi,\eta) = a_5\xi\eta + a_6\xi + a_7\eta + a_8 \qquad (3.11v)$$

$$\begin{aligned} w(\xi,\eta) = {} & a_9\xi^3\eta^3 + a_{10}\xi^3\eta^2 + a_{11}\xi^3\eta + a_{12}\xi^3 \\ & + a_{13}\xi^2\eta^3 + a_{14}\xi^2\eta^2 + a_{15}\xi^2\eta + a_{16}\xi^2 \\ & + a_{17}\xi\eta^3 + a_{18}\xi\eta^2 + a_{19}\xi\eta + a_{20}\xi \\ & + a_{21}\eta^3 + a_{22}\eta^2 + a_{23}\eta + a_{24} \end{aligned} \qquad (3.11w)$$

or more compactly

$$\{\tilde{u}\} = [P] \; \{a_i\} \qquad (3.12)$$

satisfies the requirements of interelement compatibility and completeness of strain field. Requirement three, that an exact description of all possible rigid body motions of the element be represented, is not satisfied, however, as will be shown in the following section.

D. RIGID BODY MODES

It has been established in Ref. 6, that conditions of equilibrium are not satisfied unless the displacement functions contain an exact description of rigid body motions of the element. Moreover, it has been established in Ref. 7 that inclusion of rigid body motion is essential and cannot be represented by independent displacement components. Rigid body motion, then, must be introduced without compromise to interelement compatibility requirements.

A general method for including rigid body motion is developed in Ref. 6.

1. Application to the Curvilinear Shell Finite Element

Consider a point P, specified by position vector ρ, on the element in the system of reference shown in Figure 2. If the element is submitted to general translation then,

$$\bar{\delta}_T = \delta_1 \bar{i} + \delta_2 \bar{j} + \delta_3 \bar{k} \tag{3.13}$$

in the system of reference (x, y, z). The corresponding scalar components of the displacement field are:

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \begin{bmatrix} \cos\phi \ \cos\theta & \cos\phi \ \sin\theta & -\sin\phi \\ -\sin\theta & \cos\theta & 0 \\ \sin\phi \ \cos\theta & \sin\phi \ \sin\theta & \cos\phi \end{bmatrix} \begin{Bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \end{Bmatrix} \tag{3.14}$$

or more compactly

$$\{u\} = [R_{TRANS}] \{\delta_i\}$$

When submitted to rigid body rotation of small amplitude

$$\bar{\beta} = \beta_1 \bar{i} + \beta_2 \bar{j} + \beta_3 \bar{k} \tag{3.15}$$

the displacement vector field of the element in reference system (x, y, z) is

$$\bar{\delta}_R = \bar{\beta} \times \bar{\rho} \tag{3.16}$$

The corresponding scalar expression is

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \begin{bmatrix} -\sin\theta\,(z\cos\phi + r\sin\phi) & \cos\theta\,(z\cos\phi + r\sin\phi) & 0 \\ -z\cos\theta & -z\sin\theta & r \\ -\sin\theta\,(z\sin\phi - r\cos\phi) & \cos\theta\,(z\sin\phi - r\cos\phi) & 0 \end{bmatrix} \begin{Bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{Bmatrix} \tag{3.17}$$

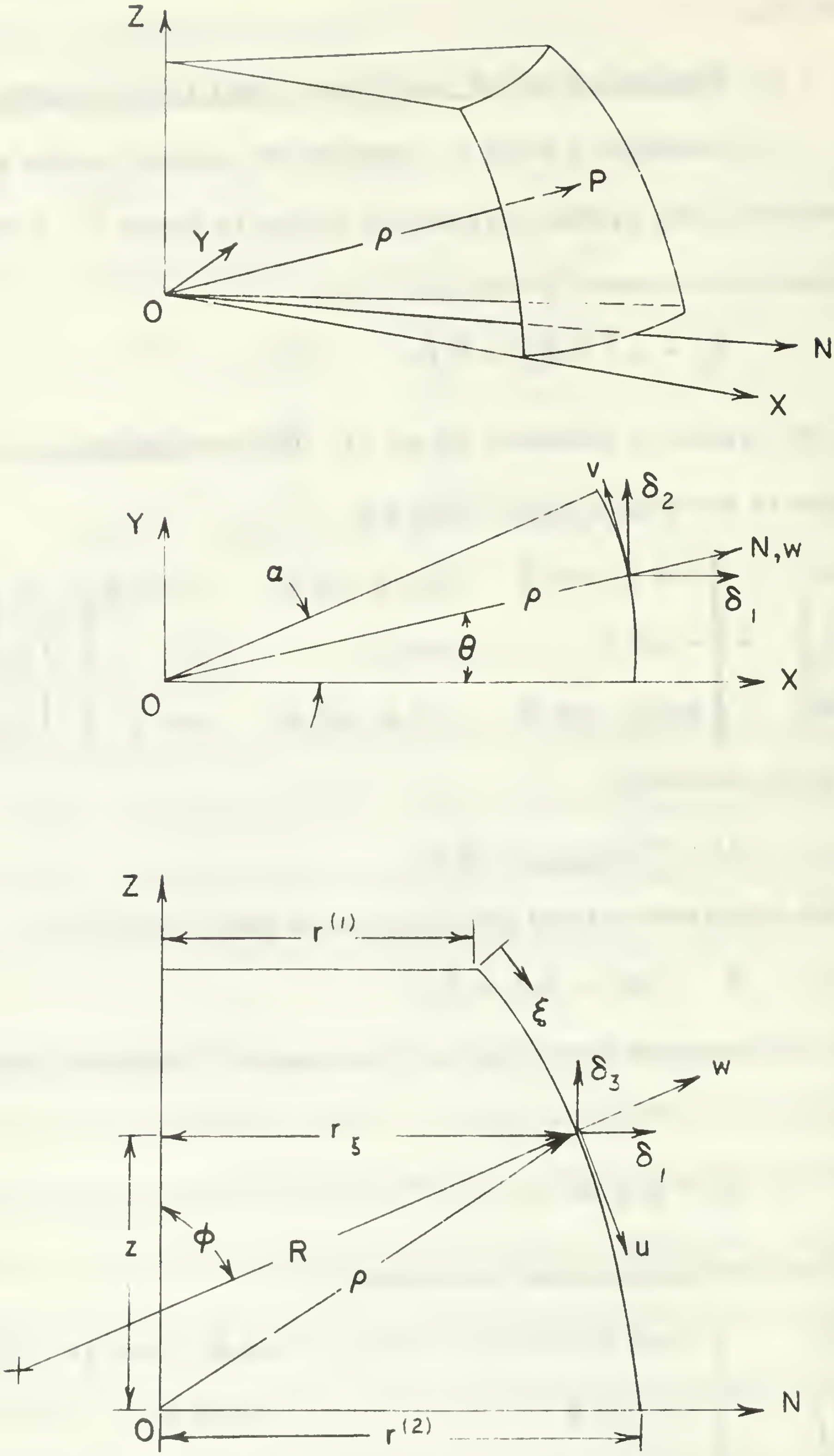


Fig 2 DEFINITION OF COORDINATE SYSTEMS

or more compactly

$$\{u\} = [R_{ROT}] \{\beta_i\}$$

Combining expressions (3.14) and (3.17) gives:

$$\begin{Bmatrix} u \\ v \\ w \end{Bmatrix} = \left[\begin{array}{c:c} R_{TRANS} & R_{ROT} \end{array} \right] \begin{Bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \hdashline \beta_1 \\ \beta_2 \\ \beta_3 \end{Bmatrix} \tag{3.18}$$

or

$$\{u_i\} = [R] \{u_j^R\} \tag{3.19}$$

Matrix [R] is given explicitly in Table 1.

E. CLASSICAL SHELL THEORY

In the introduction it was emphasized that the difference between finite element problems and problems in continuum mechanics is based on a physical approximation only. Classical theories for shells from continuum mechanics are, therefore, applicable in the finite element approach.

The consitutive law for an isotropic, homogenous, thin elastic shell can be written:

$$\begin{Bmatrix} N_\xi \\ N_\eta \\ N_{\xi\eta} \\ M_\xi \\ M_\eta \\ M_{\xi\eta} \end{Bmatrix} = \begin{bmatrix} K_1 & K_2 & 0 & 0 & 0 & 0 \\ K_2 & K_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & K_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & D_1 & D_2 & 0 \\ 0 & 0 & 0 & D_2 & D_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & D_3 \end{bmatrix} \begin{Bmatrix} \epsilon_\xi \\ \epsilon_\eta \\ \epsilon_{\xi\eta} \\ \varkappa_\xi \\ \varkappa_\eta \\ \varkappa_{\xi\eta} \end{Bmatrix} \tag{3.20}$$

where

$$K_1 = \frac{Et}{1-\nu^2} \; ; \; K_2 = \nu K_1 \; ; \; K_3 = \tfrac{1}{2}(1-\nu) K_1$$

$$D_1 = \frac{Et^3}{12(1-\nu^2)} ; D_2 = \nu D_1 \; ; \; D_3 = \tfrac{1}{2}(1-\nu) D_1$$

or more compactly

$$\{N\} = [E] \{\epsilon\} \tag{3.21}$$

Young's modulus E and Poisson's ratio ν, according to definition are constants. The thickness t, however, is allowed to vary in the meridional direction. In this thesis, shells of uniform thickness and shells of linearly varying thickness are considered.[1]

Numerous theories exist for curvilinear shells. The venerable theories have been subject to years of critical inspection; their merits and limitations are well known. Several theories investigated would have been acceptable. Kraus' formulation [Ref. 8] was selected for the following reasons:

1. The theory is consistent. That is, the theory holds for the limiting cases of cylindrical shell and flat plate elements. This quality is of basic importance since mesh size reduction to the extent where the element is nearly flat is necessary in many finite element problems.

2. The theory is presented in a form that can be directly applied to this study.

3. The theory allows strain free modes for rigid body motions.

---

[1] Any thickness which varies as a function of ξ could be considered by changing only one card in the computer program.

The strain displacement relations are:

$$\begin{Bmatrix} \epsilon_\xi \\ \epsilon_\eta \\ \epsilon_\xi \\ \varkappa_\xi \\ \varkappa_\eta \\ \varkappa_{\xi\eta} \end{Bmatrix} = \begin{bmatrix} \frac{1}{\ell}\frac{\partial}{\partial\xi} & 0 & \frac{1}{R_1} \\ \frac{\cos\phi}{r} & \frac{1}{\alpha r}\frac{\partial}{\partial\eta} & \frac{\sin\phi}{r} \\ \frac{1}{\alpha r}\frac{\partial}{\partial\eta} & \frac{1}{\ell}\frac{\partial}{\partial\xi} - \frac{1}{r\ell}\frac{\partial r}{\partial\xi} & 0 \\ \frac{1}{\ell R_1}\frac{\partial}{\partial\xi} - \frac{1}{\ell R_1^2}\frac{\partial R_1}{\partial\xi} & 0 & -\frac{1}{\ell^2}\frac{\partial^2}{\partial\xi^2} \\ \frac{\cos\phi}{r R_1} & \frac{1}{\alpha r^2}\frac{\partial}{\partial\eta} & -\frac{1}{\alpha^2 r^2}\frac{\partial^2}{\partial\eta^2} - \frac{\cos\phi}{r\ell}\frac{\partial}{\partial\xi} \\ \frac{1}{\alpha r R_1}\frac{\partial}{\partial\eta} & \frac{1}{\ell r}\frac{\partial}{\partial\xi} - \frac{2}{\ell r^2}\frac{\partial r}{\partial\xi} & \frac{2}{r^2\alpha\ell}\frac{\partial r}{\partial\xi}\frac{\partial}{\partial\eta} - \frac{2}{r\alpha\ell}\frac{\partial^2}{\partial\xi\partial\eta} \end{bmatrix} \begin{Bmatrix} u \\ v \\ w \end{Bmatrix} \tag{3.22}$$

or more compactly

$$\{\epsilon\} = [D]\,\{\tilde{u}\} \tag{3.23}$$

All classical theories investigated by the author contained inherent singularities. Kraus' formulation was no exception. It can be seen that several terms in equation (3.22) become infinite whenever r is equal to zero. This theoretical inadequacy has been avoided by specifying either a small hole or a small rigid insert with appropriate boundary conditions, at the pole point. The problem solved in Chapter 5, Section B, 2 demonstrates this method.

F. NODAL COORDINATES

For computational convenience, physical nodal coordinates are transformed to a generalized coordinate system. This transformation facilitates the expression of displacements, elastic characteristics, and strain energy in a system free from geometric complications. The following set of nodal coordinates was selected:

$$\{u_i\} = \langle u_i, v_i, w_i, w_{i,s}, w_{i,\theta}, w_{i,s\theta} \rangle^T \quad i = 1,2,3,4 \tag{3.24}$$

where s and θ are the physical coordinates of any point on the element. Then using (3.1), (3.2), and (3.12) the transformation matrix [T] is constructed such that

$$\{u_i\} = [T]\{a_i\}. \tag{3.25}$$

The transformation matrix [T] is nonsingular and is easily inverted.

Matrix [T] is given explicitly in Table 2.

G. STIFFNESS MATRIX

The strain energy in the element is:

$$U_S = \tfrac{1}{2}\iint \{N\}^T \{\epsilon\}\, r\, \alpha\, \ell\, d\xi\, d\eta \tag{3.26}$$

Substituting for stress {N} and strain {ϵ} one gets:

$$U_S = \tfrac{1}{2}\{u_i\}^T [T^{-1}]^T \left(\iint [P^*]^T [E][P^*]\, r\, \alpha\, \ell\, \partial\xi\, \partial\eta\right)[T^{-1}]\{u_i\} \tag{3.27}$$

where

$$[P^*] = [D][P] \tag{3.28}$$

The stiffness matrix is then:

$$[K] = [T^{-1}]^T \left(\iint [P^*][E][P^*] r \alpha \ell \, \partial\xi \, \partial\eta \, [T^{-1}] \right.$$

or

$$[K] = [T^{-1}]^T \, [k] \, [T^{-1}] \tag{3.29}$$

where $[T]$ and $[k]$ are both (24x24) matrices.

Stiffness matrix $[K]$ may contain some rigid body modes depending on problem type. However, by subjecting $[K]$ to eigenvalue analysis, individual rigid body modes, if present, can be identified.

When $[K]$ does not contain all six components of rigid body motion, it must be modified accordingly. The method developed in Ref. 6 is applied as shown below:

$$\{u_i\} = \left[I \;\vdots\; R \right] \left\{ \begin{array}{c} u_i \\ \hline u_j^R \end{array} \right\} \tag{3.30}$$

where $\left\{ u_j^R \right\} = \langle \delta_x, \delta_y, \delta_z, \beta_x, \beta_y, \beta_z \rangle^T$, $[I]$ is a (24x24) identity matrix, and $[R]$ is the (24x6) matrix developed in Section D.

Matrix $[K]$ is then expanded to include all six rigid body modes:

$$\left[\begin{array}{c} I \\ \hline [R]^T \end{array} \right] \left[K \right] \left[I \;\vdots\; R \right] = \left[\begin{array}{c|c} [K] & [K][R] \\ \hline [R]^T[K] & [R]^T[K][R] \end{array} \right] \tag{3.31}$$

where $\left[[R]^T [K] [R] \right]$ is a (6x6) matrix.

$$\left[\begin{array}{c} F \\ \hline 0 \end{array} \right] = \left[\begin{array}{c|c} [K] & [K][R] \\ \hline [R]^T[K] & [R]^T[K][R] \end{array} \right] \left[\begin{array}{c} u_i \\ \hline u_j^R \end{array} \right] \tag{3.32}$$

Solving for

$$\{u_j^R\} = \left[[R]^T[K][R]\right]^{-1}\left[[K][R]\right]\{u_i\} \tag{3.33}$$

However, if the stiffness matrix [K] already contains some of the rigid body modes, then matrix $\left[[R]^T[K][R]\right]$ will be singular. These modes have been previously identified and are merely suppressed in $\{u_j^R\}$ and in the corresponding column of [R]. Matrix $\left[[R]^T[K][R]\right]$ then becomes nonsingular. Solving (3.33) and substituting into the first (3.32) leads to the following expression:

$$\{F\} = \left([K] - \left[[K][R]\right]\left[[R]^T[K][R]\right]^{-1}\left[[R]^T[K]\right]\right)\{u_i\} \tag{3.34}$$

The stiffness matrix modified to include rigid body modes is then:

$$[K^*] = [K] - \left[[K][R]\right]\left[[R]^T[K][R]\right]^{-1}\left[[R]^T[K]\right]$$

Stiffness matrix [K*] is strain free for any rigid body motion.

H. DEFINITION OF MATRICES

In the following sections of this thesis, the stiffness matrix [K*] derived above is referred to as CSFE. The matrix $\left[[R]^T[K][R]\right]$ developed in (3.31) is hereafter referred to as K22. For the general case where all six rigid body modes are absent, K22 is a (6x6) matrix. In the case of a cylindrical shell segment where four rigid body modes are absent, K22 is (4x4), and in the case of a conical segment where five rigid modes are absent, K22 is (5x5).

I. LOAD VECTORS

Loads and moments acting on the element surface must be represented by vectors which are consistent with displacements and rotations in the generalized coordinate system.

A consistent load vector then, is defined as a vector made up of fictitious quantities which would give, after an inner product with the nodal point displacement, the same work as the real load.

1. Concentrated Load Vectors

For concentrated loads of F_ξ, F_η, F_ζ acting at a point (s, θ) of the element, the work done is

$$W = F_\xi u + F_\eta v + F_\zeta w$$

or more compactly

$$W = \left\{F_i\right\}^T \left\{\tilde{u}\right\}$$

Substituting for $\{\tilde{u}\}$ from (3.12) and (3.24) gives,

$$W = \left\{F_i\right\}^T \left[P\right] \left[T^{-1}\right] \left\{u_i\right\}$$

The consistent concentrated load vector is then:

$$\left\{c_{LOAD}\right\} = \left[P\right]^T \left\{F_i\right\} \tag{3.35}$$

2. Pressure Load Vectors

The work done by a pressure load p is:

$$W = \int_A p\, w\, dA \tag{3.36}$$

then if

$$\{q\} = \langle\, 0 \quad 0 \quad p \,\rangle^T$$

direct substitution of known quantities into (3.36) gives:

$$W = \iint \{q\}^T \{\tilde{u}\} \, \ell \alpha r \, d\xi d\eta = \left(\ell \alpha \iint \{q\}^T \left[P \right] \left[T^{-1} \right] r \, d\xi d\eta \right) \left\{ u_1 \right\}$$

The consistent pressure load vector is then:

$$\left\{ P_{LOAD} \right\} = \ell \alpha \iint \left[P \right]^T \{q\} \, r \, d\xi d\eta . \tag{3.37}$$

Table 1. Transformation Matrix [R] Between Nodal Coordinates and Polynomial Coefficients

$R_{1,1}$ *	0	$R_{1,3}$	0	$R_{1,5}$	0
0	1	0	$R_{2,4}$	0	$R_{2,6}$
$R_{3,1}$	0	$R_{3,3}$	0	$R_{3,5}$	0
$R_{4,1}$	0	$R_{4,3}$	0	$R_{4,5}$	0
0	$R_{5,2}$	0	$R_{5,4}$	0	0
0	$R_{6,2}$	0	$R_{6,4}$	0	0
$R_{7,1}$	0	$R_{7,3}$	0	$R_{7,5}$	0
0	1	0	0	0	$R_{8,6}$
$R_{9,1}$	0	$R_{9,3}$	0	$R_{9,5}$	0
$R_{10,1}$	0	$R_{10,3}$	0	$R_{10,5}$	0
0	$R_{11,2}$	0	$R_{11,4}$	0	0
0	$R_{12,2}$	0	$R_{12,4}$	0	0
$R_{13,1}$	$R_{13,2}$	$R_{13,3}$	$R_{13,4}$	$R_{13,5}$	0
$R_{14,1}$	$R_{14,2}$	0	0	0	$R_{14,6}$
$R_{15,1}$	$R_{15,2}$	$R_{15,3}$	$R_{15,4}$	$R_{15,5}$	0
$R_{16,1}$	$R_{16,2}$	$R_{16,3}$	$R_{16,4}$	$R_{16,5}$	0
$R_{17,1}$	$R_{17,2}$	0	$R_{17,4}$	$R_{17,5}$	0
$R_{18,1}$	$R_{18,2}$	0	$R_{18,4}$	$R_{18,5}$	0
$R_{19,1}$	$R_{19,2}$	$R_{19,3}$	$R_{19,4}$	$R_{19,5}$	0
$R_{20,1}$	$R_{20,2}$	0	$R_{20,4}$	$R_{20,5}$	$R_{20,6}$
$R_{21,1}$	$R_{21,2}$	$R_{21,3}$	$R_{21,4}$	$R_{21,5}$	0
$R_{22,1}$	$R_{22,2}$	$R_{22,3}$	$R_{22,4}$	$R_{22,5}$	0
$R_{23,1}$	$R_{23,2}$	0	$R_{23,4}$	$R_{23,5}$	0
$R_{24,1}$	$R_{24,2}$	0	$R_{24,4}$	$R_{24,5}$	0

* Commas in this table are used as separators, no partial differentiation is implied.

where:

$$R_{1,1} = \cos \phi^{(1)}$$

$$R_{1,3} = -\sin \phi^{(1)}$$

$$R_{1,5} = R_{1,1}\, z - R_{1,3}\, r^{(1)}$$

$$R_{2,4} = -z$$

$$R_{2,6} = r^{(1)}$$

$$R_{3,1} = \sin \phi^{(1)}$$

$$R_{3,3} = \cos \phi^{(1)}$$

$$R_{3,5} = R_{3,1}\, z - R_{3,3}\, r^{(1)}$$

$$R_{4,1} = \frac{R_{1,1}\, c_1}{R_{3,1}\, \ell}$$

$$R_{4,3} = -\frac{c_1}{\ell}$$

$$R_{4,5} = R_{3,1}^{\;2} + R_{4,1}\, z + R_{1,1}\, c_o + \frac{c_1\, r^{(1)}}{\ell}$$

$$R_{5,2} = \frac{R_{3,1}}{r^{(1)}}$$

$$R_{5,4} = -\frac{R_{3,5}}{r^{(1)}}$$

$$R_{6,2} = -\frac{R_{4,1}}{r^{(1)}}$$

$$R_{6,4} = \frac{R_{4,5}}{r^{(1)}}$$

$$R_{7,1} = \cos \phi^{(2)}$$

$$R_{7,3} = -\sin \phi^{(2)}$$

$$R_{7,5} = -R_{7,3}\, r^{(2)}$$

$$R_{8,6} = r^{(2)}$$

$$R_{9,1} = \sin \phi^{(2)}$$

$$R_{9,3} = \cos \phi^{(2)}$$

$$R_{9,5} = - R_{9,3}\, r^{(2)}$$

$$R_{10,1} = \frac{R_{7,1}(c_1 + 2c_2)}{R_{9,1}\ell}$$

$$R_{10,3} = -\frac{(c_1 + 2c_2)}{\ell}$$

$$R_{10,5} = R_{9,1}^{\,2} + R_{9,3}(c_o + c_1 + c_2) + \frac{(c_1 + 2c_2)r^{(2)}}{\ell}$$

$$R_{11,2} = \frac{R_{9,1}}{r^{(2)}}$$

$$R_{11,4} = \cos \phi^{(2)}$$

$$R_{12,2} = -\frac{R_{10,1}}{r^{(2)}}$$

$$R_{12,4} = -\frac{R_{10,5}}{r^{(2)}}$$

$$R_{13,1} = \cos \phi^{(2)} \cos \alpha$$

$$R_{13,2} = \cos \phi^{(2)} \sin \alpha$$

$$R_{13,3} = - \sin \phi^{(2)}$$

$$R_{13,4} = - \sin \phi^{(2)} \sin \alpha\, r^{(2)}$$

$$R_{13,5} = \sin \phi^{(2)} \cos \alpha\, r^{(2)}$$

$$R_{14,1} = - \sin \alpha$$

$$R_{14,2} = \cos \alpha$$

$$R_{14,6} = r^{(2)}$$

$$R_{15,1} = \sin \phi^{(2)} \cos \alpha$$

$$R_{15,2} = \sin \phi^{(2)} \sin \alpha$$

$$R_{15,3} = \cos \phi^{(2)}$$

$$R_{15,4} = \cos \phi^{(2)} \sin \alpha \, r^{(2)}$$

$$R_{15,5} = - \cos \phi^{(2)} \cos \alpha \, r^{(2)}$$

$$R_{16,1} = R_{10,1} R_{14,2}$$

$$R_{16,2} = - R_{10,1} R_{14,1}$$

$$R_{16,3} = R_{10,3}$$

$$R_{16,4} = R_{14,1} R_{10,5}$$

$$R_{16,5} = + R_{14,2} R_{10,5}$$

$$R_{17,1} = R_{9,1} \frac{R_{14,1}}{r^{(2)}}$$

$$R_{17,2} = \frac{R_{15,1}}{r^{(2)}}$$

$$R_{17,4} = R_{13,1}$$

$$R_{17,5} = R_{13,2}$$

$$R_{18,1} = -R_{10,1} \frac{R_{14,1}}{r^{(2)}}$$

$$R_{18,2} = - \frac{R_{16,1}}{r^{(2)}}$$

$$R_{18,4} = R_{14,2} R_{12,4}$$

$$R_{18,5} = - R_{14,1} R_{12,4}$$

$$R_{19,1} = R_{1,1} R_{14,2}$$

$$R_{19,2} = - R_{3,3} R_{14,1}$$

$$R_{19,3} = R_{1,3}$$

$$R_{19,4} = R_{14,1} R_{1,5}$$

$$R_{19,5} = R_{14,2} R_{1,5}$$

$$R_{20,1} = R_{14,1}$$

$$R_{20,2} = R_{14,2}$$

$$R_{20,4} = R_{2,4} R_{14,2}$$

$$R_{20,5} = - R_{2,4} R_{14,1}$$

$$R_{20,6} = R_{2,6}$$

$$R_{21,1} = R_{3,1} R_{14,2}$$

$$R_{21,2} = - R_{3,1} R_{14,1}$$

$$R_{21,3} = R_{1,1}$$

$$R_{21,4} = R_{14,1} R_{3,5}$$

$$R_{21,5} = R_{14,2} R_{3,5}$$

$$R_{22,1} = R_{4,1} R_{14,2}$$

$$R_{22,2} = - R_{4,1} R_{14,1}$$

$$R_{22,3} = R_{4,3}$$

$$R_{22,4} = R_{14,1} R_{4,5}$$

$$R_{22,5} = R_{14,2} R_{4,5}$$

$$R_{23,1} = R_{5,2} R_{14,1}$$

$$R_{23,2} = R_{5,2} R_{14,2}$$

$$R_{23,4} = R_{14,2} R_{5,4}$$

$$R_{23,5} = - R_{14,1} R_{5,4}$$

$$R_{24,1} = - R_{4,1} \frac{R_{14,1}}{r^{(1)}}$$

$$R_{24,2} = - \frac{R_{22,1}}{r^{(1)}}$$

$$R_{24,4} = R_{14,2} R_{6,4}$$

$$R_{24,5} = - R_{14,1} R_{6,4}$$

Table 2: Transformation Matrix [T] Between Nodal Coordinates and Polynomial Coefficients (Columns 1 - 12)

0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	$3T_2$
0	0	0	0	0	0	0	0	0	0	T_4	0
0	0	0	0	0	0	0	0	0	0	$3T_6$	0
1	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0	$3T_2$	$3T_2$	$3T_2$	$3T_2$
0	0	0	0	0	0	0	0	$3T_4$	$2T_4$	T_4	0
0	0	0	0	0	0	0	0	$9T_6$	$6T_6$	$3T_6$	0
0	0	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0

Table 2: (Continued) (Columns 13 - 24)

0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	T_1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	T_3	0
0	0	0	0	0	0	T_5	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0	0	0	1
0	0	0	$2T_2$	0	0	0	T_2	0	0	0	0
0	0	T_4	0	0	0	T_4	0	0	0	T_4	0
0	0	$2T_6$	0	0	0	T_6	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1
$2T_2$	$2T_2$	$2T_2$	$2T_2$	T_2	T_2	T_2	T_2	0	0	0	0
$3T_4$	$2T_4$	T_4	0	$3T_4$	$2T_4$	T_4	0	$3T_4$	$2T_4$	T_4	0
$6T_6$	$4T_6$	$2T_6$	0	$3T_6$	$2T_6$	T_6	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1
0	0	0	0	T_1	T_1	T_1	T_1	0	0	0	0
0	0	0	0	0	0	0	0	$3T_3$	$2T_3$	T_3	0
0	0	0	0	$3T_5$	$2T_5$	T_5	0	0	0	0	0

where:

$$T_1 = -\frac{1}{\ell r^{(1)}} \qquad T_2 = -\frac{1}{\ell r^{(2)}}$$

$$T_3 = \frac{1}{\alpha r^{(1)}} \qquad T_4 = \frac{1}{\alpha r^{(2)}}$$

$$T_5 = \frac{1}{\alpha \ell r^{(1)}} \qquad T_6 = \frac{1}{\alpha \ell r^{(2)}}$$

IV. NUMERICAL CONSIDERATIONS AND PROGRAM INFORMATION

A. GENERAL

All numerical computations were accomplished on the IBM 360/67 digital computer using FORTRAN 4 language. The entire program was coded in double precision arithmetic. An overlay scheme under control of the linkage editor was used to conserve core storage.

B. QUADRATURE

It was evident that the integrands of (3.27) and (3.37) could not be manipulated into closed form. Since the above expressions were in nondimensional form, lower and upper limits of integration zero and one respectively, it was advantageous to use Gaussian quadrature. An additional advantage of using Gaussian forms is that a polynomial of degree 2n-1 can be exactly described by n points [9]. For polynomials of unknown degree, maximum accuracy is achieved by selecting a large number of points. Additional accuracy, however, is obtained at the expense of computation time. Investigation by the author showed that Gauss Legendre—8 point quadrature provided accuracy within the range of double precision round off error for shells whose geometry could be described by analytic expressions. For shells of more complicated geometry, required accuracy is obtained when standard finite element mesh size reduction is accomplished.

C. LIMITING CASES

In this thesis, cylindrical shells, truncated cones, and flat plates have been considered as special cases. When a shell is specified as one of the above forms, geometric approximation is precluded.

Exact solutions for these shells provided a powerful tool for the analysis of the stiffness properties of the general case of the curvilinear shell finite element.

D. CASE WHERE ϕ IS EQUAL TO ZERO

Singularities exist in the geometric approximation whenever the element is constructed in such a way that a point on its surface is locates at ϕ equal to zero (see Figure 1). Discussion of the effects of this singularity is contained in Chapter 5, Section B, 2.

E. INITIAL CONSTRUCTION OF THE STIFFNESS MATRIX

The author originally constructed the stiffness matrix by constructing all matrices in (3.27) and then integrating the entire expression. Accurate results were achieved. However, the construction of the stiffness matrix for a single element required 500,000 bytes of computer storage and approximately 20 minutes of computation time. This was obviously unsatisfactory for other than pedagogical application.

Fortunately, identical results were obtained by individual stiffness matrix element computation and by taking advantage of symmetry. This was done in the remarkably short time of 15 seconds using only 143,000 bytes of computer storage.

CSFE was constructed and evaluated in a separate program prior to being incorporated in a direct stiffness program.

F. DIRECTION STIFFNESS PROGRAM

Numerical solution of engineering problems was accomplished through the use of a stress analysis program which was based on the direct stiffness method. The total stiffness matrix and the load vectors were assembled using programming techniques furnished by Professor G. Cantin. This program gave numerical values for displacements, moments, stresses, and strains at each nodal point of the mesh. A flow diagram of the computer program is presented in Figure 3. It can be seen that the program is functionally divided into two parts. In the CSFE portion, the stiffness matrix and its associated pressure load, strain-displacement, and stress-strain relationships are computed for individual elements. Results of these computations are stored on disk units. In the stress analysis portion, individual element information is retrieved from disk and assembled in a manner consistent with the direct stiffness method.

The author's primary purpose for construction of the stress analysis portion of the program was to verify the stiffness properties of the curvilinear shell finite element. It was therefore, designed to handle problems of limited size. Specifically, the program is limited to a maximum of 36 elements, 49 nodal points, or 294 equations. Coding methods used, however, are general to allow for future program size expansion. The computer program is found in Appendix C.

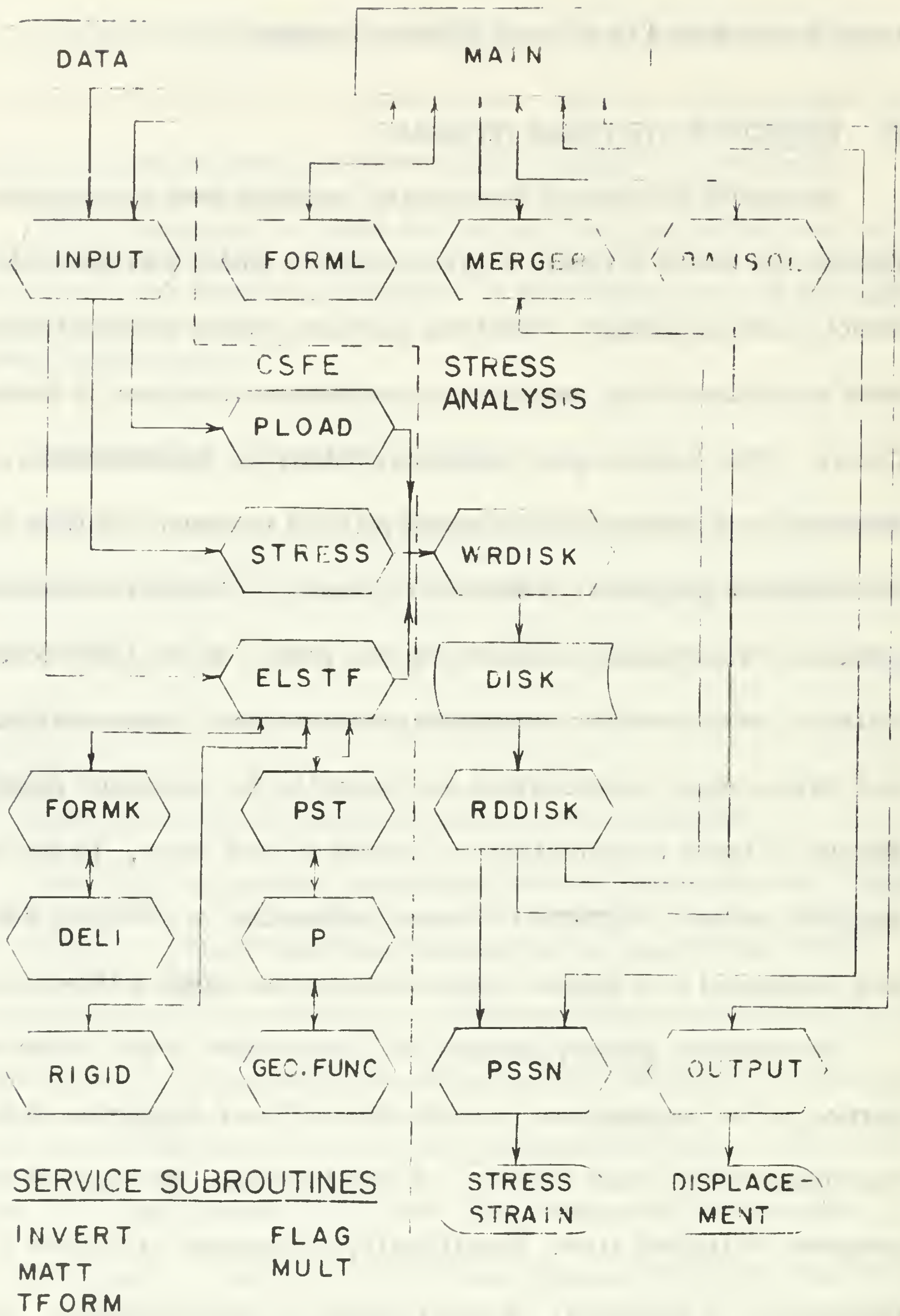


Fig. 3 FUNCTIONAL FLOW DIAGRAM

V. DISCUSSION

A. INTEGRITY OF THE CURVILINEAR SHELL FINITE ELEMENT

Confidence in the integrity of CSFE was established by comparing stiffness properties of limiting cases (flat plate and cylindrical shell) with properties found in the literature. Element by element comparison with values given by Bogner, et. al. [Ref. 10] revealed exact agreement for the case of the flate plate. In the case of the cylindrical shell element similar comparison was made with values given by Cantin [Ref. 1]; once again exact agreement existed.

Confidence in the integrity of the stress analysis program was established by direct comparison of problem solutions using CSFE with solutions found in the literature. Results of comparison are discussed in the following sections.

B. NUMERICAL SOLUTION OF CLASSICAL PROBLEMS

The classical problems analyzed in this section were selected to demonstrate the integrity of the element, and to exhibit both the flexibility and limitations associated with CSFE.

1. Pinched Cylinder

The pinched cylinder illustrated in Figure 4 has been analyzed by Bogner, et al [Ref. 11] using a (48x48) element and by Cantin [Ref. 6] using a (24x24) element. Table 3 is a comparison of results obtained using CSFE with those found in the literature. The obvious agreement of results confirms the integrity of CSFE. Moreover, it is

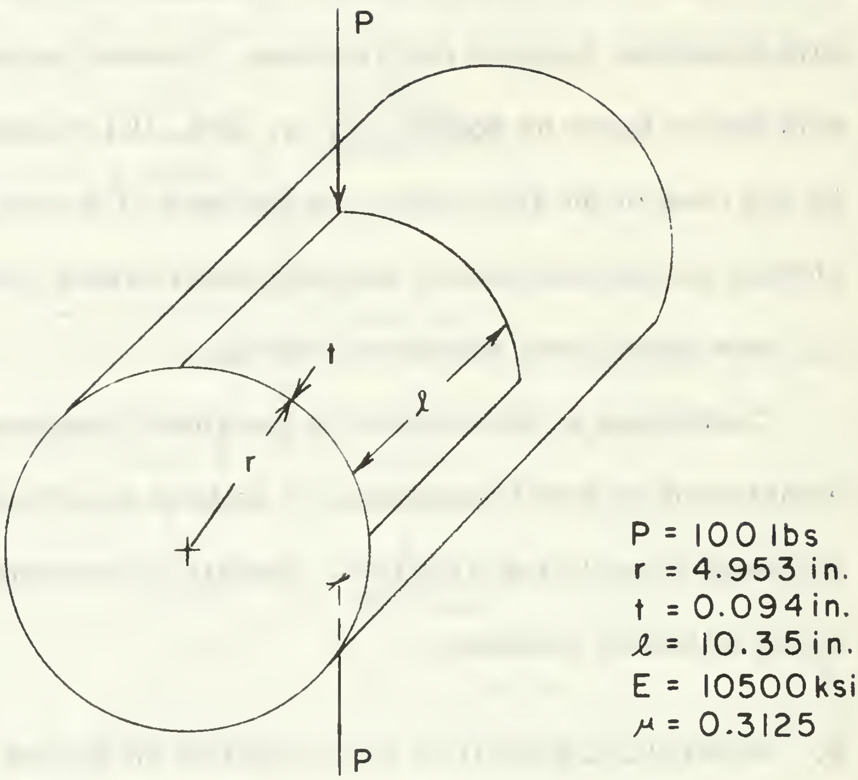


Fig. 4 PINCHED CYLINDER

evident that CSFE has the desirable characteristic of rapid convergence to the true solution (-0.1139) given by Cantin.

No. of EQS.	Bogner et al (48 x 48)		Cantin (28 x 28)		CSFE (24 x 24)	
	Mesh	Displace. (in.)	Mesh	Displace.	Mesh	Displace.
54			2x2	-0.0931	2x2	-0.0932
108	2x2	-0.0808				
150			4x4	-0.1126	4x4	- .1128
180	2x4	-0.1098				
294			6x6	-0.1137	6x6	-0.1138
486			8x8	-0.1139		
726			10x10	-0.1139		

Table 3. Displacement under load P for pinched cylinder

2. Spherical Cap

The spherical cap illustrated in Figure 5 demonstrates the general case of CSFE. This problem which has an exact solution given by Timoshenko [Ref. 12] is useful for demonstrating the capabilities and limitations of the new element. Figure 6 contains a plot of meridional bending moments.[2]

[2] The signs of Timoshenko's results have been changed to agree with the author's convention.

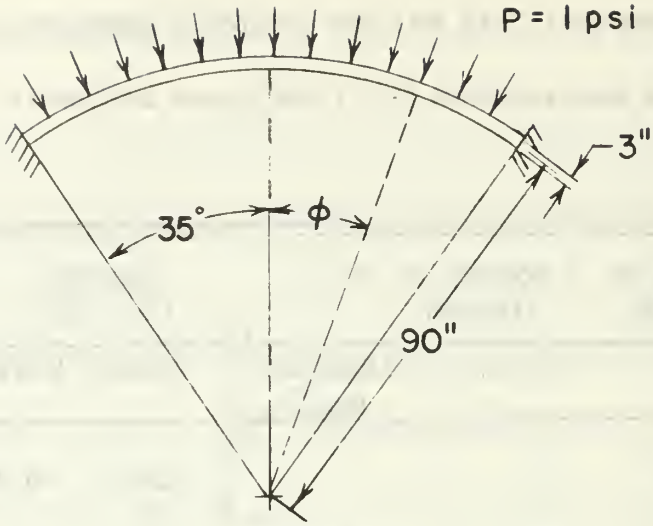


Fig. 5 SPHERICAL CAP

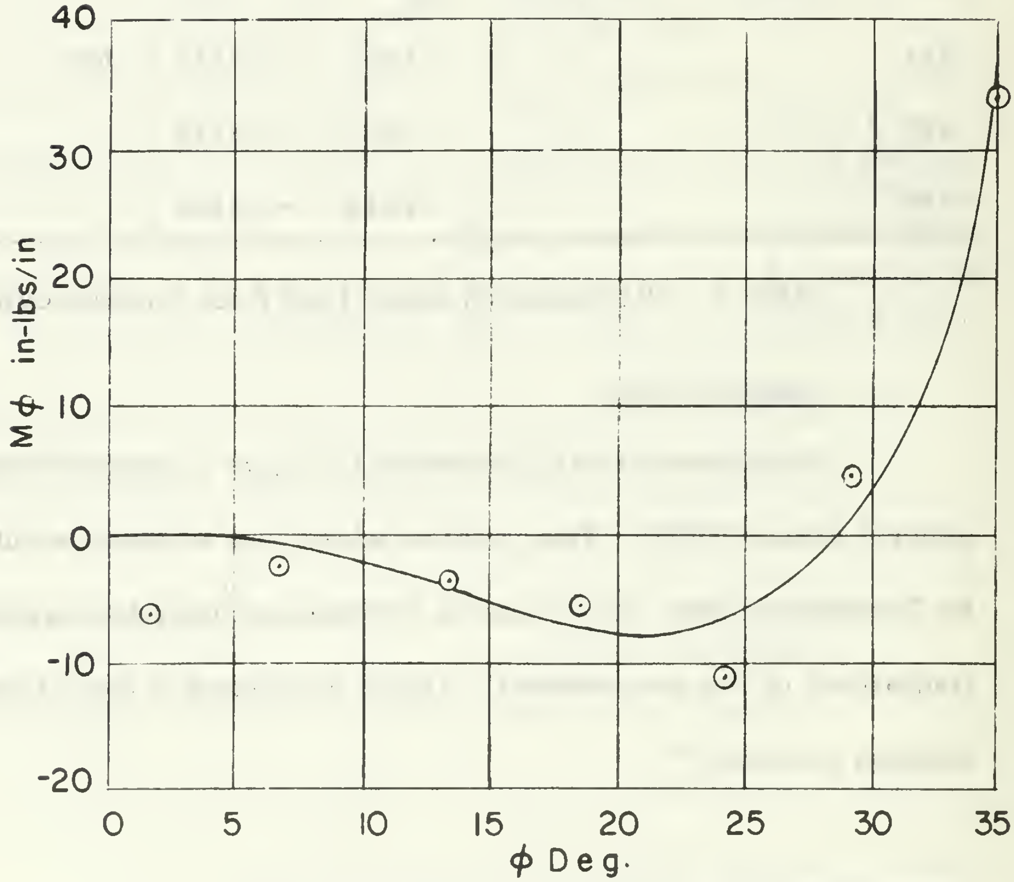


Fig. 6 BENDING MOMENT Mφ VERSUS φ

It is worthy of note that the pole of the cap was located at a singular point ($\phi = 0$). In order to fulfill connectivity requirements, a small rigid polar "cap" was assumed to exist. Boundary conditions were applied to allow this "cap" freedom of displacement in the z-direction at zero slope. Zero slope specification had the effect of introducing a false moment in elements adjacent to the polar cap. This false moment is visible in Figure 6. It is important to recognize, however, that the detrimental effect of the erroneous specification is localized in the near-pole elements.

Results indicate that the bending predictions are quite accurate even though a coarse mesh was used.

C. CONCLUSIONS

A finite element for a shell of revolution which has arbitrarily curved meridians has been developed and tested. Element description was based on displacement functions which met all requirements to guarantee monotonic convergence to an exact solution by mesh size reduction.

Results for classical problems obtained by using the new element indicated that good approximations for bending stresses could be obtained with a very coarse mesh. Mesh size reduction will be necessary to judge element performance in problems where membrane action is predominant.

The geometrical deficiency where ($\phi = 0$) is a limitation which must be recognized by users. Its effect is problem dependent. For the case of the spherical cap it was seen that the detrimental effect of artificially imposed boundary conditions at this singular point did not propagate throughout the solution. On the other hand, the author was unable to successfully approximate stresses in a pressurized toroidal shell due to error propagated from similar artificial boundary conditions. Time precluded the determination of a method to efficiently handle this singularity.

Not withstanding this geometric limitation, results indicate that the new element (1) is accurate, (2) requires low computational effort and (3) does not require an analytical description of shell geometry. Based on these facts it is concluded that the curvilinear finite element can be efficiently used in many engineering applications.

D. RECOMMENDATIONS

It is recommended that:

(1) the stress analysis program be expanded in order to handle realistic engineering problems;

(2) a consistent gravity load vector be included in the stress analysis program; and

(3) a study be conducted to determine a method of efficiently handling the singular point which exists where (ϕ = 0) in the new element.

APPENDIX A NUMERICAL EXAMPLE OF THE STIFFNESS MATRIX

In this appendix, a typical example of the stiffness matrix CSFE is presented. Eigenvalues, before and after rigid body motion was included, and matrix K22 are also presented. Double precision figures were truncated to conserve space.

STIFFNESS MATRIX FOR A TYPICAL SHELL ELEMENT

YOUNG'S MODULUS	=	1.000000
POISSON'S RATIO	=	0.300000
RADIUS 1	=	7.071100
RADIUS 2	=	8.660300
LENGTH	=	2.618000
PHI 1	=	0.785400
PHI 2	=	1.047200
THICKNESS 1	=	1.000000
THICKNESS 2	=	1.000000
ALPHA	=	0.261800

STIFFNESS MATRIX (TRACE= 0.593851D 01)

3.5611D-01	1.5929D-01	-6.8600D-02	2.9499D-02	-4.0483D-02	-1.7844D-02
1.5929D-01	5.3407D-01	-7.2813D-02	2.3204D-02	-6.5356D-03	-1.9920D-03
-6.8600D-02	-7.2813D-02	2.4638D-01	-1.3820D-01	1.7291D-01	7.1317D-02
2.9499D-02	2.3204D-02	-1.3820D-01	1.6268D-01	-8.8986D-02	-6.1067D-02
-4.0483D-02	-6.5356D-03	1.7291D-01	-8.8986D-02	1.7341D-01	6.4635D-02
-1.7844D-02	-1.9920D-03	7.1317D-02	-6.1067D-02	6.4635D-02	4.6593D-02
-1.5309D-01	-4.6740D-02	4.1612D-02	-2.3965D-02	3.8112D-02	1.7317D-02
5.4221D-03	1.7066D-01	-2.0061D-02	1.2751D-02	-6.5605D-04	-1.9728D-03
-1.8526D-02	-2.6942D-02	-2.9875D-02	3.7550D-02	1.0137D-02	5.4018D-03
-3.3664D-02	-1.8040D-02	-4.4005D-02	2.6895D-02	1.0624D-03	3.3152D-03
-3.0417D-02	-5.1456D-03	1.7946D-02	-7.0653D-03	2.0877D-02	2.1061D-02
1.7398D-02	2.0519D-03	-8.9233D-03	5.5501D-03	-1.3830D-02	-1.8096D-02
-1.9986D-01	-1.9887D-01	3.5934D-02	-2.5125D-02	3.3383D-02	1.6557D-02
-1.5971D-01	-2.9334D-01	4.4812D-02	-1.8715D-02	2.7684D-03	2.5494D-03
-3.7327D-02	-5.1499D-02	-1.6658D-02	3.1762D-02	-3.1086D-02	-2.8087D-02
-1.2873D-02	-1.7229D-02	-3.3334D-02	4.1215D-02	-3.2398D-02	-2.4043D-02
3.7270D-02	7.3043D-03	4.4488D-02	-3.8907D-02	4.7924D-02	2.7651D-02
-1.5398D-02	-2.4943D-03	-3.3367D-02	3.0352D-02	-3.1572D-02	-1.6888D-02
-2.6080D-02	-2.9669D-02	5.9622D-03	1.4002D-02	-3.4861D-02	-1.5469D-02
2.9669D-02	-3.8381D-01	4.2499D-02	-1.5900D-02	3.9484D-03	1.2858D-03
5.9622D-03	-4.2499D-02	-1.6326D-01	5.0193D-02	-1.3132D-01	-3.9674D-02
1.4002D-02	1.5900D-02	5.0193D-02	-2.5083D-02	4.3574D-02	1.2615D-02
3.4861D-02	3.9484D-03	1.3132D-01	-4.3574D-02	9.0981D-02	2.9590D-02
1.5469D-02	1.2858D-03	3.9674D-02	-1.2615D-02	2.9590D-02	5.8422D-03

(COL. 1 TO 6)

-1.5309D-01	5.4221D-03	-1.8526D-02	-2.3664D-02	-3.0417D-02	1.7398D-02
-4.6740D-02	1.7066D-01	-2.6942D-02	-1.8040D-02	-5.1455D-03	2.0515D-03
4.1612D-02	-2.0061D-02	-2.9875D-02	-4.4005D-02	1.7946D-02	-8.9233D-03
-2.3965D-02	1.2751D-02	3.7550D-02	2.6895D-02	-7.0653D-03	5.5501D-03
3.8112D-02	-6.5605D-04	1.0137D-02	1.0624D-03	2.0877D-02	-1.8830D-02
1.7317D-02	-1.9728D-03	5.4018D-03	3.3152D-03	2.1061D-02	-1.8096D-02
3.4590D-01	-1.9439D-01	5.4379D-02	2.6077D-02	3.4939D-02	-1.8631D-02
-1.9439D-01	4.7717D-01	-7.4531D-02	-2.3851D-02	-2.2922D-02	7.5601D-03
5.4379D-02	-7.4531D-02	1.8628D-01	1.3340D-01	1.5473D-01	-7.4214D-02
2.6077D-02	-2.3851D-02	1.3340D-01	1.7145D-01	1.0386D-01	-7.5731D-02
3.4939D-02	-2.2922D-02	1.5473D-01	1.0386D-01	2.0193D-01	-8.8457D-02
-1.8631D-02	7.5601D-03	-7.4214D-02	-7.5731D-02	-8.8457D-02	6.6723D-02
4.7325D-02	-7.0173D-02	1.7365D-02	1.7055D-02	2.6866D-02	-1.5613D-02
7.0173D-02	-3.7700D-01	5.4480D-02	2.4512D-02	2.1159D-02	-7.1987D-03
1.7365D-02	-5.4480D-02	-1.0166D-01	-3.8267D-02	-1.0789D-01	3.4468D-02
1.7055D-02	-2.4512D-02	-3.8267D-02	-1.3045D-02	-3.7019D-02	6.8113D-03
-2.6866D-02	2.1159D-02	1.0789D-01	3.7019D-02	3.1981D-02	-2.7257D-02
1.5613D-02	-7.1987D-03	-3.4468D-02	-6.8113D-03	-2.7257D-02	2.5364D-03
-1.9986D-01	1.5971D-01	-3.7327D-02	-1.2873D-02	-2.7270D-02	1.5398D-02
1.9987D-01	-2.9334D-01	5.1499D-02	1.7229D-02	7.3043D-03	-2.4943D-03
3.5934D-02	-4.4812D-02	-1.6658D-02	-3.3334D-02	-4.4483D-02	3.8867D-02
-2.5125D-02	1.8715D-02	3.1962D-02	4.1215D-02	3.9907D-02	-3.0352D-02
-3.8388D-02	2.7684D-03	3.1086D-02	3.2398D-02	4.7924D-02	-3.1572D-02
-1.6557D-02	2.5494D-03	2.8087D-02	2.4043D-02	2.7651D-02	-1.6888D-02

(COL. 7 TO 12)

```
-1.9986D-01-1.5971D-01-3.7327D-02-1.2873D-02 2.7270D-02-1.5398D-02

-1.9887D-01-2.9334D-01-5.1499D-02-1.7229D-02 7.3043D-03-2.4943D-03

 3.5934D-02 4.4812D-02-1.6658D-02-3.3334D-02 4.4483D-02-3.8867D-02

-2.5125D-02-1.8715D-02 3.1962D-02 4.1215D-02-3.8907D-02 3.0352D-02

 3.8388D-02 2.7684D-03-3.1086D-02-3.2398D-02 4.7924D-02-3.1572D-02

 1.6557D-02 2.5494D-03-2.8087D-02-2.4043D-02 2.7651D-02-1.6888D-02

 4.7325D-02 7.0173D-02 1.7365D-02 1.7055D-02-2.6866D-02 1.5613D-02

-7.0173D-02-3.7700D-01-5.4480D-02-2.4512D-02 2.1159D-02-7.1987D-03

 1.7365D-02 5.4480D-02-1.0166D-01-3.8267D-02 1.0789D-01-3.4468D-02

 1.7055D-02 2.4512D-02-3.8267D-02-1.3045D-02 3.7019D-02-6.8113D-03

 2.6866D-02 2.1159D-02-1.0789D-01-3.7019D-02 8.1981D-02-2.7257D-02

-1.5613D-02-7.1987D-03 3.4468D-02 6.8113D-03-2.7257D-02 2.5364D-03

 3.4590D-01 1.9439D-01 5.4379D-02 2.6077D-02-3.4939D-02 1.8631D-02

 1.9439D-01 4.7717D-01 7.4531D-02 2.3851D-02-2.2922D-02 7.5601D-03

 5.4379D-02 7.4531D-02 1.8628D-01 1.3340D-01-1.5473D-01 7.4214D-02

 2.6077D-02 2.3851D-02 1.3340D-01 1.7145D-01-1.0386D-01 7.5731D-02

-3.4939D-02-2.2922D-02-1.5473D-01-1.0386D-01 2.0199D-01-8.8457D-02

 1.8631D-02 7.5601D-03 7.4214D-02 7.5731D-02-8.8457D-02 6.6723D-02

-1.5309D-01-5.4221D-03-1.8526D-02-2.3664D-02 3.0417D-02-1.7398D-02

 4.6740D-02 1.7066D-01 2.6942D-02 1.8040D-02-5.1455D-03 2.0515D-03

 4.1612D-02 2.0061D-02-2.9875D-02-4.4005D-02-1.7946D-02 8.9233D-03

-2.3965D-02-1.2751D-02 3.7550D-02 2.6895D-02 7.0653D-03-5.5501D-03

-3.8112D-02-6.5605D-04-1.0137D-02-1.0624D-03 2.0877D-02-1.8830D-02

-1.7317D-02-1.9728D-03-5.4018D-03-3.3152D-03 2.1061D-02-1.8096D-02


 (COL. 13 TO 18 )
```

-2.6080D-02	2.9669D-02	5.9622D-03	1.4002D-02	3.4861D-02	1.5469D-02
-2.9669D-02	-3.8381D-01	-4.2499D-02	1.5900D-02	3.9484D-03	1.2858D-03
5.9622D-03	4.2499D-02	-1.6326D-01	5.0193D-02	1.3132D-01	3.9674D-02
1.4002D-02	-1.5900D-02	5.0193D-02	-2.5083D-02	-4.3574D-02	-1.2615D-02
-3.4861D-02	3.9484D-03	-1.3132D-01	4.3574D-02	9.0981D-02	2.9590D-02
-1.5469D-02	1.2858D-03	-3.9674D-02	1.2615D-02	2.9590D-02	5.8422D-03
-1.9986D-01	1.9887D-01	3.5934D-02	-2.5125D-02	-3.3388D-02	-1.6557D-02
1.5971D-01	-2.9334D-01	-4.4812D-02	1.8715D-02	2.7684D-03	2.5494D-03
-3.7327D-02	5.1499D-02	-1.6658D-02	3.1962D-02	3.1086D-02	2.8087D-02
-1.2873D-02	1.7229D-02	-3.3334D-02	4.1215D-02	3.2393D-02	2.4043D-02
-2.7270D-02	7.3043D-03	-4.4488D-02	3.3907D-02	4.7924D-02	2.7651D-02
1.5398D-02	-2.4943D-03	3.8867D-02	-3.0352D-02	-3.1572D-02	-1.6888D-02
-1.5309D-01	4.6740D-02	4.1612D-02	-2.3965D-02	-3.8112D-02	-1.7317D-02
-5.4221D-03	1.7066D-01	2.0051D-02	-1.2751D-02	-6.5605D-04	-1.9728D-03
-1.8526D-02	2.6942D-02	-2.9875D-02	3.7550D-02	-1.0137D-02	-5.4018D-03
-2.3664D-02	1.8040D-02	-4.4005D-02	2.6895D-02	-1.0624D-03	-3.3152D-03
3.0417D-02	-5.1455D-03	-1.7946D-02	7.0653D-03	2.0877D-02	2.1061D-02
-1.7398D-02	2.0515D-03	8.9233D-03	-5.5501D-03	-1.8830D-02	-1.8096D-02
3.5611D-01	-1.5929D-01	-6.8600D-02	2.9499D-02	4.0483D-02	1.7844D-02
-1.5929D-01	5.3407D-01	7.2813D-02	-2.3204D-02	-6.5356D-03	-1.9920D-03
-6.8600D-02	7.2813D-02	2.4688D-01	-1.3820D-01	-1.7291D-01	-7.1317D-02
2.9499D-02	-2.3204D-02	-1.3820D-01	1.6268D-01	8.8986D-02	6.1067D-02
4.0483D-02	-6.5356D-03	-1.7291D-01	8.8986D-02	1.7341D-01	6.4635D-02
1.7844D-02	-1.9920D-03	-7.1317D-02	6.1067D-02	6.4635D-02	4.6593D-02

(COL. 19 TO 24)

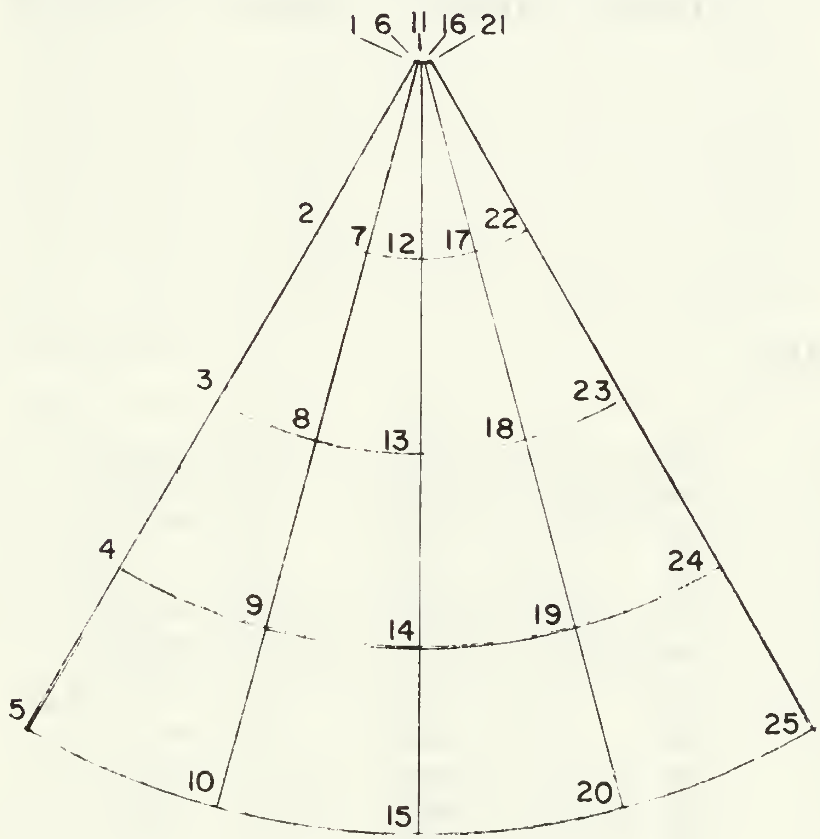
```
EIGENVALUES BEFORE RIGID BODY MOTION INCLUDED

1.6079D 00 8.9358D-01 8.0319D-01 7.5080D-01 6.8212D-01 5.8128D-01

4.4740D-01 4.1214D-01 1.6854D-01 1.0160D-01 8.7839D-02 8.4986D-02

6.1099D-02 4.8384D-02 1.8723D-02 1.6404D-02 1.2280D-02 6.2271D-03

4.5160D-03 3.6494D-03 2.6296D-04 3.5945D-05 1.0945D-06 1.5447D-07




MATRIX K22

 2.1512D-04-4.6499D-05-1.6498D-04 6.5636D-05 1.6977D-04-2.0939D-05

-4.6499D-05 5.6219D-04-2.1721D-05-6.5969D-04-6.5636D-05 1.5905D-04

-1.6498D-04-2.1721D-05 2.1494D-04-1.0862D-04 8.2509D-04-6.6052D-17

 6.5636D-05-6.5969D-04-1.0862D-04 5.3243D-03-8.6983D-04-6.1624D-03

 1.6977D-04-6.5636D-05 8.2508D-04-8.6983D-04 1.2317D-02-8.1130D-04

-2.0939D-05 1.5905D-04-2.6369D-16-6.1624D-03-8.1130D-04 8.5078D-03




EIGENVALUES AFTER RIGID BODY MOTION WAS INCLUDED

1.6039D 00 8.7532D-01 7.0365D-01 5.7682D-01 5.7055D-01 4.6802D-01

3.7658D-01 2.8611D-01 1.0353D-01 1.0083D-01 8.7779D-02 7.2831D-02

4.2277D-02 2.0689D-02 1.7525D-02 1.5052D-02 1.1759D-02 5.3237D-03

9.2971D-17 9.1124D-17 3.9136D-17-7.7923D-18-4.6515D-17-7.7301D-17
```

APPENDIX B COMPUTER OUTPUT

In this appendix, a complete set of results for the spherical cap analyzed in this study are presented. Results for the spherical cap were obtained using a 4x4 mesh. The mesh layout accompanies the problem.

SPHERICAL SHELL WITH EXTERNAL PRESSURE (1 PSI)

TOTAL NUMBER OF ELEMENTS	=	16
NUMBER OF DIFFERENT TYPE ELEMENTS	=	4
TOTAL NUMBER OF NODES	=	25
NUMBER OF NODES WITH BOUNDARY CONDITIONS	=	25
HALFBAND WIDTH	=	42
NUMBER OF POINTS WITH CONCENTRATED LOADS	=	0



MESH LAYOUT

EL	SPEC	E /POIS	RAD1/RAD2	PHI1/PHI2	H1/H2	L /ALPHA
1	GEN	4.0000D 06	3.1491D 00	3.4910D-02	3.0000D 00	1.2567D 01
		1.6670D-01	1.5628D 01	1.7453D-01	3.0000D 00	3.9270D-01
2	GEN	4.0000D 06	1.5628D 01	1.7453D-01	3.0000D 00	1.3351D 01
		1.6670D-01	2.8557D 01	3.2289D-01	3.0000D 00	3.9270D-01
3	GEN	4.0000D 06	2.8557D 01	3.2289D-01	3.0000D 00	1.3351D 01
		1.6670D-01	4.0859D 01	4.7124D-01	3.0000D 00	3.9270D-01
4	GEN	4.0000D 06	4.0859D 01	4.7124D-01	3.0000D 00	1.2567D 01
		1.6670D-01	5.1622D 01	6.1087D-01	3.0000D 00	3.9270D-01

CONNECTIVITY

I	J	K	L	TYPE
1	2	7	6	1
2	3	8	7	2
3	4	9	8	3
4	5	10	9	4
6	7	12	11	1
7	8	13	12	2
8	9	14	13	3
9	10	15	14	4
11	12	17	16	1
12	13	18	17	2
13	14	19	18	3
14	15	20	19	4
16	17	22	21	1
17	18	23	22	2
18	19	24	23	3
19	20	25	24	4

BOUNDARY CONDITIONS

JOINT	X TRANS	Y TRANS	Z TRANS	W,X	W,Y	W,XY
1	0	0	1	0	0	0
6	0	0	1	0	0	0
11	0	0	1	0	0	0
16	0	0	1	0	0	0
21	0	0	1	0	0	0
5	0	0	0	0	0	0
10	0	0	0	0	0	0
15	0	0	0	0	0	0
20	0	0	0	0	0	0
25	0	0	0	0	0	0
2	1	0	1	1	0	0
3	1	0	1	1	0	0
4	1	0	1	1	0	0
22	1	0	1	1	0	0
23	1	0	1	1	0	0
24	1	0	1	1	0	0
7	1	1	1	1	0	0
8	1	1	1	1	0	0
9	1	1	1	1	0	0
12	1	1	1	1	0	0
13	1	1	1	1	0	0
14	1	1	1	1	0	0
17	1	1	1	1	0	0
18	1	1	1	1	0	0
19	1	1	1	1	0	0

DISPLACEMENTS

JOINT	U	V	W	W,X	W,Y	W,XY
1	-0.0	-0.0	-6.0021D-04	-0.0	-0.0	-0.0
2	3.0912D-05	-0.0	-5.6411D-04	-5.1250D-06	-0.0	-0.0
3	6.4830D-05	-0.0	-4.3231D-04	-1.5035D-05	-0.0	-0.0
4	6.1524D-05	-0.0	-1.8226D-04	-2.1583D-05	-0.0	-0.0
5	-0.0	-0.0	-0.0	-0.0	-0.0	-0.0
6	-0.0	-0.0	-6.0021D-04	-0.0	-0.0	-0.0
7	3.0912D-05	-3.2051D-17	-5.6411D-04	-5.1250D-06	-0.0	-0.0
8	6.4830D-05	2.0449D-18	-4.3231D-04	-1.5035D-05	-0.0	-0.0
9	6.1524D-05	1.7595D-16	-1.8226D-04	-2.1583D-05	-0.0	-0.0
10	-0.0	-0.0	-0.0	-0.0	-0.0	-0.0
11	-0.0	-0.0	-6.0021D-04	-0.0	-0.0	-0.0
12	3.0912D-05	-4.8160D-17	-5.6411D-04	-5.1250D-06	-0.0	-0.0
13	6.4830D-05	4.3083D-17	-4.3231D-04	-1.5035D-05	-0.0	-0.0
14	6.1524D-05	2.2962D-16	-1.8226D-04	-2.1583D-05	-0.0	-0.0
15	-0.0	-0.0	-0.0	-0.0	-0.0	-0.0
16	-0.0	-0.0	-6.0021D-04	-0.0	-0.0	-0.0
17	3.0912D-05	-3.2074D-17	-5.6411D-04	-5.1250D-06	-0.0	-0.0
18	6.4830D-05	1.9988D-18	-4.3231D-04	-1.5035D-05	-0.0	-0.0
19	6.1524D-05	1.7593D-16	-1.8226D-04	-2.1583D-05	-0.0	-0.0
20	-0.0	-0.0	-0.0	-0.0	-0.0	-0.0
21	-0.0	-0.0	-6.0021D-04	-0.0	-0.0	-0.0
22	3.0912D-05	-0.0	-5.6411D-04	-5.1250D-06	-0.0	-0.0
23	6.4830D-05	-0.0	-4.3231D-04	-1.5035D-05	-0.0	-0.0
24	6.1524D-05	-0.0	-1.8226D-04	-2.1583D-05	-0.0	-0.0
25	-0.0	-0.0	-0.0	-0.0	-0.0	0.0

STRESSES

JOINT	NX	NY	NXY	MX	MY	MXY
1	-1.3299D 02	-1.0200D 02	0.0	-4.6892D 00	-7.8170D-01	0.0
2	-4.9417D 01	-6.0075D 01	-3.8050D-12	-4.4468D 00	-3.4681D 00	-2.9254D-14
3	-5.0434D 01	-4.0216D 01	-3.9636D-11	-7.6273D 00	-5.5502D 00	-3.2112D-13
4	-5.8230D 01	-1.7908D 01	-5.1070D-11	-2.6165D 00	-4.5375D 00	-4.2703D-13
5	-6.0429D 01	-1.0073D 01	0.0	3.1799D 01	5.3010D 00	0.0
6	-1.3299D 02	-1.0200D 02	-1.3116D-11	-4.6892D 00	-7.8170D-01	-3.1238D-12
7	-4.9417D 01	-6.0075D 01	-1.3746D-11	-4.4468D 00	-3.4681D 00	9.1164D-13
8	-5.0434D 01	-4.0216D 01	7.3553D-12	-7.6273D 00	-5.5502D 00	7.6483D-13
9	-5.8230D 01	-1.7908D 01	-4.4728D-11	-2.6165D 00	-4.5375D 00	-9.5848D-13
10	-6.0429D 01	-1.0073D 01	-7.2004D-11	3.1799D 01	5.3010D 00	-1.0461D-12
11	-1.3299D 02	-1.0200D 02	-1.9709D-11	-4.6892D 00	-7.8170D-01	-4.6939D-12
12	-4.9417D 01	-6.0075D 01	-2.1166D-11	-4.4468D 00	-3.4681D 00	1.5262D-12
13	-5.0434D 01	-4.0216D 01	2.0061D-11	-7.6273D 00	-5.5502D 00	8.0177D-13
14	-5.8230D 01	-1.7908D 01	-3.0717D-11	-2.6165D 00	-4.5375D 00	-1.0974D-12
15	-6.0429D 01	-1.0073D 01	-9.3966D-11	3.1799D 01	5.3010D 00	-1.3652D-12
16	-1.3299D 02	-1.0200D 02	-1.3126D-11	-4.6892D 00	-7.8170D-01	-3.1261D-12
17	-4.9417D 01	-6.0075D 01	-1.3763D-11	-4.4468D 00	-3.4681D 00	8.1180D-13
18	-5.0434D 01	-4.0216D 01	7.3259D-12	-7.6273D 00	-5.5502D 00	7.6454D-13
19	-5.8230D 01	-1.7908D 01	-4.4746D-11	-2.6165D 00	-4.5375D 00	-9.5850D-13
20	-6.0429D 01	-1.0073D 01	-7.1937D-11	3.1799D 01	5.3010D 00	-1.0460D-12
21	-1.3299D 02	-1.0200D 02	0.0	-4.6892D 00	-7.8170D-01	0.0
22	-4.9417D 01	-6.0075D 01	-3.8121D-12	-4.4468D 00	-3.4681D 00	-2.9316D-14
23	-5.0434D 01	-4.0216D 01	-3.8703D-11	-7.6273D 00	-5.5502D 00	-3.2166D-13
24	-5.8230D 01	-1.7908D 01	-5.1118D-11	-2.6165D 00	-4.5375D 00	-4.2743D-13
25	-6.0429D 01	-1.0073D 01	0.0	3.1799D 01	5.3010D 00	1.2995D-25

STRAINS

JOINT	SNX	SNY	SNXY	SMX	SMY	SMXY
1	-9.6658D-06	-6.6525D-06	0.0	-5.0655D-07	-2.1684D-19	0.0
2	-3.2836D-06	-4.3198D-06	-7.3967D-19	-4.2985D-07	-3.0298D-07	-7.5844D-21
3	-3.6442D-06	-2.6507D-06	-7.5132D-18	-7.4467D-07	-4.7541D-07	-8.3254D-20
4	-4.6037D-06	-6.8344D-07	-9.9304D-18	-2.0668D-07	-4.5570D-07	-1.1072D-19
5	-4.8958D-06	2.9952D-22	0.0	3.4351D-06	-3.7441D-22	0.0
6	-9.6658D-06	-6.6525D-06	-2.5509D-18	-5.0655D-07	-1.1925D-18	-8.0990D-19
7	-3.2836D-06	-4.3198D-06	-2.6730D-18	-4.2985D-07	-3.0298D-07	2.1043D-19
8	-3.6442D-06	-2.6507D-06	1.4301D-18	-7.4467D-07	-4.7541D-07	1.9830D-19
9	-4.6037D-06	-6.8344D-07	-8.6974D-18	-2.0668D-07	-4.5570D-07	-2.4950D-19
10	-4.8958D-06	-3.3280D-22	-1.4001D-17	3.4351D-06	-3.0368D-22	-2.7122D-19
11	-9.6658D-06	-6.6525D-06	-3.8324D-18	-5.0655D-07	-1.8974D-19	-1.2170D-18
12	-3.2836D-06	-4.3198D-06	-4.1159D-19	-4.2985D-07	-3.0298D-07	3.3569D-19
13	-3.6442D-06	-2.6507D-06	3.9004D-18	-7.4467D-07	-4.7541D-07	2.0798D-19
14	-4.6037D-06	-6.8344D-07	-5.9731D-18	-2.0668D-07	-4.5570D-07	-2.8452D-19
15	-4.8958D-06	-3.3280D-22	-1.8272D-17	3.4351D-06	-3.0368D-22	-3.5395D-19
16	-9.6658D-06	-6.6525D-06	-2.5523D-18	-5.0655D-07	1.2739D-18	-8.1050D-19
17	-3.2836D-06	-4.3198D-06	-2.6767D-18	-4.2985D-07	-3.0298D-07	2.1047D-19
18	-3.6442D-06	-2.6507D-06	1.4239D-18	-7.4467D-07	-4.7541D-07	1.9833D-19
19	-4.6037D-06	-6.8344D-07	-8.7011D-18	-2.0668D-07	-4.5570D-07	-2.4851D-19
20	-4.8958D-06	-3.3280D-22	-1.4000D-17	3.4351D-06	-3.0368D-22	-2.7120D-19
21	-9.6658D-06	-6.6525D-06	0.0	-5.0655D-07	3.7947D-19	0.0
22	-3.2836D-06	-4.3198D-06	-7.4137D-19	-4.2985D-07	-3.0298D-07	-7.6003D-21
23	-3.6442D-06	-2.6507D-06	-7.5261D-18	-7.4467D-07	-4.7541D-07	-8.3396D-20
24	-4.6037D-06	-6.8344D-07	-9.9379D-18	-2.0668D-07	-4.5570D-07	-1.1092D-19
25	-4.8958D-06	-9.6513D-22	0.0	3.4351D-06	-2.3296D-22	3.3692D-32

APPENDIX C COMPUTER PROGRAM

In this appendix the stress-strain analysis program is presented. User and functional information are presented through the use of comment cards located throughout the program.

The program input data deck is illustrated below. Data values presented were those used for the spherical cap (2x2 mesh).

```
12345678901234567890123456789012345678901234567890
/*    Orange  card
      BLANK
      BLANK
      BLANK
         9
         8         1                  1         1
         7                            1
         6
         5         1         1        1         1
         4                            1
         3
         2         1                  1         1
         1                            1
         5         6         9        8         2
         4         5         8        7         1
         2         3         6        5         2
         1         2         5        4         1
4000000.  .1667     3.        3.       -1.
27.8118   51.6222   27.6039   .31416   .61087    .7854
4000000.  .1667     3.        3.       -1.
3.1491    27.8118   25.1325   .03491   .31416    .7854
         4         2         9        9        30
   TEST PROB. (SPHERICAL CAP  2x2 MESH)
```

```
C   STRESS ANALYSIS PROGRAM FOR CONSTRUCTING AND TESTING CSFE
C
C   CONTROL SUBPROGRAM
        IMPLICIT REAL*8 (A-H,O-Z)
        REAL*8 K1,K2,K3,L
        COMMON/B1/E,H1,H2,POIS,RAD1,RAD2,L,PHI1,PHI2,ALPHA,
       #THETA,KASE
        COMMON/MBN/BLOAD(294),NBC(50,7),NCON(36,5)
        COMMON/PARM/NEL,NELT,NDPT,NPBC,NBAND,NEQ
   50   DO 100 I=1,294
  100   BLOAD(I)=0.0D0
        CALL INPUT
        IF(NELT.EQ.0) STOP
        CALL OV1
        CALL OV2
        GO TO 50
        END

        SUBROUTINE OV1
        IMPLICIT REAL*8 (A-H,O-Z)
        COMMON/MBN/BLOAD(294),NBC(50,7),NCON(36,5)
        COMMON/PARM/NEL,NELT,NDPT,NPBC,NBAND,NEQ
        DIMENSION TOTK(294,54)
        DO 100 I=1,294
        DO 100 J=1,54
  100   TOTK(I,J)=0.0D0
        CALL FORML
        CALL MERGER (TOTK)
        CALL BANSOL(TOTK,BLOAD,NEQ,NBAND)
        RETURN
        END

        SUBROUTINE OV2
        IMPLICIT REAL*8 (A-H,O-Z)
        CALL OUTPUT
        CALL PSSN
        RETURN
        END

        SUBROUTINE INPUT
        IMPLICIT REAL*8 (A-H,O-Z)
        REAL*8 K1,K2,K3,L
        INTEGER*4 SPTY(4),SPEC
        DIMENSION STFMOD(24,24),TINV(24,24),TINVT(24,24)
        DIMENSION PVEC(24),CFORCE(6)
        COMMON/B1/E,H1,H2,POIS,RAD1,RAD2,L,PHI1,PHI2,ALPHA,
       #THETA,KASE
        COMMON/MBN/BLOAD(294),NBC(50,7),NCON(36,5)
        COMMON/PARM/NEL,NELT,NDPT,NPBC,NBAND,NEQ
        COMMON/SS SN/SS(24,24),SN(24,24)
        DATA SPTY/'GEN CYL PLT CONE'/
C   BLOAD IS TOTAL LOAD VECTOR
C   NBC IDENTIFIES THE NODE POINT WITH ITS BOUNDARY CONDITION
C   NCON IDENTIFIES ELEMENT NODAL POINT AND ELEMENT TYPE
C   NBAND IS EQUAL TO HALFBAND WIDTH
C   NPCL IS THE NUMBER OF POINTS WITH CONCENTRATED LOADS
        NTK=0
C   NTK IS CYLINDER TRACK COUNTER
        WRITE(6,1100)
 1100   FORMAT(1H1,//)
        READ (5,1000)
        WRITE (6,1000)
 1000   FORMAT(80H
       #                                                  )
        READ (5,2000) NELT,NEL,NDPT,NPBC,NBAND,NPCL
 2000   FORMAT(6I10)
        IF(NELT.EQ.0) RETURN
        NEQ=6*NDPT
C   NELT IS TOTAL NUMBER OF ELEMENTS
```

```
C   NEL IS NUMBER OF DIFFERENT TYPE ELEMENTS
C   NDPT IS TOTAL NUMBER OF NODES
C   NPBC IS NUMBER OF NODE POINTS WITH BOUNDARY CONDITIONS
      WRITE (6,2500) NELT,NEL,NDPT,NPBC,NBAND,NPCL
 2500 FORMAT(/,38X,'TOTAL NUMBER OF ELEMENTS',18X,'=',I10,/,
     #38X,'NUMBER OF DIFFERENT TYPE ELEMENTS',9X,'=',I10,/,
     #38X,'TOTAL NUMBER OF NODES',21X,'=',I10,/,
     #38X,'NUMBER OF NODES WITH BOUNDARY CONDITIONS  =',I10,
     #/,38X,'HALFBAND WIDTH',28X,'=',I10,/,38X,
     #'NUMBER OF POINTS WITH CONCENTRATED LOADS  =',I10)
      WRITE (6,3000)
 3000 FORMAT(1H1,///,32X,'EL   SPEC      E /POIS     RAD1/',
     #'RAD2   PHI1/PHI2      H1/H2       L /ALPHA')
      DO 100 I=1,NEL
      READ(5,3500) RAD1,RAD2,L,PHI1,PHI2,ALPHA
 3500 FORMAT(7F10.0)
      READ (5,4000) E,POIS,H1,H2,P,KASE
 4000 FORMAT(5F10.0,I10)
C   P IS THE PRESSURE LOAD
C   KASE SPECIFICATIONS FOLLOW
C                            KASE=0  CURVILINEAR SHELL
C                            KASE=1  CYLINDRICAL SHELL
C                            KASE=2  FLAT PLATE
C                            KASE=3  TRUNCATED CONE
C   IF A FLAT PLATE IS SPECIFIED WIDTH IS WRITTEN FOR ALPHA
      IF(KASE.EQ.2) ALPHA=1.0D0/THETA
      IF(KASE.EQ.0) SPEC=SPTY(1)
      IF(KASE.EQ.1) SPEC=SPTY(2)
      IF(KASE.EQ.2) SPEC=SPTY(3)
      IF(KASE.EQ.3) SPEC=SPTY(4)
      WRITE (6,4100) I,SPEC,E,RAD1,PHI1,H1,L,POIS,RAD2,PHI2,
     #H2,ALPHA
 4100 FORMAT(//,32X,I2,3X,A4,2X,1P5D12.4//43X,1P5D12.4)
      CALL ELSTF(STFMOD)
      NTK=NTK+1
      CALL WRDISK(NTK,STFMOD,576)
      CALL TFORM(TINV,TINVT)
      CALL STRESS(TINV,SS,SN)
      NTK=NTK+1
      CALL WRDISK(NTK,SN,576)
      NTK=NTK+1
      CALL WRDISK(NTK,SS,576)
      NTK=NTK+1
      CALL PLOAD(P,PVEC)
      CALL WRDISK(NTK,PVEC,24)
  100 CONTINUE
      WRITE (6,4400)
 4400 FORMAT(1H1,///,32X,'CONNECTIVITY')
      WRITE (6,4500)
 4500 FORMAT(//,33X,'I',15X,'J',15X,'K',15X,'L',13X,'TYPE')
      DO 200 I=1,NELT
      READ (5,5000) (NCON(I,J),J=1,5)
 5000 FORMAT(5I10)
      WRITE(6,5500) (NCON(I,J),J=1,5)
 5500 FORMAT(/,32X,I2,14X,I2,14X,I2,14X,I2,14X,I2)
  200 CONTINUE
      WRITE (6,6000)
 6000 FORMAT(1H1,///,27X,'BOUNDARY CONDITIONS')
      WRITE(6,6100)
 6100 FORMAT(//,27X,'JOINT',3X,'X TRANS',5X,'Y TRANS',5X,
     #'Z TRANS',8X,'W,X',9X,'W,Y',9X,'W,XY')
C   READ BOUNDARY CONDITIONS
      DO 300 I=1,NPBC
      READ (5,6500) (NBC(I,J),J=1,7)
 6500 FORMAT(7I10)
      WRITE(6,7000) (NBC(I,J),J=1,7)
 7000 FORMAT(/,29X,I2, 8X,I1,11X,I1,11X,I1,11X,I1,11X,I1,
     #11X,I1)
  300 CONTINUE
      IF(NPCL.EQ.0) RETURN
      WRITE (6,7050)
 7050 FORMAT(1H1,///,27X,'CONCENTRATED LOADS')
```

```
      WRITE (6,7100)
 7100 FORMAT(//,27X,'JOINT'3X,'FORCE X',5X,'FORCE Y',5X,
     #'FORCE Z',8X,'W,X',9X,'W,Y',9X,'W,XY')
C   READ CONCENTRATED LOADS
      DO 500 I=1,NPCL
      READ (5,7200) JNR,(CFORCE(K),K=1,6)
 7200 FORMAT (I10,6F10.0)
      WRITE (6,7300) JNR,(CFORCE(M),M=1,6)
 7300 FORMAT(/,19X,I12,1P6D12.4)
      IB=6*JNR-6
      DO 400 J=1,6
      IBL=IB+J
  400 BLOAD(IBL)=CFORCE(J)
  500 CONTINUE
      RETURN
      END


      FUNCTION COSPHI(X)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K1,K2,K3,L
      COMMON/B1/E,H1,H2,POIS,RAD1,RAD2,L,PHI1,PHI2,ALPHA,
     #THETA,KASE
      COMMON/B2/CPHI1,CPHI2,C0,C1,C2
      COSPHI      =C0+C1*X+C2*X**2
      IF(KASE.GE.1)  COSPHI=0.0D0
      IF(KASE.EQ.3)  COSPHI=DCOS(PHI1)
      IF(DABS(COSPHI).GT.1.0D0) GO TO 1
      RETURN
    1 CONTINUE
      CALL FLAG(2,&100,&200)
  100 CONTINUE
  200 CONTINUE
C   VALUE ASSIGNED IS INTERIM VALUE ONLY
      COSPHI=1.0D0
      KASE=9
      RETURN
      END


      FUNCTION SINPHI(X)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K1,K2,K3,L
      SINPHI=DSQRT(1.0D0-(COSPHI(X))**2)
      RETURN
      END


      FUNCTION R(X)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K1,K2,K3,L
      COMMON/B1/E,H1,H2,POIS,RAD1,RAD2,L,PHI1,PHI2,ALPHA,
     #THETA,KASE
      COMMON/B2/CPHI1,CPHI2,C0,C1,C2
      IF(KASE.EQ.2)  GO TO 1
      R      =RAD1+L*(C0*X+(C1*X**2)/2.0D0+(C2*X**3)/3.0D0)
      IF(KASE.EQ.1)    R=RAD1
      RETURN
    1 CONTINUE
C   VALUE ASSIGNED IS INTERIM VALUE ONLY
      R=1.0D0
      RETURN
      END


      FUNCTION Q(X)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K1,K2,K3,L
      COMMON/B1/E,H1,H2,POIS,RAD1,RAD2,L,PHI1,PHI2,ALPHA,
     #THETA,KASE
      Q=1.0D0/R(X)
      IF(KASE.EQ.2)  Q=0.0D0
```

```
      RETURN
      END

      FUNCTION QQ(X)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K1,K2,K3,L
      COMMON/B1/E,H1,H2,POIS,RAD1,RAD2,L,PHI1,PHI2,ALPHA,
     #THETA,KASE
      QQ=Q(X)/ALPHA
      IF(KASE.EQ.2)  QQ=THETA
      RETURN
      END

      FUNCTION R1(X)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K1,K2,K3,L
      COMMON/B1/E,H1,H2,POIS,RAD1,RAD2,L,PHI1,PHI2,ALPHA,
     #THETA,KASE
      COMMON/B2/CPHI1,CPHI2,C0,C1,C2
      IF(KASE.GE.1)  GO TO 1
      R1=-L*SINPHI(X)/(C1+2.0D0*X*C2)
C  THE FOLLOWING CARD PROTECTS AGAINST OVERFLOW IF X IS
C  EXACTLY AT A POINT OF INFLECTION
      IF(R1.GT.1.0D50) GO TO 1
      RETURN
    1 CONTINUE
C  VALUE ASSIGNED IS INTERIM VALUE ONLY
      R1=1.0D0
      RETURN
      END

      FUNCTION Q1(X)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K1,K2,K3,L
      COMMON/B1/E,H1,H2,POIS,RAD1,RAD2,L,PHI1,PHI2,ALPHA,
     #THETA,KASE
      Q1=1.0D0/R1(X)
      IF(KASE.GE.1)  Q1=0.0D0
      RETURN
      END

      FUNCTION DR(X)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K1,K2,K3,L
      COMMON/B1/E,H1,H2,POIS,RAD1,RAD2,L,PHI1,PHI2,ALPHA,
     #THETA,KASE
      DR=L*COSPHI(X)
      RETURN
      END

      FUNCTION DR1(X)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K1,K2,K3,L
      COMMON/B1/E,H1,H2,POIS,RAD1,RAD2,L,PHI1,PHI2,ALPHA,
     #THETA,KASE
      COMMON/B2/CPHI1,CPHI2,C0,C1,C2
      IF(KASE.GE.1)  GO TO 1
      DR1=DR(X)/SINPHI(X)+2.0D0*C2*(-R1(X))/(C1+2.0D0*X*C2)
      RETURN
    1 CONTINUE
      DR1=0.0D0
      RETURN
      END
```

```
      SUBROUTINE ELSTF(STFMOD)
C   ELSTF IS THE CONTROL PROGRAM FOR CONSTRUCTING THE
C   ELEMENT STIFFNESS MATRIX.
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K1,K2,K3,L
      DIMENSION TINVT(24,24),TINV(24,24),XG(8),WTW(8,8),
     #STIFF1(24,24),STF(24,24),STK(24,24),WT(8),
     #STIFFF(24,24),STFMOD(24,24)
      COMMON/B1/E,H1,H2,POIS,RAD1,RAD2,L,PHI1,PHI2,ALPHA,
     #THETA,KASE
      COMMON/B2/CPHI1,CPHI2,C0,C1,C2
      CPHI1=DCOS(PHI1)
      CPHI2=DCOS(PHI2)
      C0=CPHI1
      C1=2.0D0*(3.0D0*(RAD2-RAD1)/L -2.0D0*CPHI1-CPHI2)
      C2=3.0D0*(-2.0D0*(RAD2-RAD1)/L+CPHI1+CPHI2)
      IF(KASE.EQ.0) GO TO 20
      C1=0.0D0
      C2=0.0D0
   20 CONTINUE
      DO 50 I=1,24
      DO 50 J=1,24
   50 STK(I,J)=0.0D0
C ZEROS OF GAUSSIAN 8 POINT FORMULAS ARE
      XX1=.96028985649753623D0
      XX2=.79666647741362674D0
      XX3=.52553240991632899D0
      XX4=.18343464249564980D0
C   COEFFICIENTS OF ZEROS ARE
      AA1=.10122853629037626D0
      AA2=.22238103445337447D0
      AA3=.31370664587788729D0
      AA4=.36268378337836198D0
C   XG CONTAINS ALL THE 8 ZEROS AND WT CONTAINS ALL THE
C   COEFFICIENTS.  ALL ARE SHIFTED TO RANGE 0 TO 1.
      WT(1)=AA1*0.5D0
      WT(2)=AA2*0.5D0
      WT(3)=AA3*0.5D0
      WT(4)=AA4*0.5D0
      WT(5)=WT(4)
      WT(6)=WT(3)
      WT(7)=WT(2)
      WT(8)=WT(1)
      A=0.5D0
      XG(1)=A+XX1*0.5D0
      XG(2)=A+XX2*0.5D0
      XG(3)=A+XX3*0.5D0
      XG(4)=A+XX4*0.5D0
      XG(5)=A-XX4*0.5D0
      XG(6)=A-XX3*0.5D0
      XG(7)=A-XX2*0.5D0
      XG(8)=A-XX1*0.5D0
      DO 100 I=1,8
      DO 100 J=1,8
  100 WTW(I,J)=WT(I)*WT(J)
      DO 200 I=1,8
      X=XG(I)
      DO 200 J=1,8
      Y=XG(J)
      CALL FORMK(STF,X,Y)
      RALF=1.0D0/QQ(X)
      DO 200 K=1,24
      DO 200 M=1,24
  200 STK(K,M)=STK(K,M)+WTW(J,I)*STF(K,M)*L*RALF
      DO 300 I=1,24
      DO 300 J=1,24
      STK(I,J)=(STK(I,J)+STK(J,I))*0.5D0
  300 STK(J,I)=STK(I,J)
```

```
      IF(KASE.EQ.9) RETURN
      CALL TFORM(TINV,TINVT)
      CALL MULT(TINVT,STK,STIFF1,24,24,24)
      CALL MULT(STIFF1,TINV,STIFFF,24,24,24)
      DO 600 I=1,24
      DO 600 J=1,24
      STIFFF(I,J)=0.5D0*(STIFFF(I,J)+STIFFF(J,I))
      STIFFF(J,I)=STIFFF(I,J)
  600 CONTINUE
      IF(KASE.EQ.2) RETURN
      CALL RIGID (STIFFF,STFMOD)
      RETURN
      END

      SUBROUTINE FORMK(STF,X,Y)
C   FORMK FORMS THE ELEMENT STIFFNESS MATRIX.
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K1,K2,K3,L
      DIMENSION STF(24,24),WORK1(6,24),DEL(6,6),P(6,24)
      CALL DEL1(DEL,X)
      CALL PST(P,X,Y)
      DO 200 I=1,6
      DO 200 J=1,24
      WORK1(I,J)=0.0D0
      DO 200 K=1,6
  200 WORK1(I,J)=WORK1(I,J)+DEL(I,K)*P(K,J)
      DO 300 I=1,24
      DO 300 J=1,24
      STF(I,J)=0.0D0
      DO 300 K=1,6
  300 STF(I,J)=STF(I,J)+P(K,I)*WORK1(K,J)
      RETURN
      END

      SUBROUTINE TFORM(TINV,TINVT)
C   TFORM FORMS THE TRANSFORMATION MATRIX AND ITS INVERSE.
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K1,K2,K3,L
      DIMENSION TINVT(24,24),TINV(24,24)
      COMMON/B1/E,H1,H2,POIS,RAD1,RAD2,L,PHI1,PHI2,ALPHA,
     #THETA,KASE
C   INITIALIZE TINVT MATRIX
      DO 350 I=1,24
      DO 350 J=1,24
  350 TINVT(I,J)=0.0D0
      IF(KASE.EQ.2) ALPHA=WIDTH
      IF(KASE.EQ.2) RAD1=1.0D0
      IF(KASE.EQ.2) RAD2=1.0D0
      TINVT( 1, 1)=1.0D0
      TINVT( 1, 2)=-1.0D0
      TINVT( 1, 3)=-1.0D0
      TINVT( 1, 4)=1.0D0
      TINVT( 2, 5)=1.0D0
      TINVT( 2, 6)=-1.0D0
      TINVT( 2, 7)=-1.0D0
      TINVT( 2, 8)=1.0D0
      TINVT( 3, 9)=4.0D0
      TINVT( 3,10)=-6.0D0
      TINVT( 3,12)=2.0D0
      TINVT( 3,13)=-6.0D0
      TINVT( 3,14)=9.0D0
      TINVT( 3,16)=-3.0D0
      TINVT( 3,21)=2.0D0
      TINVT( 3,22)=-3.0D0
      TINVT( 3,24)=1.0D0
      TINVT( 4, 9)=2.0D0   *L
      TINVT( 4,10)=-3.0D0  *L
      TINVT( 4,12)=1.0D0   *L
      TINVT( 4,13)=-4.0D0  *L
      TINVT( 4,14)=6.0D0   *L
```

```
TINVT( 4,16)=-2.0D0 *L
TINVT( 4,17)=2.0D0  *L
TINVT( 4,18)=-3.0D0 *L
TINVT( 4,20)=1.0D0  *L
TINVT( 5, 9)=2.0D0  *ALPHA*RAD1
TINVT( 5,10)=-4.0D0 *ALPHA*RAD1
TINVT( 5,11)=2.0D0  *ALPHA*RAD1
TINVT( 5,13)=-3.0D0 *ALPHA*RAD1
TINVT( 5,14)=6.0D0  *ALPHA*RAD1
TINVT( 5,15)=-3.0D0 *ALPHA*RAD1
TINVT( 5,21)=1.0D0  *ALPHA*RAD1
TINVT( 5,22)=-2.0D0 *ALPHA*RAD1
TINVT( 5,23)=1.0D0  *ALPHA*RAD1
TINVT( 6, 9)=1.0D0  *L*ALPHA*RAD1
TINVT( 6,10)=-2.0D0 *L*ALPHA*RAD1
TINVT( 6,11)=1.0D0  *L*ALPHA*RAD1
TINVT( 6,13)=-2.0D0 *L*ALPHA*RAD1
TINVT( 6,14)=4.0D0  *L*ALPHA*RAD1
TINVT( 6,15)=-2.0D0 *L*ALPHA*RAD1
TINVT( 6,17)=1.0D0  *L*ALPHA*RAD1
TINVT( 6,18)=-2.0D0 *L*ALPHA*RAD1
TINVT( 6,19)=1.0D0  *L*ALPHA*RAD1
TINVT( 7, 1)=-1.0D0
TINVT( 7, 2)=1.0D0
TINVT( 8, 5)=-1.0D0
TINVT( 8, 6)=1.0D0
TINVT( 9, 9)=-4.0D0
TINVT( 9,10)=6.0D0
TINVT( 9,12)=-2.0D0
TINVT( 9,13)=6.0D0
TINVT( 9,14)=-9.0D0
TINVT( 9,16)=3.0D0
TINVT(10, 9)=2.0D0  *L
TINVT(10,10)=-3.0D0 *L
TINVT(10,12)=1.0D0  *L
TINVT(10,13)=-2.0D0 *L
TINVT(10,14)=3.0D0  *L
TINVT(10,16)=-1.0D0 *L
TINVT(11, 9)=-2.0D0 *ALPHA*RAD2
TINVT(11,10)=4.0D0  *ALPHA*RAD2
TINVT(11,11)=-2.0D0 *ALPHA*RAD2
TINVT(11,13)=3.0D0  *ALPHA*RAD2
TINVT(11,14)=-6.0D0 *ALPHA*RAD2
TINVT(11,15)=3.0D0  *ALPHA*RAD2
TINVT(12, 9)=1.0D0  *L*ALPHA*RAD2
TINVT(12,10)=-2.0D0 *L*ALPHA*RAD2
TINVT(12,11)=1.0D0  *L*ALPHA*RAD2
TINVT(12,13)=-1.0D0 *L*ALPHA*RAD2
TINVT(12,14)=2.0D0  *L*ALPHA*RAD2
TINVT(12,15)=-1.0D0 *L*ALPHA*RAD2
TINVT(13, 1)=1.0D0
TINVT(14, 5)=1.0D0
TINVT(15, 9)=4.0D0
TINVT(15,10)=-6.0D0
TINVT(15,13)=-6.0D0
TINVT(15,14)=9.0D0
TINVT(16, 9)=-2.0D0 *L
TINVT(16,10)=3.0D0  *L
TINVT(16,13)=2.0D0  *L
TINVT(16,14)=-3.0D0 *L
TINVT(17, 9)=-2.0D0 *ALPHA*RAD2
TINVT(17,10)=2.0D0  *ALPHA*RAD2
TINVT(17,13)=3.0D0  *ALPHA*RAD2
TINVT(17,14)=-3.0D0 *ALPHA*RAD2
TINVT(18, 9)=1.0D0  *L*ALPHA*RAD2
TINVT(18,10)=-1.0D0 *L*ALPHA*RAD2
TINVT(18,13)=-1.0D0 *L*ALPHA*RAD2
TINVT(18,14)=1.0D0  *L*ALPHA*RAD2
TINVT(19, 1)=-1.0D0
TINVT(19, 3)=1.0D0
TINVT(20, 5)=-1.0D0
TINVT(20, 7)=1.0D0
```

```
      TINVT(21, 9)=-4.0D0
      TINVT(21,10)=6.0D0
      TINVT(21,13)=6.0D0
      TINVT(21,14)=-9.0D0
      TINVT(21,21)=-2.0D0
      TINVT(21,22)=3.0D0
      TINVT(22, 9)=-2.0D0 *L
      TINVT(22,10)=3.0D0  *L
      TINVT(22,13)=4.0D0  *L
      TINVT(22,14)=-6.0D0 *L
      TINVT(22,17)=-2.0D0 *L
      TINVT(22,18)=3.0D0  *L
      TINVT(23, 9)=2.0D0  *ALPHA*RAD1
      TINVT(23,10)=-2.0D0 *ALPHA*RAD1
      TINVT(23,13)=-3.0D0 *ALPHA*RAD1
      TINVT(23,14)=3.0D0  *ALPHA*RAD1
      TINVT(23,21)=1.0D0  *ALPHA*RAD1
      TINVT(23,22)=-1.0D0 *ALPHA*RAD1
      TINVT(24, 9)=1.0D0  *L*ALPHA*RAD1
      TINVT(24,10)=-1.0D0 *L*ALPHA*RAD1
      TINVT(24,13)=-2.0D0 *L*ALPHA*RAD1
      TINVT(24,14)=2.0D0  *L*ALPHA*RAD1
      TINVT(24,17)=1.0D0  *L*ALPHA*RAD1
      TINVT(24,18)=-1.0D0 *L*ALPHA*RAD1
      DO 375 I=4,22,6
      DO 375 J=1,24
  375 TINVT(I,J)=-TINVT(I,J)
      DO 500 I=1,24
      DO 500 J=1,24
  500 TINV(I,J)=TINVT(J,I)
      RETURN
      END


      SUBROUTINE DEL1(DEL,X)
C   DEL1 FORMS THE CONSTITUTIVE LAW MATRIX.
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K1,K2,K3,L
      DIMENSION DEL(6,6)
      COMMON/B1/E,H1,H2,POIS,RAD1,RAD2,L,PHI1,PHI2,ALPHA,
     #THETA,KASE
      H(X)=H1+(H2-H1)*X
      K1=(H(X)*E)/(1.0D0-POIS**2)
      K2=K1*POIS
      K3=(K1*(1.0D0-POIS))/2.0D0
      D1=(H(X)**3*E)/(1.2D1*(1.0D0-POIS**2))
      D2=D1*POIS
      D3=(D1*(1.0D0-POIS))/2.0D0
      DO 100 I=1,6
      DO 100 J=1,6
  100 DEL(I,J)=0.0D0
      DEL(1,1)=K1
      DEL(1,2)=K2
      DEL(2,1)=K2
      DEL(2,2)=K1
      DEL(3,3)=K3
      DEL(4,4)=D1
      DEL(4,5)=D2
      DEL(5,4)=D2
      DEL(5,5)=D1
      DEL(6,6)=D3
      RETURN
      END


      FUNCTION P 101(X,Y)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K1,K2,K3,L
      COMMON/B1/E,H1,H2,POIS,RAD1,RAD2,L,PHI1,PHI2,ALPHA,
     #THETA,KASE
      P101      =Y/L
      RETURN
```

```
      ENTRY     P 102(X,Y)
      P102      =1.0D0/L
      RETURN
      ENTRY     P 109(X,Y)
      P109      =Q1(X   )*X**3*Y**3
      RETURN
      ENTRY     P 110(X,Y)
      P110      =Q1(X   )*X**3*Y**2
      RETURN
      ENTRY     P 111(X,Y)
      P111      =Q1(X   )*X**3*Y
      RETURN
      ENTRY     P 112(X,Y)
      P112      =Q1(X   )*X**3
      RETURN
      ENTRY     P 113(X,Y)
      P113      =Q1(X   )*X**2*Y**3
      RETURN
      ENTRY     P 114(X,Y)
      P114      =Q1(X   )*X**2*Y**2
      RETURN
      ENTRY     P 115(X,Y)
      P115      =Q1(X   )*X**2*Y
      RETURN
      ENTRY     P 116(X,Y)
      P116      =Q1(X   )*X**2
      RETURN
      ENTRY     P 117(X,Y)
      P117      =Q1(X   )*X*Y**3
      RETURN
      ENTRY     P 118(X,Y)
      P118      =Q1(X   )*X*Y**2
      RETURN
      ENTRY     P 119(X,Y)
      P119      =Q1(X   )*X*Y
      RETURN
      ENTRY     P 120(X,Y)
      P120      =Q1(X   )*X
      RETURN
      ENTRY     P 121(X,Y)
      P121      =Q1(X   )*Y**3
      RETURN
      ENTRY     P 122(X,Y)
      P122      =Q1(X   )*Y**2
      RETURN
      ENTRY     P 123(X,Y)
      P123      =Q1(X   )*Y
      RETURN
      ENTRY     P 124(X,Y)
      P124      =Q1(X   )
      RETURN
      END


      FUNCTION P 201(X,Y)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K1,K2,K3,L
      COMMON/B1/E,H1,H2,POIS,RAD1,RAD2,L,PHI1,PHI2,ALPHA,
     #THETA,KASE
      P201      =(COSPHI(X   )*Q(X   ))*X*Y
      RETURN
      ENTRY     P 202(X,Y)
      P202      =(COSPHI(X   )*Q(X   ))*X
      RETURN
      ENTRY     P 203(X,Y)
      P203      =(COSPHI(X   )*Q(X   ))*Y
      RETURN
      ENTRY     P 204(X,Y)
      P204      =(COSPHI(X   )*Q(X   ))
      RETURN
      ENTRY     P 205(X,Y)
      P205      =Q0(X   )*X
```

```
      RETURN
      ENTRY     P 207(X,Y)
      P207      =GQ(X   )
      RETURN
      ENTRY     P 209(X,Y)
      P209      =(SINPHI(X   )*Q(X   ))*X**3*Y**3
      RETURN
      ENTRY     P 210(X,Y)
      P210      =(SINPHI(X   )*Q(X   ))*X**3*Y**2
      RETURN
      ENTRY     P 211(X,Y)
      P211      =(SINPHI(X   )*Q(X   ))*X**3*Y
      RETURN
      ENTRY     P 212(X,Y)
      P212      =(SINPHI(X   )*Q(X   ))*X**3
      RETURN
      ENTRY     P 213(X,Y)
      P213      =(SINPHI(X   )*Q(X   ))*X**2*Y**3
      RETURN
      ENTRY     P 214(X,Y)
      P214      =(SINPHI(X   )*Q(X   ))*X**2*Y**2
      RETURN
      ENTRY     P 215(X,Y)
      P215      =(SINPHI(X   )*Q(X   ))*X**2*Y
      RETURN
      ENTRY     P 216(X,Y)
      P216      =(SINPHI(X   )*Q(X   ))*X**2
      RETURN
      ENTRY     P 217(X,Y)
      P217      =(SINPHI(X   )*Q(X   ))*X*Y**3
      RETURN
      ENTRY     P 218(X,Y)
      P218      =(SINPHI(X   )*Q(X   ))*X*Y**2
      RETURN
      ENTRY     P 219(X,Y)
      P219      =(SINPHI(X   )*Q(X   ))*X*Y
      RETURN
      ENTRY     P 220(X,Y)
      P220      =(SINPHI(X   )*Q(X   ))*X
      RETURN
      ENTRY     P 221(X,Y)
      P221      =(SINPHI(X   )*Q(X   ))*Y**3
      RETURN
      ENTRY     P 222(X,Y)
      P222      =(SINPHI(X   )*Q(X   ))*Y**2
      RETURN
      ENTRY     P 223(X,Y)
      P223      =(SINPHI(X   )*Q(X   ))*Y
      RETURN
      ENTRY     P 224(X,Y)
      P224      =(SINPHI(X   )*Q(X   ))
      RETURN
      END

      FUNCTION P 301(X,Y)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K1,K2,K3,L
      COMMON/B1/E,H1,H2,POIS,RAD1,RAD2,L,PHI1,PHI2,ALPHA,
     #THETA,KASE
      P301      =GQ(X   )*X
      RETURN
      ENTRY     P 303(X,Y)
      P303      =GQ(X   )
      RETURN
      ENTRY     P 305(X,Y)
      P305      =     Y/L-(Q(X   )/L)*DR(X   )*X*Y
      RETURN
      ENTRY     P 306(X,Y)
      P306      =1.0D0/L-(Q(X   )/L)*DR(X   )*X
      RETURN
      ENTRY     P 307(X,Y)
```

```
      P307      =(-Q(X   )/L)*DR(X   )*Y
      RETURN
      ENTRY     P 308(X,Y)
      P308      =(-Q(X   )/L)*DR(X   )
      RETURN
      END

      FUNCTION P 401(X,Y)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K1,K2,K3,L
      COMMON/B1/E,H1,H2,POIS,RAD1,RAD2,L,PHI1,PHI2,ALPHA,
     #THETA,KASE
      P401      =-(Q1(X   )**2/L)*DR1(X   )*X*Y+(Q1(X   )/L)*Y
      RETURN
      ENTRY     P 402(X,Y)
      P402      =-(Q1(X   )**2/L)*DR1(X   )*X   +(Q1(X   )/L)
      RETURN
      ENTRY     P 403(X,Y)
      P403      =-(Q1(X   )**2/L)*DR1(X   )*Y
      RETURN
      ENTRY     P 404(X,Y)
      P404      =-(Q1(X   )**2/L)*DR1(X   )
      RETURN
      ENTRY     P 409(X,Y)
      P409      =(-6.0D0/L**2)*X*Y**3
      RETURN
      ENTRY     P 410(X,Y)
      P410      =(-6.0D0/L**2)*X*Y**2
      RETURN
      ENTRY     P 411(X,Y)
      P411      =(-6.0D0/L**2)*X*Y
      RETURN
      ENTRY     P 412(X,Y)
      P412      =(-6.0D0/L**2)*X
      RETURN
      ENTRY     P 413(X,Y)
      P413      =(-2.0D0/L**2)*Y**3
      RETURN
      ENTRY     P 414(X,Y)
      P414      =(-2.0D0/L**2)*Y**2
      RETURN
      ENTRY     P 415(X,Y)
      P415      =(-2.0D0/L**2)*Y
      RETURN
      ENTRY     P 416(X,Y)
      P416      =(-2.0D0/L**2)
      RETURN
      ENTRY     P 490(X,Y)
      P490      =COSPHI(X   )*Q(X   )/L
      RETURN
      ENTRY     P 491(X,Y)
      P491      =QQ(X   )**2
      RETURN
      END

      FUNCTION P 501(X,Y)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K1,K2,K3,L
      COMMON/B1/E,H1,H2,POIS,RAD1,RAD2,L,PHI1,PHI2,ALPHA,
     #THETA,KASE
      P501      =COSPHI(X   )*Q(X   )*Q1(X   )*X*Y
      RETURN
      ENTRY     P 502(X,Y)
      P502      =COSPHI(X   )*Q(X   )*Q1(X   )*X
      RETURN
      ENTRY     P 503(X,Y)
      P503      =COSPHI(X   )*Q(X   )*Q1(X   )*Y
      RETURN
      ENTRY     P 504(X,Y)
      P504      =COSPHI(X   )*Q(X   )*Q1(X   )
```

```
      RETURN
      ENTRY      P 505(X,Y)
      P505       =Q(X    )*QQ(X    )*X
      RETURN
      ENTRY      P 507(X,Y)
      P507       =Q(X    )*QQ(X    )
      RETURN
      ENTRY      P 509(X,Y)
      P509       =-3.0D0*P490(X,Y)*X**2*Y**3-6.0D0*P491(X,Y)*
     #X**3*Y
      RETURN
      ENTRY      P 510(X,Y)
      P510       =-3.0D0*P490(X,Y)*X**2*Y**2-2.0D0*P491(X,Y)*
     #X**3
      RETURN
      ENTRY      P 511(X,Y)
      P511       =-3.0D0*P490(X,Y)*X**2*Y
      RETURN
      ENTRY      P 512(X,Y)
      P512       =-3.0D0*P490(X,Y)*X**2
      RETURN
      ENTRY      P 513(X,Y)
      P513       =-2.0D0*P490(X,Y)*X*Y**3-6.0D0*P491(X,Y)*X**2
     #*Y
      RETURN
      ENTRY      P 514(X,Y)
      P514       =-2.0D0*P490(X,Y)*X*Y**2-2.0D0*P491(X,Y)*X**2
      RETURN
      ENTRY      P 515(X,Y)
      P515       =-2.0D0*P490(X,Y)*X*Y
      RETURN
      ENTRY      P 516(X,Y)
      P516       =-2.0D0*P490(X,Y)*X
      RETURN
      ENTRY      P 517(X,Y)
      P517       =-P490(X,Y)*Y**3-6.0D0*P491(X,Y)*X*Y
      RETURN
      ENTRY      P 518(X,Y)
      P518       =-P490(X,Y)*Y**2-2.0D0*P491(X,Y)*X
      RETURN
      ENTRY      P 519(X,Y)
      P519       =-P490(X,Y)*Y
      RETURN
      ENTRY      P 520(X,Y)
      P520       =-P490(X,Y)
      RETURN
      ENTRY      P 521(X,Y)
      P521       =-6.0D0*P491(X,Y)*Y
      RETURN
      ENTRY      P 522(X,Y)
      P522       =-2.0D0*P491(X,Y)
      RETURN
      ENTRY      P 590(X,Y)
      P590       =(Q(X    )**2/L)*DR(X    )
      RETURN
      ENTRY      P 591(X,Y)
      P591       =(Q(X    )*QQ(X    )/L)*DR(X    )
      RETURN
      ENTRY      P 592(X,Y)
      P592       =QQ(X    )/L
      RETURN
      END


      FUNCTION P 601(X,Y)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K1,K2,K3,L
      COMMON/B1/E,H1,H2,POIS,RAD1,RAD2,L,PHI1,PHI2,ALPHA,
     #THETA,KASE
      P601       =(QQ(X    )*Q1(X    ))*X
      RETURN
      ENTRY      P 603(X,Y)
```

```
      P603      =QQ(X   )*Q1(X   )
      RETURN
      ENTRY     P 605(X,Y)
      P605      =-2.0D0*P590(X,Y)*X*Y+(Q(X   )/L)*Y
      RETURN
      ENTRY     P 606(X,Y)
      P606      =-2.0D0*P590(X,Y)*X   +(Q(X   )/L)
      RETURN
      ENTRY     P 607(X,Y)
      P607      = -2.0D0*P590(X,Y)*Y
      RETURN
      ENTRY     P 608(X,Y)
      P608      = -2.0D0*P590(X,Y)
      RETURN
      ENTRY     P 609(X,Y)
      P609      =6.0D0*P591(X,Y)*X**3*Y**2-18.0D0*P592(X,Y)*
     #X**2*Y**2
      RETURN
      ENTRY     P 610(X,Y)
      P610      =4.0D0*P591(X,Y)*X**3*Y    -12.0D0*P592(X,Y)*
     #X**2*Y
      RETURN
      ENTRY     P 611(X,Y)
      P611      =2.0D0*P591(X,Y)*X**3-6.0D0*P592(X,Y)*X**2
      RETURN
      ENTRY     P 613(X,Y)
      P613      =6.0D0*P591(X,Y)*X**2*Y**2-12.0D0*P592(X,Y)*
     #X*Y**2
      RETURN
      ENTRY     P 614(X,Y)
      P614      =4.0D0*P591(X,Y)*X**2*Y-8.0D0*P592(X,Y)*X*Y
      RETURN
      ENTRY     P 615(X,Y)
      P615      =2.0D0*P591(X,Y)*X**2          - 4.0D0*P592(X,Y)*X
      RETURN
      ENTRY     P 617(X,Y)
      P617      =6.0D0*P591(X,Y)*X*Y**2-6.0D0*P592(X,Y)*Y**2
      RETURN
      ENTRY     P 618(X,Y)
      P618      =4.0D0*P591(X,Y)*X    *Y    - 4.0D0*P592(X,Y)*Y
      RETURN
      ENTRY     P 619(X,Y)
      P619      =2.0D0*P591(X,Y)*X          - 2.0D0*P592(X,Y)
      RETURN
      ENTRY     P 621(X,Y)
      P621      =6.0D0*P591(X,Y)*Y**2
      RETURN
      ENTRY     P 622(X,Y)
      P622      =4.0D0*P591(X,Y)*Y
      RETURN
      ENTRY     P 623(X,Y)
      P623      =2.0D0*P591(X,Y)
      RETURN
      END

      SUBROUTINE PST(P,X,Y)
C   PST CONVERTS THE P'S FROM FUNCTIONS OF X AND Y TO VALUES
C   FOR ELEMENTS OF THE P MATRIX.
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K1,K2,K3,L
      DIMENSION P(6,24)
      P(1, 1)=P101(X,Y)
      P(1, 2)=P102(X,Y)
      P(1, 3)=0.0D0
      P(1, 4)=0.0D0
      P(1, 5)=0.0D0
      P(1, 6)=0.0D0
      P(1, 7)=0.0D0
      P(1, 8)=0.0D0
      P(1, 9)=P109(X,Y)
      P(1,10)=P110(X,Y)
```

```
      RETURN
      ENTRY      P 505(X,Y)
      P505       =Q(X   )*QQ(X   )*X
      RETURN
      ENTRY      P 507(X,Y)
      P507       =Q(X   )*QQ(X   )
      RETURN
      ENTRY      P 509(X,Y)
      P509       =-3.0D0*P490(X,Y)*X**2*Y**3-6.0D0*P491(X,Y)*
     #X**3*Y
      RETURN
      ENTRY      P 510(X,Y)
      P510       =-3.0D0*P490(X,Y)*X**2*Y**2-2.0D0*P491(X,Y)*
     #X**3
      RETURN
      ENTRY      P 511(X,Y)
      P511       =-3.0D0*P490(X,Y)*X**2*Y
      RETURN
      ENTRY      P 512(X,Y)
      P512       =-3.0D0*P490(X,Y)*X**2
      RETURN
      ENTRY      P 513(X,Y)
      P513       =-2.0D0*P490(X,Y)*X*Y**3-6.0D0*P491(X,Y)*X**2
     #*Y
      RETURN
      ENTRY      P 514(X,Y)
      P514       =-2.0D0*P490(X,Y)*X*Y**2-2.0D0*P491(X,Y)*X**2
      RETURN
      ENTRY      P 515(X,Y)
      P515       =-2.0D0*P490(X,Y)*X*Y
      RETURN
      ENTRY      P 516(X,Y)
      P516       =-2.0D0*P490(X,Y)*X
      RETURN
      ENTRY      P 517(X,Y)
      P517       =-P490(X,Y)*Y**3-6.0D0*P491(X,Y)*X*Y
      RETURN
      ENTRY      P 518(X,Y)
      P518       =-P490(X,Y)*Y**2-2.0D0*P491(X,Y)*X
      RETURN
      ENTRY      P 519(X,Y)
      P519       =-P490(X,Y)*Y
      RETURN
      ENTRY      P 520(X,Y)
      P520       =-P490(X,Y)
      RETURN
      ENTRY      P 521(X,Y)
      P521       =-6.0D0*P491(X,Y)*Y
      RETURN
      ENTRY      P 522(X,Y)
      P522       =-2.0D0*P491(X,Y)
      RETURN
      ENTRY      P 590(X,Y)
      P590       =(Q(X   )**2/L)*DR(X   )
      RETURN
      ENTRY      P 591(X,Y)
      P591       =(Q(X   )*QQ(X   )/L)*DR(X   )
      RETURN
      ENTRY      P 592(X,Y)
      P592       =QQ(X   )/L
      RETURN
      END


      FUNCTION P 601(X,Y)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K1,K2,K3,L
      COMMON/B1/E,H1,H2,POIS,RAD1,RAD2,L,PHI1,PHI2,ALPHA,
     #THETA,KASE
      P601       =(QQ(X   )*Q1(X   ))*X
      RETURN
      ENTRY      P 603(X,Y)
```

```
      P603      =GQ(X   )*Q1(X   )
      RETURN
      ENTRY     P 605(X,Y)
      P605      =-2.0D0*P590(X,Y)*X*Y+(Q(X   )/L)*Y
      RETURN
      ENTRY     P 606(X,Y)
      P606      =-2.0D0*P590(X,Y)*X   +(Q(X   )/L)
      RETURN
      ENTRY     P 607(X,Y)
      P607      = -2.0D0*P590(X,Y)*Y
      RETURN
      ENTRY     P 608(X,Y)
      P608      = -2.0D0*P590(X,Y)
      RETURN
      ENTRY     P 609(X,Y)
      P609      =6.0D0*P591(X,Y)*X**3*Y**2-18.0D0*P592(X,Y)*
     #X**2*Y**2
      RETURN
      ENTRY     P 610(X,Y)
      P610      =4.0D0*P591(X,Y)*X**3*Y    -12.0D0*P592(X,Y)*
     #X**2*Y
      RETURN
      ENTRY     P 611(X,Y)
      P611      =2.0D0*P591(X,Y)*X**3-6.0D0*P592(X,Y)*X**2
      RETURN
      ENTRY     P 613(X,Y)
      P613      =6.0D0*P591(X,Y)*X**2*Y**2-12.0D0*P592(X,Y)*
     #X*Y**2
      RETURN
      ENTRY     P 614(X,Y)
      P614      =4.0D0*P591(X,Y)*X**2*Y-8.0D0*P592(X,Y)*X*Y
      RETURN
      ENTRY     P 615(X,Y)
      P615      =2.0D0*P591(X,Y)*X**2       - 4.0D0*P592(X,Y)*X
      RETURN
      ENTRY     P 617(X,Y)
      P617      =6.0D0*P591(X,Y)*X*Y**2-6.0D0*P592(X,Y)*Y**2
      RETURN
      ENTRY     P 618(X,Y)
      P618      =4.0D0*P591(X,Y)*X    *Y    - 4.0D0*P592(X,Y)*Y
      RETURN
      ENTRY     P 619(X,Y)
      P619      =2.0D0*P591(X,Y)*X          - 2.0D0*P592(X,Y)
      RETURN
      ENTRY     P 621(X,Y)
      P621      =6.0D0*P591(X,Y)*Y**2
      RETURN
      ENTRY     P 622(X,Y)
      P622      =4.0D0*P591(X,Y)*Y
      RETURN
      ENTRY     P 623(X,Y)
      P623      =2.0D0*P591(X,Y)
      RETURN
      END

      SUBROUTINE PST(P,X,Y)
C   PST CONVERTS THE P'S FROM FUNCTIONS OF X AND Y TO VALUES
C   FOR ELEMENTS OF THE P MATRIX.
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K1,K2,K3,L
      DIMENSION P(6,24)
      P(1, 1)=P101(X,Y)
      P(1, 2)=P102(X,Y)
      P(1, 3)=0.0D0
      P(1, 4)=0.0D0
      P(1, 5)=0.0D0
      P(1, 6)=0.0D0
      P(1, 7)=0.0D0
      P(1, 8)=0.0D0
      P(1, 9)=P109(X,Y)
      P(1,10)=P110(X,Y)
```

```
P(1,11)=P111(X,Y)
P(1,12)=P112(X,Y)
P(1,13)=P113(X,Y)
P(1,14)=P114(X,Y)
P(1,15)=P115(X,Y)
P(1,16)=P116(X,Y)
P(1,17)=P117(X,Y)
P(1,18)=P118(X,Y)
P(1,19)=P119(X,Y)
P(1,20)=P120(X,Y)
P(1,21)=P121(X,Y)
P(1,22)=P122(X,Y)
P(1,23)=P123(X,Y)
P(1,24)=P124(X,Y)
P(2, 1)=P201(X,Y)
P(2, 2)=P202(X,Y)
P(2, 3)=P203(X,Y)
P(2, 4)=P204(X,Y)
P(2, 5)=P205(X,Y)
P(2, 6)=0.0D0
P(2, 7)=P207(X,Y)
P(2, 8)=0.0D0
P(2, 9)=P209(X,Y)
P(2,10)=P210(X,Y)
P(2,11)=P211(X,Y)
P(2,12)=P212(X,Y)
P(2,13)=P213(X,Y)
P(2,14)=P214(X,Y)
P(2,15)=P215(X,Y)
P(2,16)=P216(X,Y)
P(2,17)=P217(X,Y)
P(2,18)=P218(X,Y)
P(2,19)=P219(X,Y)
P(2,20)=P220(X,Y)
P(2,21)=P221(X,Y)
P(2,22)=P222(X,Y)
P(2,23)=P223(X,Y)
P(2,24)=P224(X,Y)
P(3, 1)=P301(X,Y)
P(3, 2)=0.0D0
P(3, 3)=P303(X,Y)
P(3, 4)=0.0D0
P(3, 5)=P305(X,Y)
P(3, 6)=P306(X,Y)
P(3, 7)=P307(X,Y)
P(3, 8)=P308(X,Y)
P(3, 9)=0.0D0
P(3,10)=0.0D0
P(3,11)=0.0D0
P(3,12)=0.0D0
P(3,13)=0.0D0
P(3,14)=0.0D0
P(3,15)=0.0D0
P(3,16)=0.0D0
P(3,17)=0.0D0
P(3,18)=0.0D0
P(3,19)=0.0D0
P(3,20)=0.0D0
P(3,21)=0.0D0
P(3,22)=0.0D0
P(3,23)=0.0D0
P(3,24)=0.0D0
P(4, 1)=P401(X,Y)
P(4, 2)=P402(X,Y)
P(4, 3)=P403(X,Y)
P(4, 4)=P404(X,Y)
P(4, 5)=0.0D0
P(4, 6)=0.0D0
P(4, 7)=0.0D0
P(4, 8)=0.0D0
P(4, 9)=P409(X,Y)
P(4,10)=P410(X,Y)
```

```
      P(4,11)=P411(X,Y)
      P(4,12)=P412(X,Y)
      P(4,13)=P413(X,Y)
      P(4,14)=P414(X,Y)
      P(4,15)=P415(X,Y)
      P(4,16)=P416(X,Y)
      P(4,17)=0.0D0
      P(4,18)=0.0D0
      P(4,19)=0.0D0
      P(4,20)=0.0D0
      P(4,21)=0.0D0
      P(4,22)=0.0D0
      P(4,23)=0.0D0
      P(4,24)=0.0D0
      P(5, 1)=P501(X,Y)
      P(5, 2)=P502(X,Y)
      P(5, 3)=P503(X,Y)
      P(5, 4)=P504(X,Y)
      P(5, 5)=P505(X,Y)
      P(5, 6)=0.0D0
      P(5, 7)=P507(X,Y)
      P(5, 8)=0.0D0
      P(5, 9)=P509(X,Y)
      P(5,10)=P510(X,Y)
      P(5,11)=P511(X,Y)
      P(5,12)=P512(X,Y)
      P(5,13)=P513(X,Y)
      P(5,14)=P514(X,Y)
      P(5,15)=P515(X,Y)
      P(5,16)=P516(X,Y)
      P(5,17)=P517(X,Y)
      P(5,18)=P518(X,Y)
      P(5,19)=P519(X,Y)
      P(5,20)=P520(X,Y)
      P(5,21)=P521(X,Y)
      P(5,22)=P522(X,Y)
      P(5,23)=0.0D0
      P(5,24)=0.0D0
      P(6, 1)=P601(X,Y)
      P(6, 2)=0.0D0
      P(6, 3)=P603(X,Y)
      P(6, 4)=0.0D0
      P(6, 5)=P605(X,Y)
      P(6, 6)=P606(X,Y)
      P(6, 7)=P607(X,Y)
      P(6, 8)=P608(X,Y)
      P(6, 9)=P609(X,Y)
      P(6,10)=P610(X,Y)
      P(6,11)=P611(X,Y)
      P(6,12)=0.0D0
      P(6,13)=P613(X,Y)
      P(6,14)=P614(X,Y)
      P(6,15)=P615(X,Y)
      P(6,16)=0.0D0
      P(6,17)=P617(X,Y)
      P(6,18)=P618(X,Y)
      P(6,19)=P619(X,Y)
      P(6,20)=0.0D0
      P(6,21)=P621(X,Y)
      P(6,22)=P622(X,Y)
      P(6,23)=P623(X,Y)
      P(6,24)=0.0D0
      RETURN
      END

      SUBROUTINE RIGID (STIFFF,STFMOD)
C  RIGID FORMS THE RIGID BODY AND THE K22(TKT) MATRIX.
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K1,K2,K3,L
      DIMENSION TTT(24,6),STFRB(24,24),AKT(24,6),TKTINV(6,6)
     #,RB(24,6),STIFFF(24,24),STFMOD(24,24),TKT(6,6),AK(36),
```

```
     #AKTT(6,24)
      COMMON/B1/E,H1,H2,POIS,RAD1,RAD2,L,PHI1,PHI2,ALPHA,
     #THETA,KASE
      COMMON/B2/CPHI1,CPHI2,C0,C1,C2
      EQUIVALENCE (TKT(1,1),AK(1))
      FZ(X)=L*DSQRT(1.0D0-(C0+C1*X+C2*X**2)**2)
      C0=CPHI1
      IF(KASE.GE.1) C1=0.0D0
      IF(KASE.GE.1) C2=0.0D0
      IF(KASE.GE.1) CPHI1=0.0D0
      IF(KASE.EQ.3) CPHI1=(RAD2-RAD1)/L
      IF(KASE.GE.1) CPHI2=CPHI1
      SPHI1=DSQRT(1.0D0-CPHI1**2)
      SPHI2=DSQRT(1.0D0-CPHI2**2)
C   NUMERICALLY EVALUATE FUNCTION Z.
      A=0.5D0
      C=.48014492824876812D0
      Z=.50614268145188130D-1*(FZ(A+C)+FZ(A-C))
      C=.39833323870681337D0
      Z=Z+.11119051722668724D0*(FZ(A+C)+FZ(A-C))
      C=.26276620495816449D0
      Z=Z+.15685332293894364D0*(FZ(A+C)+FZ(A-C))
      C=.917173212478249OD-1
      Z=Z+.18134189168918099D0*(FZ(A+C)+FZ(A-C))
      IF(KASE.EQ.1) Z=L
      IF(KASE.EQ.2) Z=L
      IF(KASE.EQ.3) Z=L*SPHI1
C   INITIALIZE RIGID BODY MATRIX AND K22(TKT) MATRIX.
      DO 1 I=1,6
      DO 1 J=1,6
    1 TKT(I,J)=0.0D0
      DO 2 I=1,24
      DO 2 J=1,6
    2 RB(I,J)=0.0D0
      RB( 1,1)=  CPHI1
      RB( 1,3)= -SPHI1
      RB( 1,5)=  Z*RB(1,1)-RAD1*RB(1,3)
      RB( 2,2)=  1.0D0
      RB( 2,4)= -Z
      RB( 2,6)=  RAD1
      RB( 3,1)=  SPHI1
      RB( 3,3)=  RB(1,1)
      RB( 3,5)=  Z*RB(3,1)-RAD1*RB(3,3)
      RB( 4,1)= -(( RB(1,1)/RB(3,1))*C1)/L
      RB( 4,3)=  C1/L
      RB( 4,5)= -RB(3,1)**2+Z*RB(4,1)-C0*RB(1,1)-(RAD1*C1)/L
      RB( 5,2)=  RB(3,1)/RAD1
      RB( 5,4)= -RB(3,5)/RAD1
      RB( 6,2)=  RB(4,1)/RAD1
      RB( 6,4)= -RB(4,5)/RAD1
      RB( 7,1)=  CPHI2
      RB( 7,3)= -SPHI2
      RB( 7,5)= -RB(7,3)*RAD2
      RB( 8,2)=  1.0D0
      RB( 8,6)=  RAD2
      RB( 9,1)=  SPHI2
      RB( 9,3)=  CPHI2
      RB( 9,5)= -RAD2*RB(9,3)
      RB(10,1)= -((RB(7,1)/RB(9,1))*(C1+2.0D0*C2))/L
      RB(10,3)=  (C1+2.0D0*C2)/L
      RB(10,5)= -RB(9,1)**2-(C0+C1+C2)*RB(9,3)-(RAD2*(C1+2.0
      RB(11,2)=  RB(9,1)/RAD2
      RB(11,4)=  RB(9,3)
      RB(12,2)=  RB(10,1)/RAD2
      RB(12,4)= -RB(10,5)/RAD2
      RB(13,1)=  RB(7,1)*DCOS(ALPHA)
      RB(13,2)=  RB(7,1)*DSIN(ALPHA)
      RB(13,3)=  RB(7,3)
      RB(13,4)=  RAD2*DSIN(ALPHA)*RB(7,3)
      RB(13,5)=  RAD2*RB(9,1)*DCOS(ALPHA)
      RB(14,1)= -DSIN(ALPHA)
      RB(14,2)=  DCOS(ALPHA)
```

```
      RB(14,6)=  RAD2
      RB(15,1)=  RB(9,1)*RB(14,2)
      RB(15,2)= -RB(9,1)*RB(14,1)
      RB(15,3)=  RB(7,1)
      RB(15,4)=  RB(13,2)*RAD2
      RB(15,5)= -RB(13,1)*RAD2
      RB(16,1)=  RB(10,1)*RB(14,2)
      RB(16,2)= -RB(10,1)*RB(14,1)
      RB(16,3)=  RB(10,3)
      RB(16,4)=  RB(14,1)*RB(10,5)
      RB(16,5)=  RB(14,2)*RB(10,5)
      RB(17,1)=  RB(9,1)*RB(14,1)/RAD2
      RB(17,2)=  RB(15,1)/RAD2
      RB(17,4)=  RB(13,1)
      RB(17,5)=  RB(13,2)
      RB(18,1)=  RB(10,1)*RB(14,1)/RAD2
      RB(18,2)=  RB(16,1)/RAD2
      RB(18,4)=  RB(14,2)*RB(12,4)
      RB(18,5)= -RB(14,1)*RB(12,4)
      RB(19,1)=  RB(1,1)*RB(14,2)
      RB(19,2)= -RB(3,3)*RB(14,1)
      RB(19,3)=  RB(1,3)
      RB(19,4)=  RB(14,1)*RB(1,5)
      RB(19,5)=  RB(14,2)*RB(1,5)
      RB(20,1)=  RB(14,1)
      RB(20,2)=  RB(14,2)
      RB(20,4)=  RB(2,4)*RB(14,2)
      RB(20,5)= -RB(2,4)*RB(14,1)
      RB(20,6)=  RB(2,6)
      RB(21,1)=  RB(3,1)*RB(14,2)
      RB(21,2)= -RB(3,1)*RB(14,1)
      RB(21,3)=  RB(1,1)
      RB(21,4)=  RB(14,1)*RB(3,5)
      RB(21,5)=  RB(14,2)*RB(3,5)
      RB(22,1)=  RB(4,1)*RB(14,2)
      RB(22,2)= -RB(4,1)*RB(14,1)
      RB(22,3)=  RB(4,3)
      RB(22,4)=  RB(14,1)*RB(4,5)
      RB(22,5)=  RB(14,2)*RB(4,5)
      RB(23,1)=  RB(5,2)*RB(14,1)
      RB(23,2)=  RB(5,2)*RB(14,2)
      RB(23,4)=  RB(14,2)*RB(5,4)
      RB(23,5)= -RB(14,1)*RB(5,4)
      RB(24,1)=  RB(4,1)*RB(14,1)/RAD1
      RB(24,2)=  RB(22,1)/RAD1
      RB(24,4)=  RB(14,2)*PB(6,4)
      RB(24,5)= -RB(14,1)*RB(6,4)
      DO 3 I=4,22,6
      DO 3 J=1,6
    3 RB(I,J)=-RB(I,J)
      IF(KASE.EQ.0) GO TO 5
      IF(KASE.GT.1) NDIM=5
      DO 4 I=1,24
      DO 4 J=1,3
      JL=2+J
      JR=3+J
    4 RB(I,JL)=RB(I,JR)
      IF(KASE.EQ.1) NDIM=4
      GO TO 7
    5 NDIM=6
    7 CONTINUE
      DO 10 I=1,24
      DO 10 J=1,NDIM
      AKT(I,J)=0.0D0
      DO 10 K=1,24
      AKT(I,J)=AKT(I,J)+STIFFF(I,K)*RB(K,J)
   10 CONTINUE
      DO 20 I=1,NDIM
      DO 20 J=1,NDIM
      TKT(I,J)=C.0D0
      DO 20 K=1,24
C   RB(K,I)=RB(I,K) TRANSPOSE
```

```
      TKT(I,J)=TKT(I,J)+RB(K,I)*AKT(K,J)
   20 CONTINUE
      IF(NDIM.EQ.6) GO TO 25
      DO 21 I=1,5
      IST=5*I+1
      DO 21 J=IST,35
      JP=J+1
   21 AK(J)=AK(JP)
      IF(NDIM.EQ.5) GO TO 25
      DO 22 I=1,5
      IST=4*I+1
      DO 22 J=IST,35
      JP=J+1
   22 AK(J)=AK(JP)
   25 CONTINUE
      CALL INVERT (NDIM,1.0D-12,TKT,TKTINV,KER,NDIM)
      IF(KER.EQ.2) WRITE(6,1000)
      CALL MULT(AKT,TKTINV,TTT,24,NDIM,NDIM)
      CALL MATT(AKT,AKTT,24,NDIM)
      CALL MULT(TTT,AKTT,STFRB,24,NDIM,24)
      DO 50 I=1,24
      DO 50 J=1,24
      STFMOD(I,J)=0.0D0
      STFMOD(I,J)=STIFFF(I,J)-STFRB(I,J)
   50 CONTINUE
      DO 60 I=1,24
      DO 60 J=1,24
      STFMOD(I,J)=(STFMOD(I,J)+STFMOD(J,I))*0.5D0
      STFMOD(J,I)=STFMOD(I,J)
   60 CONTINUE
 1000 FORMAT (/,23H MATRIX K22 IS SINGULAR)
      RETURN
      END

      SUBROUTINE MATT(A,AT,N1,N2)
C   MATT IS AN ONLINE MATRIX TRANSPOSE ROUTINE.
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(N1,N2),AT(N2,N1)
      DO 100 I=1,N2
      DO 100 J=1,N1
  100 AT(I,J)=A(J,I)
      RETURN
      END

       SUBROUTINE INVERT(N,EP,A,X,KER,NACT)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(NACT,NACT),X(NACT,NACT)
      DO 1 I=1,N
      DO 1 J=1,N
    1 X(I,J)=0.0D0
      DO 2 K=1,N
    2 X(K,K)=1.0D0
   10 DO 34 L=1,N
      KP=0
      Z=0.0D0
      DO 12 K=L,N
      IF(Z-DABS(A(K,L)))11,12,12
   11 Z=DABS(A(K,L))
      KP=K
   12 CONTINUE
      IF(L-KP)13,20,20
   13 DO 14 J=L,N
      Z=A(L,J)
       A(L,J)=A(KP,J)
   14 A(KP,J)=Z
      DO 15 J=1,N
      Z=X(L,J)
      X(L,J)=X(KP,J)
   15 X(KP,J)=Z
   20 IF(DABS(A(L,L))-EP)50,50,30
```

```
   30 IF(L-N)31,34,34
   31 LP1=L+1
      DO 36 K=LP1,N
      IF(A(K,L))32,36,32
   32 RATIO=A(K,L)/A(L,L)
      DO 33 J=LP1,N
   33 A(K,J)=A(K,J)-RATIO*A(L,J)
      DO 35 J=1,N
   35 X(K,J)=X(K,J)-RATIO*X(L,J)
   36 CONTINUE
   34 CONTINUE
   40 DO 43 I=1,N
      II=N+1-I
      DO 43 J=1,N
      S=0.0D0
      IF(II-N)41,43,43
   41 IIP1=II+1
      DO 42 K=IIP1,N
   42 S=S+A(II,K)*X(K,J)
   43 X(II,J)=(X(II,J)-S)/A(II,II)
      KER=1
      GO TO 75
   50 KER=2
   75 CONTINUE
      RETURN
      END


      SUBROUTINE MULT(A,B,R,N,M,L)
C   MULT IS AN ONLINE MATRIX MULTIPLICATION ROUTINE.
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(N,M),B(M,L),R(N,L)
      DO 20 I=1,N
      DO 20 J=1,L
      R(I,J)=0.0D0
      DO 10 K=1,M
      R(I,J)=R(I,J)+A(I,K)*B(K,J)
   10 CONTINUE
   20 CONTINUE
      RETURN
      END


      SUBROUTINE FLAG(NO,*,*)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K1,K2,K3,L
      COMMON/B1/E,H1,H2,POIS,RAD1,RAD2,L,PHI1,PHI2,ALPHA,
     #THETA,KASE
      GO TO (1,2,3,4,5,6),NO
    1 WRITE (6,1000)
 1000 FORMAT (/,5X,'DATA INCONSISTENT, ARC LENGTH LESS THAN'
     #,' DIFFERENCE IN RADII')
      RETURN 1
    2 WRITE (6,2000)
 2000 FORMAT (/,5X,'DATA INCONSISTENT, COSPHI IS CALCULATED'
     #,' TO BE GREATER THAN ONE')
      RETURN 1
    3 WRITE (6,3000)
 3000 FORMAT (/,5X,'ELEMENT IS SPECIFIED A CYLINDER')
      RETURN 2
    4 WRITE (6,4000)
      WIDTH=1.0D0/THETA
      WRITE (6,4100) L,WIDTH
 4000 FORMAT (/,5X,'ELEMENT IS SPECIFIED A FLAT PLATE')
 4100 FORMAT (/,8H LENGTH=,D15.8,7H WIDTH=,D15.8)
      RETURN 2
    5 WRITE (6,5000)
 5000 FORMAT (/,5X,'ELEMENT IS SPECIFIED A TRUNCATED CONE')
      RETURN 2
    6 WRITE (6,6000)
 6000 FORMAT (/,' SIN PHI1 OR SIN PHI2 IS EQUAL TO ZERO,',
     #' RIGID BODY MATRIX WILL HAVE SINGULAR POINTS,',/,
```

```
     #' ELEMENT STIFFNESS MATRIX DOES NOT CONTAIN RIGID ',
     #'BODY MOTION, RELOCATE NODE POINTS ')
      RETURN 2
      END


      SUBROUTINE STRESS(TINV,SS,SN)
C   STRESS FORMS ELEMENT STRESS AND STRAIN RELATIONSHIPS .
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K1,K2,K3,L
      DIMENSION SS(24,24),SN(24,24),TINV(24,24),W1(24,24),
     #P(6,24),CORNER(4,2),DEL(6,6)
      DATA CORNER/0.0D0,2*1.0D0,3*0.0D0,2*1.0D0/
C   INITIALIZE W1 MATRIX
      DO 50 I=1,24
      DO 50 J=1,24
   50 W1(I,J)=0.0D0
C   CALCULATE STRAIN RELATIONSHIP, SN IS STRAIN MATRIX
      DO 100 I=1,4
      X=CORNER(I,1)
      Y=CORNER(I,2)
      CALL PST(P,X,Y)
      II=6*(I-1)
      DO 100 J=1,6
      JJ=II+J
      DO 100 K=1,24
  100 W1(JJ,K)=P(J,K)
      CALL MULT(W1,TINV,SN,24,24,24)
C   RESET W1 MATRIX TO ZERO
      DO 200 I=1,24
      DO 200 J=1,24
  200 W1(I,J)=0.0D0
C   CALCULATE STRESS RELATIONSHIP, SS IS STRESS MATRIX
      DO 300 I=1,4
      X=CORNER(I,1)
      CALL DEL1(DEL,X)
      II=6*(I-1)
      DO 300 J=1,6
      JJ=II+J
      DO 300 K=1,6
      KK=II+K
  300 W1(JJ,KK)=DEL(J,K)
      CALL MULT(W1,SN,SS,24,24,24)
      RETURN
      END


      SUBROUTINE MERGER (TOTK)
C   MERGE ARRANGES INDIVIDUAL ELEMENT STIFFNESS MATRICES ON
C   ON THE UPPER HALFBAND OF THE TOTAL STIFFNESS MATRIX AND
C   PLACES APPROPRIATE BOUNDARY VALUES IN THE LOAD VECTOR.
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION AK(24,24),LM(4)
      COMMON/MBN/BLOAD(294),NBC(50,7),NCON(36,5)
      COMMON/PARM/NEL,NELT,NDPT,NPBC,NBAND,NEQ
      DIMENSION TOTK(294,54)
      DO 400 N=1,NELT
      NTK=(NCON(N,5)-1)*4+1
      CALL RDDISK(NTK,AK,576)
      LM(1)=NCON(N,1)*6-6
      LM(2)=NCON(N,2)*6-6
      LM(3)=NCON(N,3)*6-6
      LM(4)=NCON(N,4)*6-6
      DO 400 I=1,4
      DO 400 J=1,4
      DO 400 K=1,6
      DO 400 LL=1,6
      II=LM(I)+K
      JJ=LM(J)+LL-LM(I)-K+1
      IF(JJ) 400,400,300
  300 CONTINUE
      KK=6*I-6+K
```

```
      LLL=6*J-6+LL
C   AK IS ELEMENT STIFFNESS MATRIX
C   TOTK IS THE TOTAL STIFFNES MATRIX
      TOTK(II,JJ)=TOTK(II,JJ)+AK(KK,LLL)
  400 CONTINUE
      DO 800 I=1,NPBC
      IPT=(NBC(I,1)-1)*6
      DO 700 J=1,6
      IAD=IPT+J
      IF(NBC(I,J+1).EQ.1) GO TO 700
      DO 500 K=1,NBAND
  500 TOTK(IAD,K)=0.0D0
      TOTK(IAD,1)=1.0D0
      BLOAD(IAD)=0.0D0
      II=0
      LLU=NBAND-1
      DO 600 LL=1,LLU
      LLL=LL+1
      II=IAD-LL
      IF(II.LE.0) GO TO 600
      TOTK(II,LLL)=0.0D0
  600 CONTINUE
  700 CONTINUE
  800 CONTINUE
      RETURN
      END


      SUBROUTINE BANSOL(A,B,NN,MM)
C   BANSOL SOLVES THE TOTAL FORCE-DISPLACEMENT EQUATIONS.
      IMPLICIT REAL*8 (A-H,O-Z)
C         NN IS THE NUMBER OF EQUATIONS
C         MM IS THE HALF BAND WIDTH
      DIMENSION A(294,54),B(294),C(54)
      N=0
  100 N=N+1
      B(N)=B(N)/A(N,1)
      IF(N-NN) 150,300,150
  150 DO 200 K=2,MM
      C(K)=A(N,K)
  200 A(N,K)=A(N,K)/A(N,1)
      DO 260 L=2,MM
      I=N+L-1
      IF(NN-I) 260,240,240
  240 J=0
      DO 250 K=L,MM
      J=J+1
  250 A(I,J)=A(I,J)-C(L)*A(N,K)
      B(I)=B(I)-C(L)*B(N)
  260 CONTINUE
      GO TO 100
  300 N=N-1
      IF(N) 350,500,350
  350 DO 400 K=2,MM
      L=N+K-1
      IF(NN-L) 400,370,370
  370 B(N)=B(N)-A(N,K)*B(L)
  400 CONTINUE
      GO TO 300
  500 RETURN
      END


      SUBROUTINE PLOAD(P,PVEC)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 L
      DIMENSION PVEC(24),QVEC(24),XG(8),WT(8),COEF(4),F(4),
     #TINV(24,24),TINVT(24,24)
      COMMON/B1/E,H1,H2,POIS,RAD1,RAD2,L,PHI1,PHI2,ALPHA,
     #THETA,KASE
C   INTEGRATE TERMS OF THE PLOAD VECTOR.
C ZEROS OF GAUSSIAN 8 POINT FORMULAS ARE
```

```
      XX1=.96028985649753623D0
      XX2=.79666647741362674D0
      XX3=.52553240991632899D0
      XX4=.18343464249564980D0
C   COEFFICIENTS OF ZEROS ARE
      AA1=.10122853629037626D0
      AA2=.22238103445337447D0
      AA3=.31370664587788729D0
      AA4=.36268378337836198D0
C   XG CONTAINS ALL THE 8 ZEROS AND WT CONTAINS ALL THE
C   COEFFICIENTS .  ALL ARE SHIFTED TO THE RANGE 0 TO 1.
      WT(1)=AA1*0.5D0
      WT(2)=AA2*0.5D0
      WT(3)=AA3*0.5D0
      WT(4)=AA4*0.5D0
      WT(5)=WT(4)
      WT(6)=WT(3)
      WT(7)=WT(2)
      WT(8)=WT(1)
      A=0.5D0
      XG(1)=A+XX1*0.5D0
      XG(2)=A+XX2*0.5D0
      XG(3)=A+XX3*0.5D0
      XG(4)=A+XX4*0.5D0
      XG(5)=A-XX4*0.5D0
      XG(6)=A-XX3*0.5D0
      XG(7)=A-XX2*0.5D0
      XG(8)=A-XX1*0.5D0
      DO 100 I=1,4
  100 F(I)=0.0D0
      DO 200 I=1,8
      QVEC(I)=0.0D0
      X=XG(I)
      WTW=WT(I)
      F(1)=F(1)+WTW*R(X)*X**3
      F(2)=F(2)+WTW*R(X)*X**2
      F(3)=F(3)+WTW*R(X)*X
      F(4)=F(4)+WTW*R(X)
  200 CONTINUE
C   CALCULATE COEFFICIENTS FOR INTERGRALS
      DO 300 I=1,4
  300 COEF(I)=P*L*ALPHA/(5.0D0-I)
C   LOAD TERMS OF QLOAD VECTOR
      DO 400 I=1,4
      DO 400 J=1,4
      K=(J+1)*4+I
      QVEC(K)=COEF(I)*F(J)
  400 CONTINUE
C   TRANSFORM QLOAD VECTOR TO GET PLOAD VECTOR
      CALL TFORM(TINV,TINVT)
      CALL MULT(TINVT,QVEC,PVEC,24,24,1)
      RETURN
      END


      SUBROUTINE WRDISK(NTRACK, A,NCT)
C   WRDISK/RDDISK READS AND WRITES ELEMENT INFORMATION ON/OFF
C   DISK.
      INTEGER LAST/0/
      REAL*8 NAME(2)/'WRDISK','RDDISK'/
C   REMOVE THE FOLLOWING CARD WHEN USING SINGLE PRECISION
      REAL*8 A
      DIMENSION A(1)
      DEFINE FILE 7(1500,368,E,I)
C   REPLACE 368 WITH 184 FOR SINGLE PRECISION
      IF(NTRACK .GT. 1999) GO TO 900
      IF(NTRACK .LT. 0) NTRACK =(LAST+9)/10
      N = NTRACK
C   THE MAXIMUM NUMBER OF WORDS IN A OR B MUST BE .LE. NRL*46
      NRL=13
      N=N*NRL+1
      IF (NCT.GT. 46)GO TO 50
```

```
      WRITE (7'N,1000) (A(J),J=1,NCT)
      IF (LAST .LT. I) LAST = I
      IF(LAST.GT. 19999)LAST=0
      RETURN
   50 JI = 47
      WRITE (7'N,1000) (A(J),J=1,46)
   75 JE = JI + 45
      IF ( JE .GE. NCT) GO TO 100
      WRITE (7'I,1000) (A(J),J=JI,JE)
      JI = JI+46
      GO TO 75
  100 WRITE (7'I,1000) (A(J),J=JI,NCT)
      IF(I.GT.LAST) LAST=I
      IF(LAST.GT. 19999)LAST=0
      RETURN
      ENTRY RDDISK(NTRACK,B,NCT)
C  REMOVE THE FOLLOWING CARD WHEN USING SINGLE PRECISION
      REAL*8 B
      DIMENSION B(1)
      NRL = 13
      IF(NTRACK .GT. 1999) GO TO 910
      N=NTRACK
      N=N*NPL+1
      IF (NCT .GT.46) GO TO 150
      READ (7'N,1000) (B(J), J=1,NCT)
      RETURN
  150 READ (7'N,1000) (B(J),J=1,46)
      JI = 47
  175 JE = JI + 45
      IF (JE .GE. NCT) GO TO 200
      READ  (7'I,1000) (B(J),J=JI,JE)
      JI = JI + 46
      GO TO 175
  200 READ  (7'I,1000) (B(J),J=JI,NCT)
      RETURN
 1000 FORMAT (46A8)
C  REPLACE FORMAT WITH (46A4) FOR SINGLE PRECISION
  900 K=1
      GO TO 920
  910 K=2
  920 WRITE(6,1920)NAME(K),NTRACK
 1920 FORMAT('0 ERROR IN CALL OF ',A6,', NTRACK TOO LARGE,(M
     1THAN 2000)' / '  NTRACK =',I10)
      STOP
      END


      SUBROUTINE FORML
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K1,K2,K3,L
      DIMENSION PVEC(24),LM(4)
      COMMON/MBN/BLOAD(294),NBC(50,7),NCON(36,5)
      COMMON/PARM/NEL,NELT,NDPT,NPBC,NBAND,NEQ
      DO 200 I=1,NELT
      NTK=NCON(I,5)*4
      CALL RDDISK(NTK,PVEC,24)
      LM(1)=6*NCON(I,1)-6
      LM(2)=6*NCON(I,2)-6
      LM(3)=6*NCON(I,3)-6
      LM(4)=6*NCON(I,4)-6
      DO 100 K=1,4
      II=LM(K)
      IP=6*(K-1)
      DO 100 J=1,6
      IB=II+J
      IIP=IP+J
  100 BLOAD(IB)=BLOAD(IB)+PVEC(IIP)
  200 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PSSN
C   THIS SUBROUTINE RECOVERS ELEMENT STRESS AND STRAIN VALUES
C   FROM DISK AND ARRANGES THEM IN RESULTANT  STRESS AND
C   STRAIN MATRICES FOR THE NODE POINTS.
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/MBN/BLOAD(294),NBC(50,7),NCON(36,5)
      COMMON/PARM/NEL,NELT,NDPT,NPBC,NBAND,NEQ
      COMMON/SS SN/SS(24,24),SN(24,24)
      DIMENSION UEL(24),SNL(24),SSL(24),RESN(49,7),
     #RESS(49,6),LM(4)
C   INITIALIZE RESULTANT STRAIN AND RESULTANT STRESS MATRICES
      DO 100 I=1,49
      DO 100 J=1,6
      RESN(I,J)=0.0D0
  100 RESS(I,J)=0.0D0
      DO 110 I=1,49
  110 RESN(I,7)=0.0D0
      DO 400 I=1,NELT
      NTK=4*NCON(I,5)-2
      CALL RDDISK (NTK,SN,576)
      NTK=NTK+1
      CALL RDDISK (NTK,SS,576)
      LM(1)=6*NCON(I,1)-6
      LM(2)=6*NCON(I,2)-6
      LM(3)=6*NCON(I,3)-6
      LM(4)=6*NCON(I,4)-6
      DO 200 J=1,4
      JAD=LM(J)
      IUEL=(J-1)*6
      DO 200 K=1,6
      KJA=JAD+K
      IU=IUEL+K
  200 UEL(IU)=BLOAD(KJA)
      DO 210 III=1,24
      SNL(III)=0.0D0
      DO 210 JJJ=1,24
  210 SNL(III)=SNL(III)+SN(III,JJJ)*UEL(JJJ)
      DO 220 III=1,24
      SSL(III)=0.0D0
      DO 220 JJJ=1,24
  220 SSL(III)=SSL(III)+SS(III,JJJ)*UEL(JJJ)
      DO 300 II=1,4
      KJA=NCON(I,II)
      IUEL=(II-1)*6
      DO 250 JJ=1,6
      IU=IUEL+JJ
      RESN(KJA,JJ)=RESN(KJA,JJ)+SNL(IU)
  250 RESS(KJA,JJ)=RESS(KJA,JJ)+SSL(IU)
  300 RESN(KJA,7)=RESN(KJA,7)+1.0D0
  400 CONTINUE
      DO 500 I=1,NDPT
      DO 500 J=1,6
      RESN(I,J)=RESN(I,J)/RESN(I,7)
  500 RESS(I,J)=RESS(I,J)/RESN(I,7)
      WRITE (6,1000)
 1000 FORMAT(1H1,///,27X,'STRAINS')
      WRITE (6,1500)
 1500 FORMAT(//,27X,'JOINT',5X,'SNX',9X,'SNY',9X,'SNXY',8X,
     #'SMX',9X,'SMY',9X,'SMXY')
      DO 600 I=1,NDPT
  600 WRITE (6,2000) I,(RESN(I,J),J=1,6)
 2000 FORMAT(/,19X,I12,1P6D12.4)
      WRITE (6,2500)
 2500 FORMAT(1H1,///,27X,'STRESSES')
      WRITE (6,3000)
 3000 FORMAT(//,27X,'JOINT',6X,'NX',10X,'NY',10X,'NXY',
     #9X,'MX',10X,'MY',10X,'MXY')
      DO 700 I=1,NDPT
```

```
  700 WRITE (6,2000) I,(RESS(I,J),J=1,6)
      RETURN
      END

      SUBROUTINE OUTPUT
C   OUTPUT WRITES NODAL DISPLACEMENTS.
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K1,K2,K3,L
      COMMON/MBN/BLOAD(294),NBC(50,7),NCON(36,5)
      COMMON/PARM/NEL,NELT,NDPT,NPBC,NBAND,NEQ
      WRITE (6,1000)
 1000 FORMAT(1H1,///,27X,'DISPLACEMENTS')
      WRITE (6,1500)
 1500 FORMAT(//,27X,'JOINT'6X,'U',11X,'V',11X,'W',10X,'W,X',
     #9X,'W,Y',9X,'W,XY')
      DO 100 I=1,NDPT
      IL=6*(I-1)+1
      IU=IL+5
  100 WRITE (6,2000) I,(BLOAD(J),J=IL,IU)
 2000 FORMAT(/,19X,I12,1P6D12.4)
      RETURN
      END
```

REFERENCES

1. Johnson, C. P., "The Analysis of Thin Shells by a Finite Element Procedure," Ph. D. Thesis, University of California, Berkeley, 1967.

2. Carr, A. J., "A Refined Finite Element Analysis of Thin Shell Structures Including Dynamic Loading," Ph. D. Thesis, University of California, Berkeley, 1967.

3. Jones, R. E. and Strome, D. R., "Direct Stiffness Method Analysis of Shells of Revolution Utilizing Curved Elements, "AIAA Journal, v. 4, pp. 1519-1525, 1966.

4. Cantin, G., "A Curved Finite Element for Cylindrical Shells," Ph. D. Thesis, University of California, Berkeley, 1968.

5. Zienkiewicz, O. C., Stress Analysis, p. 90, Wiley, 1965.

6. Cantin, G., "Rigid Body Motions in Curved Finite Elements," paper received via private communication.

7. Cantin, G. and Clough, R. W., "A Curved Cylindrical Shell Finite Element," AIAA Journal, v. 6, pp. 1057 - 1062, 1968.

8. Kraus, H., Thin Elastic Shells, p. 198, Wiley, 1967.

9. Stroud, A. H. and Secrest, D., Gaussian Quadrature Formulas, p. 7, Prentice-Hall, 1966.

10. Bogner, F. K., Fox, R. L., Schmit, L. A., "The Generation of Inter-element Compatible Stiffness and Mass Matrices By the Use of Interpolation Formulas," paper presented at a conference of matrix methods, AFIT, Ohio, 1965.

11. Bogner, F. K., Fox, R. L., and Schmit, L. A., "A Cylindrical Shell Discrete Element," AIAA Journal, v. 5, pp. 745-750, 1967.

12. Timoshenko, S. and Woinowsky-Krieger, S., Theory of Plates and Shells, pp. 551-554, McGraw-Hill, 1959.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	20
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Naval Ship Systems Command (Code 2052) Navy Department Washington, D. C. 20360	1
4.	Mechanical Engineering Department, Code 59 Naval Postgraduate School Monterey, California 93940	2
5.	Professor Gilles Cantin, Code 59Ci Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	6
6.	LT Dean W. Kiess, USN USS DOLPHIN (AGSS 555) c/o Fleet Post Office New York, New York 09501	1

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)	2a. REPORT SECURITY CLASSIFICATION
Naval Postgraduate School Monterey, California 93940	Unclassified
	2b. GROUP

3. REPORT TITLE

A Curvilinear Shell Finite Element

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Mechanical Engineer and Master of Science in Mechanical Engineering; December 1969

5. AUTHOR(S) (First name, middle initial, last name)

Dean William Kiess

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
December 1969	85	12

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

A doubly curved element for a shell of revolution which has arbitrarily curved meridians is developed and analyzed. Meridional curvature is calculated using a highly accurate polynomial approximation. The displacement functions selected satisfy interelement compatibility and contain all the lower modes of a complete set of straining functions. Non-straining modes corresponding to rigid body motions are introduced into the final stiffness matrix for any conceivable rigid body motions.

The direct stiffness method was used to construct a stress and strain analysis program and the results of the analysis of a few problems are compared to classical solutions to establish the integrity of the element

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

Security Classification

A-31408

Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Curvilinear Shell Finite Element						
Stress-Strain Analysis of Curved Shells						

DD FORM 1 NOV 65 1473 (BACK)

S/N 0101-807-6821

Security Classification

A-3140