To the Graduate Council:

I am submitting herewith a thesis written by Brian Graham Smith entitled "Analysis of Discontinuity Forces for Axisymmetric Shells of Revolution." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Engineering Science.

Jerry E. Stoneking, Major Professor

We have read this thesis and recommend its acceptance:

Thomas C. Carley

R. L. Woodward

Accepted for the Council:

Lew Minkel

Vice Provost and
Dean of the Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master of Science degree at the University of Tennessee, Knoxville, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgment of source is made.

Requests for permission for extended quotation from the reproduction of this thesis in whole or in parts may be granted by the copyright holder.

Signature Brian G. Smith

Date 5-22-85

ANALYSIS OF DISCONTINUITY FORCES FOR AXISYMMETRIC SHELLS OF REVOLUTION

A Thesis
Presented for the
Master of Science
Degree
University of Tennessee, Knoxville

Brian Graham Smith
June 1985

Copyright© Brian Graham Smith, 1985
All rights reserved

ACKNOWLEDGMENT

The author would like to thank Dr. Jerry E. Stoneking for suggesting this topic and for providing financial support as a graduate teaching assistant throughout portions of this project. The author greatly appreciates the efforts of Phyllis M. Smith for typing this report. Her enthusiasm and smile provided a great deal of inspiration needed to complete this project.

Finally, the author would like to thank his parents, Howard and Patricia, for their love and encouragement throughout life.

ABSTRACT

This report presents the development of a computer program, ALSHEL, which calculates the discontinuity stress resultants in axisymmetric shells of revolution of various geometries and loading configurations. Also included is program GRAPH which displays the effects of the discontinuity stress resultants as a function of meridional lengths.

TABLE OF CONTENTS

CHAPTER		PAGE
I.	INTRODUCTION	1
II.	THEORETICAL BACKGROUND	3
	Force Method--Matrix Formulation . . .	3
	Equations of Equilibrium	10
	Influence Coefficients	18
III.	PROGRAM ORGANIZATION	36
	Subroutine Functions	36
	Program Usage	38
	Example Problems	40
IV.	CONCLUSION	55
BIBLIOGRAPHY	. . . . .	57
APPENDICES	. . . . .	58
APPENDIX A:	EQUATIONS FOR PROGRAM GRAPH	59
APPENDIX B:	THOMPSON FUNCTIONS	73
APPENDIX C:	NUMERICAL INTEGRATION FOR ELLIPTIC INTEGRALS	80
APPENDIX D:	FORTRAN LISTING OF ALSHEL AND GRAPH . .	83
VITA	. . . . .	155

LIST OF FIGURES

FIGURE		PAGE
2-1.	Containing vessel submitted to positive generalized discontinuity stress resultants	6
2-2.	Primary edge deformations	6
2-3.	Free body diagram of a differential shell element with curvature in two principal planes	12
2-4.	Free body diagram of a differential cylindrical shell element	17
2-5.	Notation for conical shells	23
2-6.	Notation for hemispherical shells	27
2-7.	Notation for ellipsoidal shells	30
2-8.	Free body diagram of a stiffening ring	31
2-9.	Cross section of a stiffening ring subjected to edge discontinuity forces	33
2-10.	Flat circular plate subjected to edge moments	35
3-1.	Pressure vessel comprised of various shell geometries	41
3-2.	Free body diagram of example one	44
3-3.	Radial deflection of the open crown hemisphere due to discontinuity forces	45
3-4.	Meridional rotation of the open crown hemisphere due to discontinuity forces	45
3-5.	Shear stress resultant caused by discontinuity forces	46
3-6.	Meridional membrane stress resultant caused by discontinuity forces	46
3-7.	Tangential membrane stress resultant due to discontinuity forces	47
3-8.	Meridional moment due to discontinuity forces	47
3-9.	Tangential moment of the open crown hemisphere caused by discontinuity forces	48

FIGURE PAGE

3-10. Radial deflection of the small cylinder caused by discontinuity forces 48

3-11. Meridional rotation of the small cylinder due to discontinuity forces 49

3-12. Transverse shear stress resultant of the small cylinder caused by discontinuity forces 49

3-13. Tangential membrane stress resultant for the small cylinder caused by discontinuity forces 50

3-14. Meridional moment for the small cylinder due to discontinuity forces 50

3-15. Tangential moment for the small cylinder due to discontinuity forces 51

3-16. Storage tank for example two 53

3-17. Idealized cylinder-ring-dome junction 53

A-1. Circular cylinder with positive discontinuity forces . . 62

A-2. Open crown hemisphere subjected to positive discontinuity forces 66

A-3. Conical shell subjected to positive discontinuity forces 70

B-1. Characteristics of ber(y) and bei(y) 78

B-2. Characteristics of ber'(y) and bei'(y) 78

B-3. Characteristics of ker(y) and kei(y) 79

B-4. Characteristics of ker'(y) and kei'(y) 79

LIST OF SYMBOLS

Δ	Deflection of a shell meridian
β	Rotation of a shell meridian
M_θ, M_ϕ	Bending moments in a shell per unit length of meridional section and a section perpendicular to the meridian, respectively
Mx, M_θ	Bending moments in a shell per unit length of axial section and a section perpendicular to the axis of a cylindrical or conical shell respectively
M_1	Discontinuity shear per unit length of a section perpendicular to the meridian
H_1	Discontinuity shear per unit length of a section perpendicular to the meridian
$M_{\theta\phi}$,$M_{\phi\theta}$	Twisting moments in a shell per unit length of meridional section and a section perpendicular to the meridian, respectively
N_θ,$N_{\theta\phi}$,$N_\phi N_{\phi\theta}$	Membrane forces in a shell per unit length of meridional section and a section perpendicular to the meridian, respectively
Q_θ,Q_ϕ	Shearing forces in a shell per unit length of a meridional section and a section perpendicular, respectively
r_1,r_2	Radii of curvature of an axis symmetric shell of revolution in the meridional plane and in a normal plane perpendicular to the meridian, respectively

ϕ_1, ϕ_2	Angles in the plane of the meridian measured from the axis of revolution to the edges of the shell
t	Thickness of a shell
D	Flexural rigidity of a shell
ν	Poisson's ratio
E	Young's modulus
u,v,w	Components of displacement in the x, θ, r or r, θ, ϕ directions
p	Pressure
q	Intensity of a continuously distributed load
ε_x, ε_θ	Unit elongations of a shell in the meridional direction and circumferential directions, respectively
I	Thomson function bei
P	Thomson function ber
$E(n,\alpha)$	Elliptic integral of the second kind
A_r	Cross sectional area of a stiffening ring
ε_c	Unit elongation in the circumferential direction
I_r	Area moment of inertia for a stiffening ring
e	Eccentricity from centroid of stiffening rings
Fi(n)	Trigonometric and hyperbolic combinations for solutions to differential equations
$J_1(n), H_1^{(1)}(n)$	Bessel's functions of the first order
$J_0(n), H_0^{(1)}$	Bessel's functions of the zero order

I. INTRODUCTION

In many problems of deformation of axisymmetric shells of revolution, the bending stresses can be neglected and only stresses due to strain of the middle surface need to be considered. An example of this would be a cylindrical shell submitted to an internal pressure. Under this action, the middle surface of the shell undergoes a uniform strain and since the thickness is small, the resulting hoop stress may be assumed as uniformly distributed across the thickness. For a general case under such conditions, only three unknown in-plane stress resultants exist and the problem is statically determinate. The theory based on the omission of bending stresses is called membrane theory.

If the edges of the cylindrical pressure vessel are fixed, the shell is no longer free to expand laterally and bending will occur. In problems where the load is distributed symmetrically with respect to the axis of revolution, the three equilibrium equations which can be written will contain two in-plane normal stresses, a transverse shear stress and two bending moments. Hence, the problem has become statically indeterminate. The transverse shear and bending moment which are applied at the fixed end are called generalized discontinuity stresses or stress resultants if they are taken as stresses per unit length.

In a structure comprised of axisymmetric shells of various geometries and loading configurations or multishell structures, generalized discontinuity stress resultants exist at each junction. A more detailed investigation of the theoretical results shows that the effects are limited only to a narrow region near the junction. However, their presence will govern the design of a multishell structure.

For the indeterminate multishell structure, additional equations must be provided by considering the geometry of deformations. Continuity, or compatibility, at the junctions leads to additional equations. To relate deformations to forces requires a force-displacement law be specified. For conventional elastic structures, this is Hooke's Law. These conditions combined with equilibrium of forces form the basis to the solution of the discontinuity stress resultants (and the majority of practical engineering structural designs). These principles have been applied to develop the force or flexibility method.

The purpose of this project was to write a computer program which employs the force method to determine discontinuity stress resultants for multishell structures comprised of varied axisymmetric shells of revolution under varied loading conditions.

II. THEORETICAL BACKGROUND

Force Method--Matrix Formulation

Individual axisymmetric shells of revolution many times can be analyzed without considering bending stresses. This is true in cases where the stresses are caused by a strain in the middle surface of the shell. Take, as an example, a thin spherical container submitted to the action of a uniformly distributed internal pressure normal to the surface of the shell. Under this action, the middle surface of the shell undergoes a uniform strain, and since the thickness is small, the tensile stresses can be assumed as uniformly distributed across the thickness. If the edges of the spherical shell are built in, the shell is no longer free to expand, and bending must occur. The theory based on the omission of bending is called the membrane theory of shells. In membrane theory, only two unknown in-plane normal stresses and one in-plane shear exist (this will be addressed in detail later); thus, the problem is statically determinate. In cases where bending cannot be neglected, there will be more unknowns than available equilibrium equations; hence, it is statically indeterminate.

For structures comprised of several axisymmetric shells of different geometries, the problem of deformations of the shell components is similar to the spherical shell with built in edges. The edges of each different geometry will deflect differently when submitted to an externally applied load. At a junction, the relative deformations must be zero. Therefore, a bending stress resultant and a transverse

shear stress resultant are introduced. Since they occur at a point of discontinuity, they are called generalized discontinuity stress resultants. Discontinuity stresses are caused not only by changes in geometry, but also by rapid changes in loading, shell thickness, and material properties. The effects of the generalized discontinuity stress are usually more severe than membrane stress resultants and will govern the shell design.

Since the problem is statically indeterminate, the equilibrium equations are insufficient for calculating the discontinuity stress resultants. Additional equations must be provided, and these arise from consideration of the geometry of structural deformation. Continuity, or compatibility, at the junctions provides the additional required equations. The ability to relate deformations to stress resultants is done so by the use of Hooke's Law. All these conditions must be utilized to analyze any elastic indeterminate structure.

The force, or flexibility, method of analysis rests on the basic principles described which are (1) equilibrium of force, (2) continuity of deformations, and (3) assumption of a linear force-displacement law. To develop the force method, each principle will be addressed to show its significance. However, before proceeding, a global sign convention must be established. The convention to be used is:

(1) Deflections (Δ) are positive in the direction of the outward normal to the shell surface.

(2) Rotations of the meridian are positive if the vector produced by crossing the axis of revolution into the axis perpendicular to it and in the radial direction is out of the plane of the paper, i.e., follows the right hand rule.

(3) Moments are positive in the same direction as positive rotations.

(4) Shears are positive in the same direction as positive deflections.

Figure 2-1 shows an exploded view of a containing vessel comprised of two hemispherical shells connected to a circular cylindrical shell submitted to the action of positive generalized discontinuity stress resultants. Summing forces and moments at each junction gives

$$M_1 = -M_2 \tag{2-1}$$

$$H_1 = -H_2 \tag{2-2}$$

$$M_3 = -M_4 \tag{2-3}$$

$$H_3 = -H_4 \tag{2-4}$$

Figure 2-2 shows the deformations that can occur due to some type of external load. The external load is called the primary load and the resulting deformations of each individual shell component are called primary deflections. Note that since the shell components have been separated and their respective edges are free to deform, the primary deflection can be calculated using membrane theory.

Continuity at the junctions not only requires that the primary deflections and the deflections due to the action of the generalized discontinuity stress resultants be compatible. If the deflections of

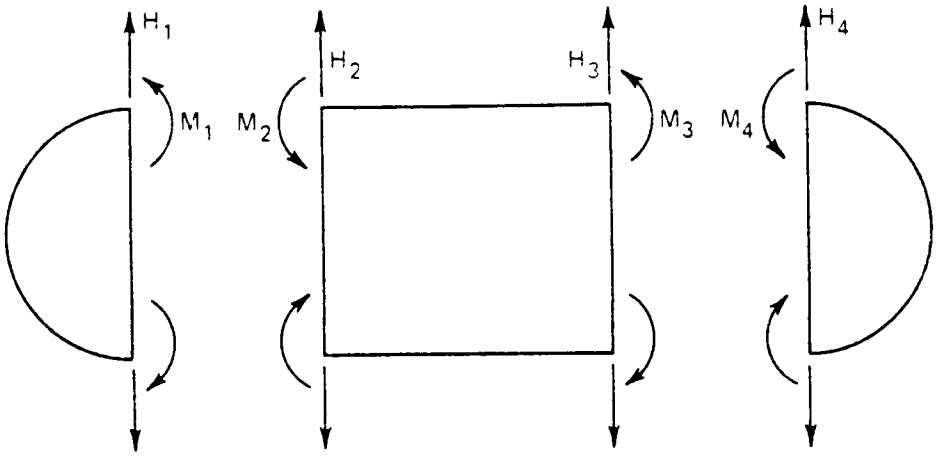


Figure 2-1. Containing vessel submitted to positive generalized discontinuity stress resultants.

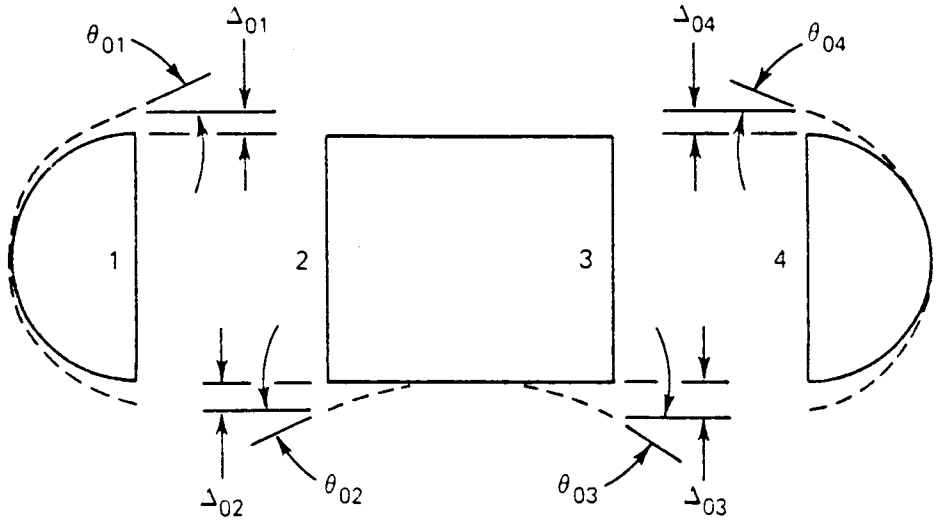


Figure 2-2. Primary edge deformations.

the shell edges due to a discontinuity stress resultants of unit magnitude are known, then the deflections of the edges could be written in terms of the unknown discontinuity stress resultants. The deflections due to unit discontinuity stress resultants are called influence coefficients. Their development incorporates the use of Hooke's Law. The notation to be used for the influence coefficients is

$\theta_i M_j$ = Rotation at edge 'i' due to unit bending stress resultant at edge 'j'.

$\theta_i H_j$ = Rotation at edge 'i' due to unit shear stress resultant at edge 'j'.

$\Delta_i M_j$ = Translation at edge 'i' due to unit bending stress resultant at edge 'j'.

$\Delta_i H_j$ = Translation at edge 'i' due to unit shear stress resultant at edge 'j'.

For the primary deformations

Δ_{oi} = translation at edge 'i',

θ_{oi} = rotation at edge 'i'.

Now the continuity of deflections at all edges can be enforced. For the vessel shown in Fig. 2-2, the compatibility equations for edge 'i' of the spherical cap and edge '2' of the cylinder are

$$\theta_{01} + (M_1)\theta_1 M_1 + (H_1)\theta_1 H_1 = \theta_{02} + (M_2)\theta_2 M_2 \quad (2\text{-}5)$$
$$- (H_2)\theta_2 H_2 + (M_3)\theta_2 M_3 + (H_3)\theta_2 H_3$$

$$\Delta_{01} + (M_1)\Delta_1 M_1 + (H_1)\Delta_1 H_1 \quad (2\text{-}6)$$
$$= \Delta_{02} - (M_2)\Delta_2 M_2 + (H_2)\Delta_2 H_2 + (M_3)\Delta_2 M_3 + (H_3)\Delta_2 H_3$$

For the junction of edge '3' of the cylinder and edge '4' of the second spherical cap

$$\theta_{03} + (M_2)\theta_3 M_2 + (H_2)\theta_3 H_2 + (M_3)\theta_3 M_3 + (H_3)\theta_3 H_3 \tag{2-7}$$
$$= \theta_{04} + (M_4)\theta_4 M_4 - (H_4)\theta_4 H_4$$

$$\Delta_{03} + (M_2)\Delta_3 M_2 + (H_2)\Delta_3 H_2 + (M_3)\Delta_3 M_3 + (H_3)\Delta_3 H_3 \tag{2-8}$$
$$= \Delta_{04} - (M_4)\Delta_4 M_4 + (H_4)\Delta_4 H_4 \cdot$$

Equations (2-5) through (2-8) contain eight unknown stress resultants. Substituting Eqs. (2-1)-(2-4) into (2-5)-(2-8) and rearranging

$$M_2(-\theta_1 M_1 - \theta_2 M_1) + H_2(-\theta_1 H_1 + \theta_2 H_2) + M_4(\theta_2 M_3) + H_4(\theta_2 H_3) \tag{2-9}$$
$$= \theta_{02} - \theta_{01}$$

$$M_2(-\Delta_1 M_1 + \Delta_2 M_2) + H_2(-\Delta_1 H_1 - \Delta_2 H_2) + M_4(\Delta_2 M_3) + H_4(\Delta_2 H_3) \tag{2-10}$$
$$= \Delta_{02} - \Delta_{01}$$

$$M_2(\theta_3 M_2) + H_2(\theta_2 H_2) + M_4(-\theta_3 M_3 - \theta_4 M_4) + H_4(-\theta_3 H_3 + \theta_4 H_4) \tag{2-11}$$
$$= \theta_{04} - \theta_{03}$$

$$M_2(\Delta_3 M_2) + H_2(\Delta_2 H_2) + M_4(-\Delta_3 M_3 + \Delta_4 M_4) + H_4(-\Delta_3 H_3 - \Delta_4 H_4) \tag{2-12}$$
$$= \Delta_{04} - \Delta_{03} \cdot$$

Using compact matrix notation

$$[f]\{F\} = \{\Delta\} \tag{2-13}$$

where

$[f]$ = the global flexibility matrix

$\{F\}$ = the generalized force vector

$\{\Delta\}$ = the generalized deformation vector.

The solution is given by

$$\{F\} = [f]^{-1}\{\Delta\} \quad . \tag{2-14}$$

A close look at the system of equations in (2-9) thru (2-12) shows that the flexibility matrix for the first spherical shell element is

$$[f]^e = \begin{bmatrix} - \theta_1 M_1 - \theta_1 H_1 \\ \Delta_1 M_1 - \Delta_1 H_1 \end{bmatrix} \quad . \tag{2-15}$$

The superscript 'e' denotes that it applies to an individual shell element.

The element deformation vector is

$$\{\Delta\}^e = \begin{Bmatrix} - \theta_{01} \\ - \Delta_{01} \end{Bmatrix} \quad . \tag{2-16}$$

For the second spherical cap element, the flexibility matrix is

$$[f]^e = \begin{bmatrix} - \theta_4 M_4 \theta_4 H_4 \\ \Delta_4 M_4 - \theta_4 H_4 \end{bmatrix} \tag{2-17}$$

and the deformation vector is

$$\{\Delta\} = \begin{Bmatrix} \theta_{04} \\ \Delta_{04} \end{Bmatrix} \quad . \tag{2-18}$$

Equations (2-15) through (2-18) hold true for any end closures.

The flexibility matrix for the cylinder is

$$\begin{bmatrix} -\theta_2 M_2 & \theta_2 H_2 & \theta_2 M_3 & \theta_2 H_3 \\ \Delta_2 M_2 & -\Delta_2 H_2 & \Delta_2 M_3 & \Delta_2 M_2 \\ \theta_3 M_2 & \theta_3 H_2 & -\theta_3 M_3 & -\theta_3 H_3 \\ \Delta_3 M_2 & \Delta_3 H_2 & -\Delta_3 H_3 & -\Delta_3 H_3 \end{bmatrix} \tag{2-19}$$

and the deformation vector

$$\{\Delta\}e \begin{Bmatrix} \theta_{02} \\ \Delta_{02} \\ -\theta_{03} \\ -\Delta_{03} \end{Bmatrix} \tag{2-20}$$

Equations (2-19) and (2-20) hold for any shaped shell element which is open at both ends.

Equations of Equilibrium

A surface of revolution is obtained by rotation of a plane curve, called the meridian, about an axis lying in the plane of the curve. The plane containing the meridian and the axis of revolution is called the meridian plane. An element of a shell is cut out by two adjacent meridians and two parallel circles (see Fig. 2-3). The position of the meridian is defined by the angle θ, measured from some datum meridian plane. The position of the parallel circle is defined by the angle made by the normal to the surface and the axis of revolution. The meridian plane and the plane perpendicular to the meridian are the planes of principal curvature and r_1 and r_2 are the corresponding principal radii of curvature. The radius of the parallel circle is denoted r_o so the

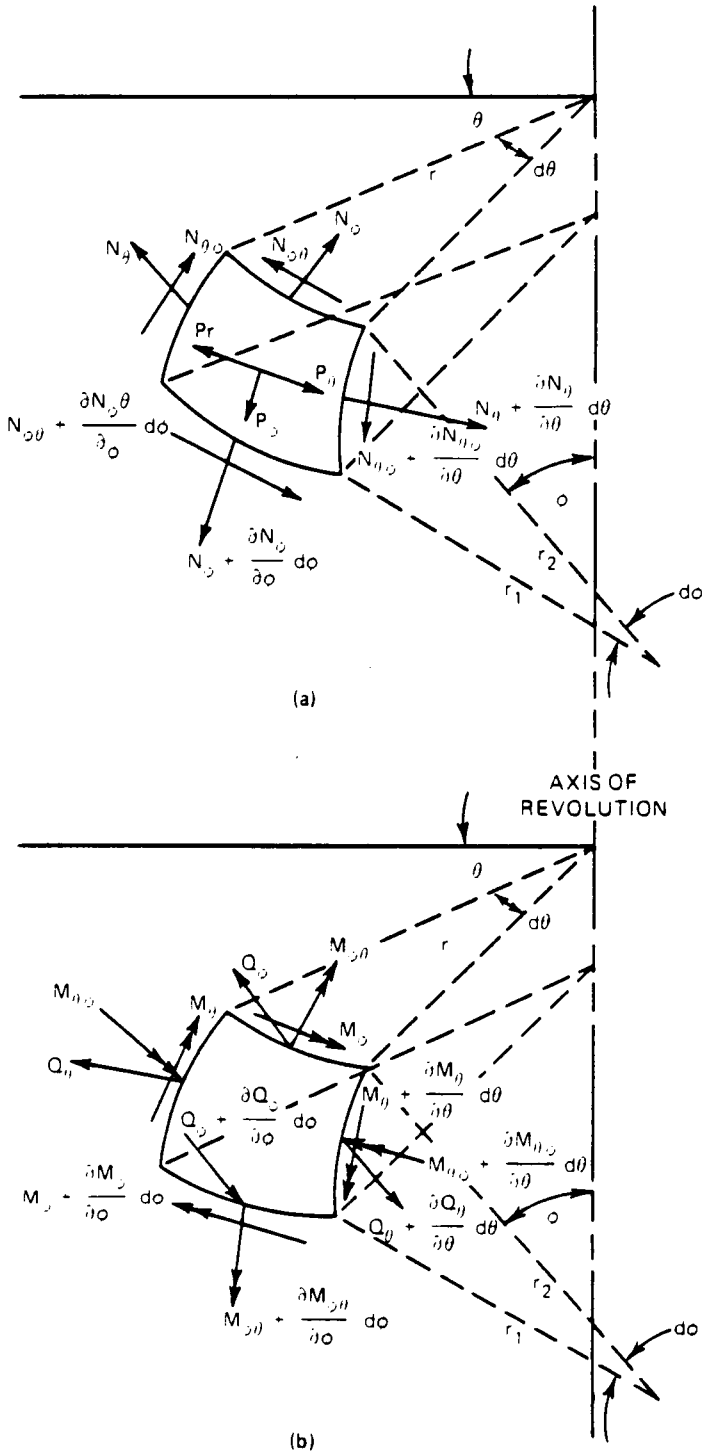


Figure 2-3. Free body diagram of a differential shell element with curvature in two principal planes.

length of the meridian is $r_1 d_\phi$ and the parallel circle is $r_o d\theta = r_2 \sin\phi \, d\theta$. The area of the element is then $r_1 r_2 \sin\phi \, d\phi d\theta$.

Now consider a free body diagram of an arbitrarily shaped shell element subjected to a load symmetrically distributed with respect to the axis of revolution (see Fig. 2-3). From the condition of symmetry, it can be concluded that only normal stresses act on the sides of the element lying in the meridian planes. Thus $N_{\theta\phi}$, $N_{\phi\theta}$, $M_{\phi\theta}$, $M_{\theta\phi}$, $\frac{dQ_\theta}{d_\theta}$ and $\frac{dN_\theta}{d\theta}$ equal zero. After these simplications have been made, the equations of equilibrium can be written by first considering the forces in the direction of the tangent to the meridian. On the upper side of the element, the force

$$N_\phi r_o d\theta = N_\theta r_2 \sin\phi \, d\theta \ . \tag{2-21}$$

is acting. The corresponding force on the lower side of the element is

$$(N_\phi + \frac{dN_\phi}{d\phi} d\phi)(r_o + \frac{dr_o}{d\phi} d\phi)d\theta \ . \tag{2-22}$$

Expanding Eq. (2-22) neglecting second order terms and combining with Eq. (2-21), the resultant is

$$N_\phi \frac{dr_o}{d\phi} d\phi d\theta + \frac{dN_\phi}{d\phi} r_o d\phi d\theta = \frac{d}{d\phi} (N_\phi r_o) d\phi d\theta \ . \tag{2-23}$$

The component of the external force in the same direction is

$$P_\phi r_1 r_o d\phi d\theta \ . \tag{2-24}$$

The forces on the lateral sides have a resultant in the direction of the radius of the parallel circle equal to $N_\theta r_1 d\phi d\theta$. This has a component of force in the direction of the tangent to the meridian of

$$-N_\theta r_1 \cos\phi \, d\phi d\theta \ . \tag{2-25}$$

Summing up Eqs. (2-23), (2-24), and (2-25), the corresponding equation of equilibrium is

$$\frac{d}{d\phi}(N_\phi r_o) - N_\theta r_1 \cos\phi + P_\phi r_1 r_o = 0 . \qquad (2\text{-}26)$$

A second equilibrium equation is obtained by summing the projections of the forces in the direction normal to the differential element. The forces acting on the upper and lower sides of the element have a resultant in this direction equal to

$$N_\phi r_o d\theta d\phi . \qquad (2\text{-}27)$$

The forces acting on the lateral sides of the element have a resultant of $N_\theta r_1 d\theta d\phi$, which has a component in the direction of the normal to the element of

$$N_\theta r_1 \sin\phi \, d\phi d\theta . \qquad (2\text{-}28)$$

The external load on the element has a component in the same direction of

$$P_r r_1 r_o d\theta d\phi . \qquad (2\text{-}29)$$

Adding Eqs. (2-27), (2-28), and (2-29) gives the second equation of equilibrium

$$N_\phi r_o + N_\theta r_1 \sin\phi + P_r r_1 r_o = 0 . \qquad (2\text{-}30)$$

The third and last equilibrium equation is obtained by considering moments with respect to the tangent to the parallel circles:

$$(M_\phi + \frac{dM_\phi}{d\phi} d\phi)(r_o + \frac{dr_o}{d\phi} d\phi) d\theta - M_\phi r_o d_\theta \qquad (2\text{-}31)$$

$$- M_\theta r_1 \cos\phi \, d\phi d\theta - Q_\phi r_2 \sin\phi \, r_1 d\phi d\theta = 0 .$$

Simplifying, this equation together with Eqs. (2-26) and (2-30) gives the system of three equilibrium equations for any shell with two distinct principal radii of curvatures.

$$\frac{d}{d\phi}(N_\phi r_o) - N_\theta r_1 \cos\phi - r_o Q_\phi + r_o r_1 P_\phi = 0 \tag{2-32}$$

$$N_\phi r_o + N_\theta r_1 \sin\phi + \frac{d(Q_\phi r_o)}{d\phi} + P_r r_1 r_o = 0 \tag{2-33}$$

$$\frac{d}{d\phi}(M_\phi r_o) - M_\theta r_1 \cos\phi - Q_\phi r_1 r_o = 0 \ . \tag{2-34}$$

Considerable simplification of Eqs. (2-32), (2-33), and (2-34) can be obtained by introducing two new variables. Take as the first variable the angle of rotation of a tangent to a meridian. Denoting this angle V, the quantity is

$$V = \frac{1}{r_1}\left(v + \frac{dw}{d\phi}\right) \ . \tag{2-35}$$

The second variable is

$$U = r_2 Q_\phi \ . \tag{2-36}$$

Replacing Eq. (2-32)

$$2\pi r_o N_\phi \sin\phi + 2\pi r_o Q_\phi \cos\phi = 0 \tag{2-37}$$

which is the equilibrium equation for a portion of the shell where $0 \leq \phi \leq \phi_2$ with no external load applied. Rearranging, one obtains

$$N_\phi = -Q_\phi \cot\phi = -\frac{1}{r_2} U \cot\phi \ . \tag{2-38}$$

Substituting this into Eq. (2-33) and taking $P_r = 0$,

$$r_1 N_\theta \sin\phi = -N_\phi r_o - \frac{d(Q_\phi r_o)}{d\phi} \quad , \tag{2-39}$$

and noting $r_o = r_2 \sin\phi$,

$$N_\theta = \frac{1}{r_1}\frac{d}{d\phi}(Q_\phi r_2) = -\frac{1}{r_1}\frac{dU}{d\phi} \, . \tag{2-40}$$

The membrane forces have been successfully represented in terms of U which is dependent upon the transverse shear stress resultant Q_ϕ. Now V and U may be connected by relating the membrane stress resultants to the displacements of the element by Hooke's Law. The resulting equations are

$$\frac{r_2}{r_1}\frac{d^2U}{d\phi^2} + \frac{1}{r_1}\left[\frac{d}{d\phi}\left(\frac{r_2}{r_1}\right)\cot\phi - \frac{r_2}{r_1 t}\frac{dt}{d\phi}\right]\frac{dU}{d\phi} \tag{2-41}$$

$$-\frac{1}{r_1}\left[\frac{r_1}{r_2}\cot^2\phi - \nu - \frac{\nu}{t}\frac{dt}{d\phi}\cot\phi\right] U = EtV,$$

and

$$\frac{r_2}{r_1}\frac{d^2V}{d\phi^2} + \frac{1}{r_1}\left[\frac{d}{d\phi}\left(\frac{r_2}{r_1}\right) + \frac{r_2}{r_1}\cot\phi + 3\frac{r_2}{r_1}\frac{dt}{d\phi}\right]\frac{dV}{d\phi}$$

$$-\frac{1}{r_1}\left[\nu - \frac{3\nu\cot\phi}{t}\frac{dt}{d\phi} + \frac{r_1}{r_2}\cot^2\phi\right] V = -\frac{U}{D} \, . \tag{2-42}$$

where

$$D = \frac{Et^3}{12(1-\nu^2)} \, . \tag{2-43}$$

E = Young's Modulus

ν = Poisson's Ratio

t = shell thickness

Introducing the notation

$$L(\ldots) = \frac{r_2}{r_1}\frac{d^2(\ldots)}{d\phi^2} + \frac{1}{r_1}\left[\frac{d}{d\phi}\left(\frac{r_2}{r_1}\right) + \frac{r_2}{r_1}\cot\phi\right]\frac{d(\ldots)}{d\phi} \qquad (2\text{-}44)$$

$$- \frac{r_1\cot^2\phi}{r_2 r_1}(\ldots) .$$

Eqs. (2-41) and (2-42) may be represented in the following simplified fashion.

$$L(U) + \frac{\nu}{r_1}U = EtV \qquad (2\text{-}45)$$

$$L(V) - \frac{\nu}{r_1}V = -\frac{U}{D} . \qquad (2\text{-}46)$$

In the previous discussion, the equations were developed for a shell with curvature in two principal planes. For cylindrical and conical shells, curvature is only in one principal plane and r_1 goes to infinity. Introducing the change of variable

$$d_x = r_1 d\phi , \qquad (2\text{-}47)$$

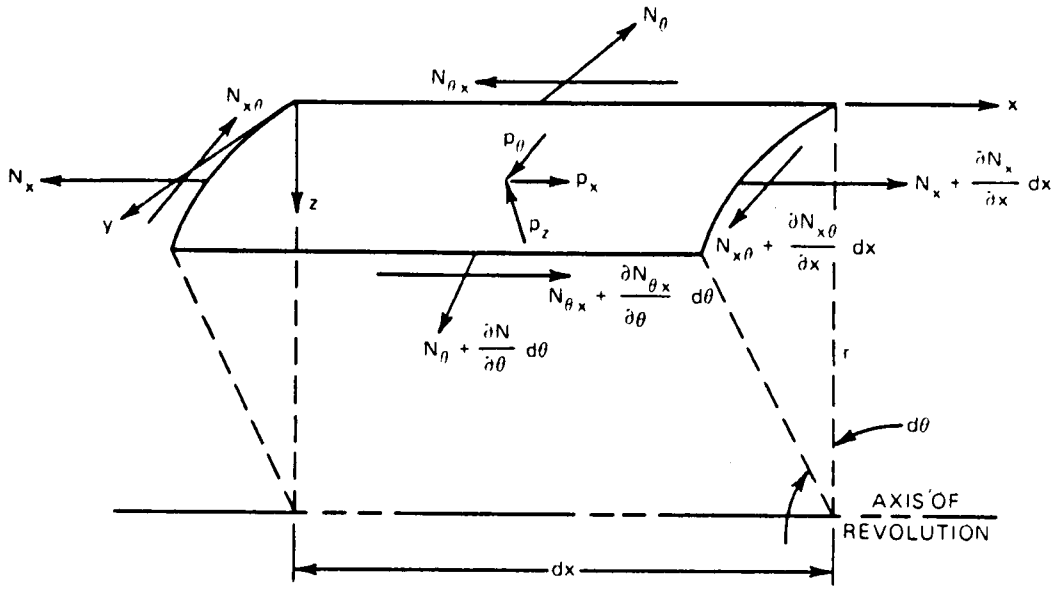
the equilibrium Eqs. (2-32), (2-33), and (2-34) become

$$\frac{r_1 d}{dx}(N_x r_o) - N_\theta r_1 \cos\phi - r_o Q_x + r_o r_1 P_\theta = 0 \qquad (2\text{-}48)$$

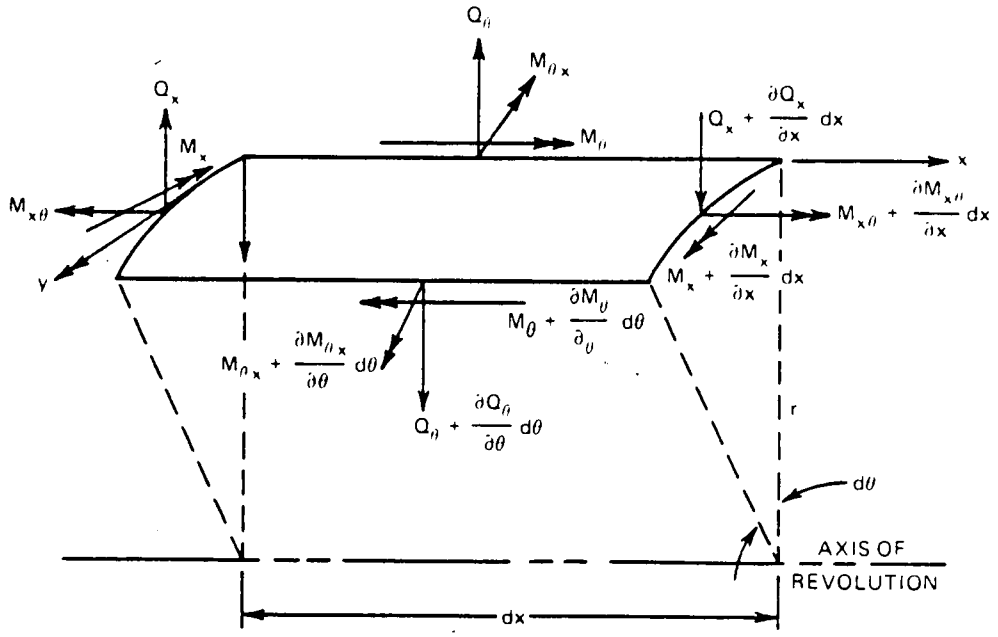
$$N_x r_o + N_\theta r_1 \sin\phi + \frac{r_1 d(Q_x r_o)}{dx} + P_z r_1 r_o = 0 \qquad (2\text{-}49)$$

$$\frac{r_1 d(M_x r_o)}{dx} - N_\theta r_1 \cos\phi - Q_x r_1 r_o = 0 . \qquad (2\text{-}50)$$

(See Fig. 2-4.)



(a)



(b)

Figure 2-4. Free body diagram of a differential cylindrical shell element.

Multiplying Eqs. (2-48), (2-49) and (2-50) by $\frac{1}{r_1}$, letting r_1 approach infinity and noting ϕ equals $\pi/2$, they become

$$\frac{d(N_x)}{dx} + P_\theta = 0 \tag{2-51}$$

$$N_\theta + \frac{d(Q_x)r_o}{dx} + P_z r_o = 0 \tag{2-52}$$

$$\frac{d(M_x)}{dx} - Q_x = 0 \ . \tag{2-53}$$

Influence Coefficients

The equations of equilibrium can be solved to determine the necessary elements of the flexibility matrix. The elements are commonly called influence coefficients. They will be developed for the same two general cases for which the equations of equilibrium were presented. The first case will be for shells which have only one principal plane of curvature, i.e., cylindrical and conical. The second case will be for shells with two principal planes of curvature, i.e., spherical and ellipsoidal.

Cylindrical and Conical Shells. The equations of equilibrium for cylindrical shells are

$$\frac{d(N_x)}{dx} + P_\theta = 0 \tag{2-54}$$

$$N_\theta + \frac{d(Q_x)r_o}{dx} + P_z r_o = 0 \tag{2-55}$$

$$\frac{d(M_x)}{dx} - Q_x = 0 \ . \tag{2-56}$$

Assuming that the external forces consist of only a pressure normal to the shell (Eqs. 2-54, 2-55, and 2-56) reduce to

$$\frac{d(N_x)}{dx} = 0 \tag{2-57}$$

$$N_\theta + \frac{d(Q_x)}{dx} + P_z r_o = 0 \tag{2-58}$$

$$\frac{d(M_x)}{dx} - Q_x = 0 \, . \tag{2-59}$$

Equation (2-57) indicates that N_x is constant and assumed to be zero. If N_x is other than zero, the deflections and stresses can be easily calculated. The remaining two Eqs. (2-58) and (2-59) contain three unknowns: N_θ, Q_x, and M_x. Since the problem is indeterminate, the deflections of the middle surface must be investigated. This problem can be simplified since the displacement in the circumferential direction θ (Fig. 2-4) vanishes because of symmetry. Thus only the components in the axial and radial directions, u and w respectively, must be considered. The expressions for the strain components are:[1]

$$\varepsilon_x = \frac{du}{dx} \tag{2-60}$$

$$\varepsilon_\phi = -\frac{w}{r} \tag{2-61}$$

where $r = r_2$ for cylindrical and conical shells. Applying Hooke's Law,

$$N_x = \frac{Et}{1-\nu^2}(\varepsilon_x + \nu\varepsilon_\phi) = \frac{Et}{1-\nu^2}\left(\frac{du}{dx} - \frac{w}{r}\nu\right) = 0 \tag{2-62}$$

$$N_\theta = \frac{Et}{1-\nu^2}(\varepsilon_\phi + \nu\varepsilon_x) = \frac{Et}{1-\nu^2}\left(-\frac{w}{r} + \nu\frac{du}{dx}\right) \, . \tag{2-63}$$

From Eq. (2-62)

$$\frac{du}{dx} = \frac{\nu w}{r} \tag{2-64}$$

substituting Eq. (2-64) into Eq. (2-63) gives

$$N_\phi = - \frac{Etw}{r} \ . \tag{2-65}$$

Again from symmetry, it can be concluded that there is no change in the curvature in the circumferential direction (θ). The curvature in the axial direction is $- \frac{d^2w}{dx^2}$.[1] The expressions for the bending moments become

$$M_\theta = \nu M_x \tag{2-66}$$

$$M_x = - D \frac{d^2w}{dx^2} \tag{2-67}$$

where D is given by Eq. (2-43). Solving Eq. (2-59) for Q_x and substituting this quantity along with Eqs. (2-67) and (2-65) into Eq. (2-58) yields

$$\frac{d^2}{dx^2} \left(D \frac{d^2w}{dx^2}\right) + \frac{Et}{r^2} w = P_z \ . \tag{2-68}$$

If the thickness is constant, the simplified notation can be introduced

$$\beta^4 = \frac{Et}{4rD} = \frac{3(1-\nu^2)}{r^2t^2} \tag{2-69}$$

and Eq. (2-68) becomes

$$\frac{d^4w}{dx^4} + 4\beta^4 w = \frac{P_z}{D} \ . \tag{2-70}$$

Hence, all problems of symmetrical deformation of a circular cylindrical shell are reduced to the integration of one fourth order nonhomogeneous differential equation.

The general solution to Eq. (2-70) is

$$w = e^{\beta x}(A_1 \cos \beta x + A_2 \sin \beta x) + e^{-\beta x}(B_1 \cos \beta x + B_2 \sin \beta x) + f(x) \tag{2-71}$$

where f (x) is a particular solution dependent on the load term P_z, and A_1, A_2, B_1, B_2 are constants of integration dependent on the boundary conditions.

As discussed earlier, the influence coefficients are the effects of a uniformly distributed unit moment and unit transverse shear stress resultants. Assuming that the shell is sufficiently long so that the end effects are independent of each other, the boundary conditions which must be satisfied are

$$M_x = - D \frac{d^2w}{dx^2} = M_o \tag{2-72}$$

$$Q_x = \frac{d(Mx)}{dx} = - D(\frac{d^3w}{dx^3}) = Q_o \tag{2-73}$$

at x = 0. Therefore, the influence coefficients are independent of any external load. Substituting Eq. (2-72) and Eq. (2-73) into Eq. (2-71), and setting $A_1 = A_2 = f(x) = 0$ gives

$$C_3 = \frac{1}{2\beta^3 D} (Q_o + \beta M_o) \tag{2-74}$$

$$C_4 = \frac{M_o}{2\beta^2 D} \cdot \tag{2-75}$$

The final expression for w becomes

$$w = \frac{e^{-\beta x}}{2\beta^3 D}\left[\beta M_o(\sin \beta x - \cos \beta x) - Q_o \cos \beta x\right] . \qquad (2\text{-}76)$$

The slope of the meridian is given by differentiating Eq. (2-76) with respect to x. Substituting x = 0 into Eq. (2-76), and into the derivative of (2-76) with respect to x along with alternating $M_o = 1$, $Q_o = 0$, and then $M_o = 0$ and $Q_o = 1$ gives

$$\theta_i M_i = \frac{1}{\beta d} \qquad (2\text{-}77)$$

$$\theta_i Q_i = \frac{1}{2\beta^2 D} \qquad (2\text{-}78)$$

$$\Delta_i M_i = \frac{1}{2\beta^2 D} \qquad (2\text{-}79)$$

$$\Delta_i Q_i = \frac{1}{2\beta^3 D} \qquad (2\text{-}80)$$

which are the respective influence coefficients for a circular cylinder.

Conical shells also have curvature in only one principal plane. Thus, the results obtained in Eq. (2-76) through Eq. (2-79) can be extended to accommodate them. Figure 2-5 shows the notation used to define open and closed conical shells. Noting that the principal radius of curvature can be expressed by $r_o = \bar{x} \cot \alpha$, then

$$\beta^4 = \frac{3(1-\nu^2)}{\bar{x}^2 \cot^2\alpha t^2} \qquad (2\text{-}81)$$

Substituting Eq. (2-81) into Eq. (2-77) through Eq. (2-80) gives the influence coefficients for a conical shell.

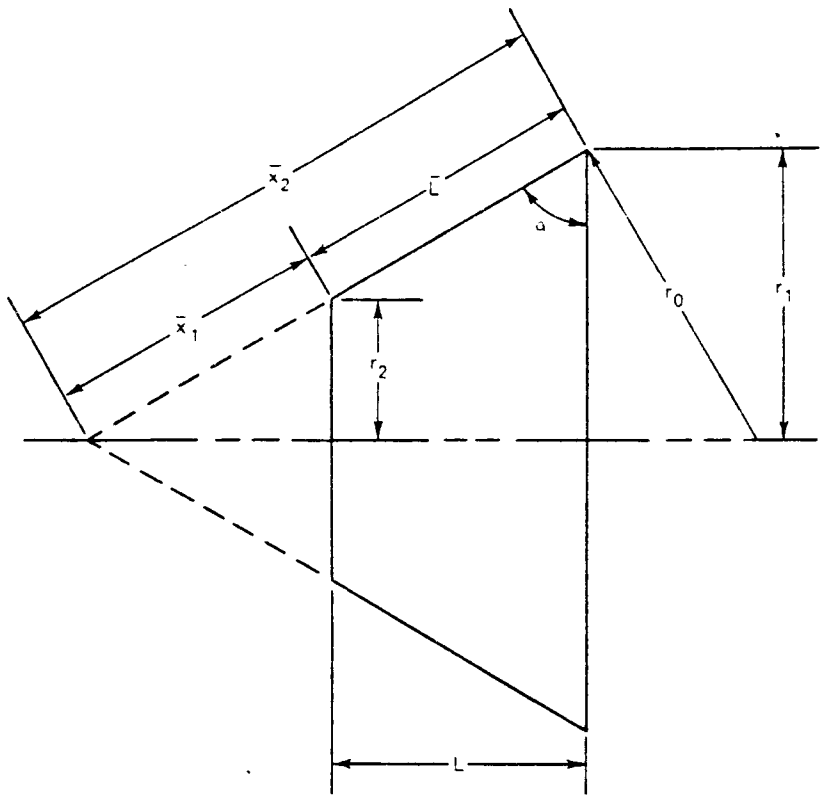


Figure 2-5. Notation for conical shells.

Spherical and Ellipsoidal Shells. The most common geometric shape of a shell with curvature in two principal planes is the hemisphere. It is used extensively in pressure vessel design and to enclose large structures. Accurate solutions to the governing equations can be very laborious; therefore, several approximate procedures have been developed. Because of simplifications introduced in these approximate methods, the results are not accurate for all types of spherical shells.[2]

One of the approximate procedures is the method of asymptotic integration.[1] This method combines the two second order differential equations (2-45) and (2-46) into one fourth order equation which has solution given by:[1]

$$Q_\phi = \frac{1}{\sin\phi} e^{\beta\phi} (C_1 \cos\beta\phi + C_2 \sin\beta\phi) + e^{-\beta\phi}(C_3 \cos\beta\phi + C_4 \sin\beta\phi) \tag{2-82}$$

where

$$4\beta^4 = (1 - \nu^2)(1 + \frac{12r^2}{t^2}) \cdot \tag{2-83}$$

Noting that the equilibrium equations (2-45 and 2-46) can be rewritten as

$$\frac{d^2Q_\phi}{d\phi^2} + \cot\phi \frac{dQ_\phi}{d\phi} - (\cot^2\phi - \nu)Q_\phi = EtV \tag{2-84}$$

$$\frac{d^2V}{d\phi^2} + \cot\phi \frac{dV}{d\phi} - (\cot^2\phi + \nu)V = \frac{r^2Q_\phi}{D} \cdot \tag{2-85}$$

For thin shells β is very large; therefore, successive derivatives of Eq. (2-82) become very large. This indicates that a satisfactory approximation can be made by neglecting the terms containing Q_ϕ and V and their first derivatives. The simplified equations can be combined eliminating V which gives

$$\frac{d^4 Q_\phi}{d\phi^4} + 4\lambda^4 Q_\phi = 0 \tag{2-86}$$

where

$$\lambda^4 = 3(1-\nu^2)\left(\frac{r}{t}\right)^2 \tag{2-87}$$

The general solution is similar to Eq. (2-71) and is

$$Q_\phi = C_1 e^{\mu} \cos \mu + C_2 e^{\mu} \sin \mu + C_3 e^{-\mu} \cos \mu + C_4 e^{-\mu} \sin \mu \tag{2-88}$$

where

$$\mu = \lambda\phi \tag{2-89}$$

and C_1, C_2, C_3, C_4 are constants of integration dependent on the boundary conditions. This method is good for large values of β and is commonly referred to as Geckeler theory.[1]

For small values of β or ϕ_2, Fig. 2-6, the accuracy of the above procedure diminishes. Shells of this nature are referred to as shallow spherical shells. Accurate solutions can be obtained to the governing equations by use of hypergeometric series.[2] However, it is cumbersome to use for this type of shells because of slow convergence. Therefore Baker and Cline[3] developed one approximate method which gives accuracy comparable to the Geckeler theory for nonshallow shells and to the methods by Flugge[2] for shallow shells. The total development of the method will not be described, but only the necessities for its application.

Introducing the variable

$$W = \Psi(\phi)\ \eta\ (\xi) \tag{2-90}$$

where

$$\xi(\phi) = \left[\frac{12(1-\nu^2)}{t^2}\right] \int_0^{\phi} \frac{r_1(\mu)}{r_2(\mu)}\, d\mu \tag{2-91}$$

$$\Psi(\phi) = \left[\frac{1}{r_2(\phi)\ \sin\phi} \int_0^{\phi} \frac{r_1(\mu)}{r_2(\mu)}\, d\mu\right] \tag{2-92}$$

allows the U and V to be written as

$$U = \Psi(\phi)\ \ [AR' + BI'] \tag{2-93}$$

$$V = k^{-1}\Psi(\phi)\ [AI' - BR'] \tag{2-94}$$

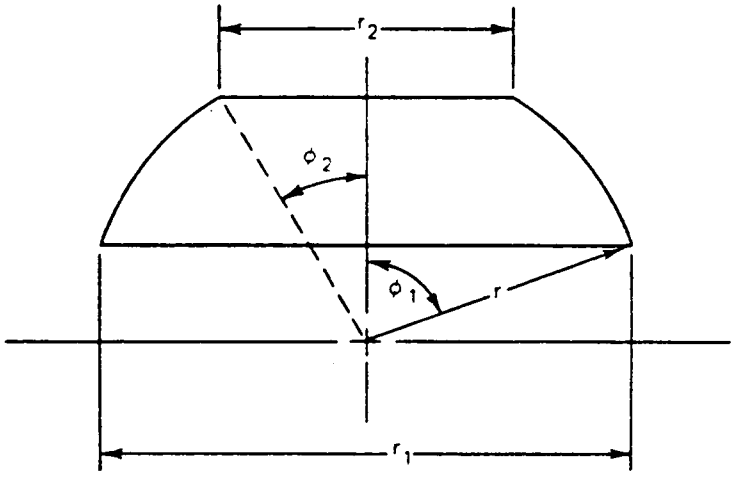


Figure 2.6. Notation for Hemispherical Shells.

where

$$k = \frac{Et^2}{12(1-\nu^2)^{1/2}} \tag{2-95}$$

and R = berξ, I = beiξ for a "closed" shell and R = kerξ, I = keiξ for an "open" shell. R and I are called Thomson functions and are discussed in Appendix B. Substituting Eqs. (2-91) and (2-92) into expressions of the transverse shear stress resultant Q_ϕ, the radial displacement Δ and the meridional bending moment resultant M_ϕ, which are in terms of U and V,[2] one can obtain the following quantities for the influence coefficients

$$\Delta_i H_i = \frac{r_2^{3/2} \sin^2\phi}{(Etk)^{1/2}} K_{11} \tag{2-96}$$

$$\theta_i H_i = \Delta_i M_i = \frac{r_2 \sin\phi}{k} K_{12} \tag{2-97}$$

$$\theta_i M_i = \frac{Etr_2}{k^{3/2}} K_{22} \tag{2-98}$$

(Q_x = H to be consistent with notation given in the previous section.)

where

$$K_{11} = \frac{1}{C} I^2 + R^2 + \frac{2}{\xi}(IR' - RI') + \frac{1-\nu^2}{\xi}(I'^2 + R'^2) \tag{2-99}$$

$$K_{12} = \frac{II' + RR'}{C} \tag{2-100}$$

$$K_{22} = \frac{I'^2 + R'^2}{C} \tag{2-101}$$

$$C = RI' - IR' - \frac{1-\nu}{\xi}(I'^2 + R'^2) \; . \tag{2-102}$$

K's are only dependent on and which are the only parameters which must be specified for different geometries.

For spherical shells $r_1 = r_2 = r$; $\Psi(\phi) \cong 1$ and

$$\xi \cong \frac{Etr}{k}^{1/2} \tag{2-103}$$

Figure 2-7 shows the notation used for ellipsoidal shells. The required parameter ξ, is

$$\xi = \frac{Etr_o b}{k^2 a} \alpha + E(n,\alpha) \tag{2-104}$$

where

$$r_o = \frac{a^2}{b} \tag{2-105}$$

$$n^2 = 1 - \frac{b^2}{a^2} \tag{2-106}$$

$$\tan\alpha = \frac{a}{b} \tan\phi \tag{2-107}$$

and $E(n,\alpha)$ is the elliptic integral of the second kind (see Appendix C).

Stiffening Rings. Stiffening rings are widely used as an integral part in multishell structures. They reduce stresses in domes when support tangent to the meridian is impossible and prevent buckling in vacuum or externally loaded vessels.

Figure 2-8 shows a portion of a stiffening ring subjected to a uniformly distributed radial force and moment. The radial force H_o, acting through the centroid, causes a constant circumferential force, T, in the ring of

$$T = H_o \int_o^{\pi/2} r \cos d\theta = H_o r \tag{2-108}$$

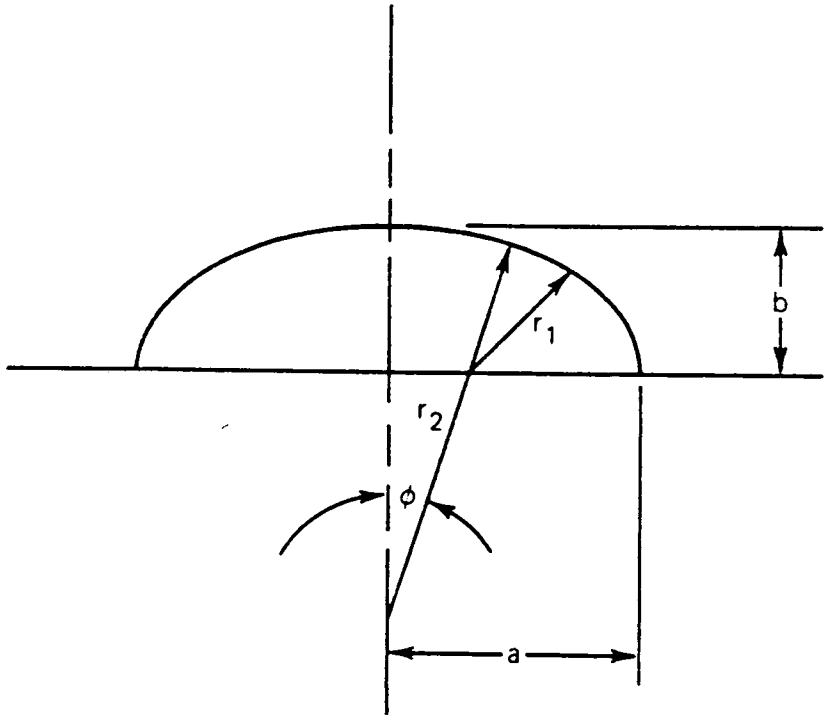


Figure 2-7. Notation for Ellipsoidal Shells.

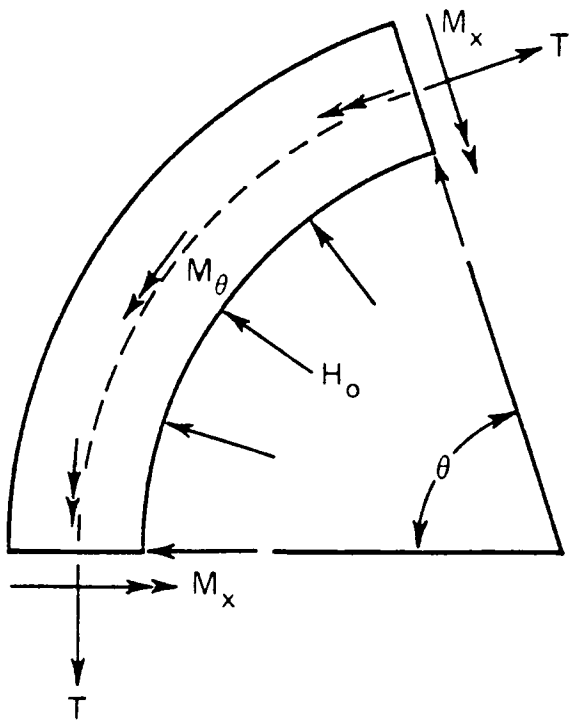


Figure 2-8. Free body diagram of a stiffening ring.

and a circumferential strain of

$$\varepsilon_c = \frac{T}{E\ A_r} \tag{2-109}$$

which gives a radial deflection of

$$\Delta = \frac{r\ T}{E\ A_r} = \frac{r^2\ H_o}{E\ A_r} \quad . \tag{2-110}$$

For a uniformly distributed radial moment M_θ, acting about the centroid, the reaction is

$$M_x = 2 \int_0^{\pi/2} M_\theta r \cos\theta\ d\theta = 2\ M_\theta r \quad . \tag{2-111}$$

The circumferential strain is

$$\varepsilon_c = \frac{M_\theta r\ y}{E\ I_r} \tag{2-112}$$

where $y = d/2$

which gives a radial deflection of

$$\Delta = \frac{r^2 y\ M_\theta}{E\ I_r} \quad . \tag{2-113}$$

and a rotation of

$$\theta = \frac{\delta r}{y} = \frac{r^2 M_\theta}{E\ I_r} \quad . \tag{2-114}$$

Figure 2-9 shows a ring cross-section subjected to positive generalized discontinuity forces. By observation, one may conclude that the rotations at edges 1 and 2 caused by M_1 and M_2 are

$$\theta_{ij} = \frac{r^2\ M_j}{E\ I_r} \quad . \qquad i,\ j = 1,2 \tag{2-115}$$

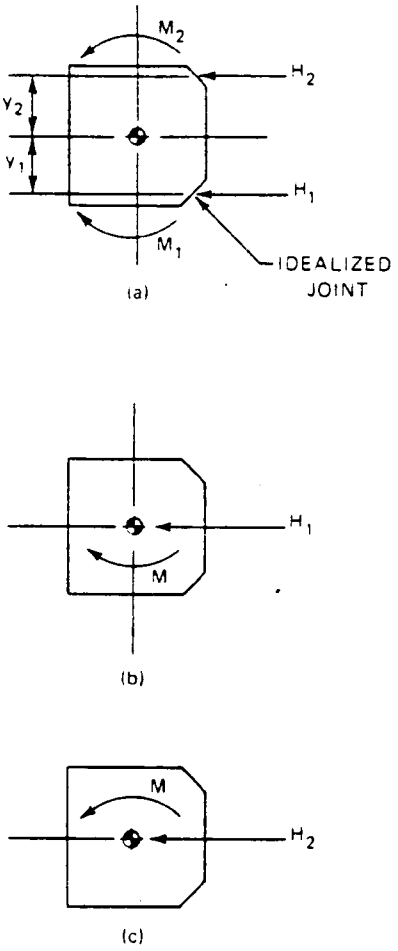


Figure 2-9. Cross section of a stiffening ring subjected to edge discontinuity forces.

If the shear force, H_1, is replaced by a force and couple at the centroid as shown in Fig. 2-9(b), the resulting rotations and deflections are

$$\theta_{11} = \frac{H_1(d/2)r^2}{E\ I_r} \tag{2-116}$$

$$\Delta_{11} = \frac{r\ H_1}{E\ A_r} + \frac{H_1(d/2)(d/2)r^2}{E\ I_r} \tag{2-117}$$

$$\theta_{21} = -\frac{H_1(d/2)r^2}{E\ I_r} \tag{2-118}$$

$$\Delta_{21} = \frac{r\ H_1}{E\ A_r} - \frac{H_1(d/2)(d/2)r^2}{E\ I_r} \quad . \tag{2-119}$$

If the shear force at edge 2 is replaced by a force and couple acting at the centroid, the corresponding rotations and deflections are

$$\theta_{12} = \frac{H_2 y_2 r^2}{E\ I_r} \tag{2-120}$$

$$\Delta_{12} = \frac{H_2 r}{E\ A_r} - \frac{H_2 y_2 y_1 r^2}{E\ I_r} \tag{2-121}$$

$$\theta_{22} = \frac{H_2 y_2 r^2}{E\ I_r} \tag{2-122}$$

$$\Delta_{22} = \frac{r\ H_2}{E\ A_r} + \frac{H_2 y_2 y_1 r^2}{E\ I_r} \quad . \tag{2-123}$$

If $H_1 = H_2 = M_1 = M_2 = 1$, then Eqs. (2-115) through (2-123) yields the influence coefficients.

Closed Flat Circular Plates. Since a shear perpendicular to the axis of revolution has a negligible effect on a flat plate, only the effects from the bending moment need to be calculated, Fig. 2-10. The rotation of the edge of a simply supported circular plate is given as

$$\theta = \frac{M_r}{D(1 + \mu)} \quad . \tag{2-124}$$

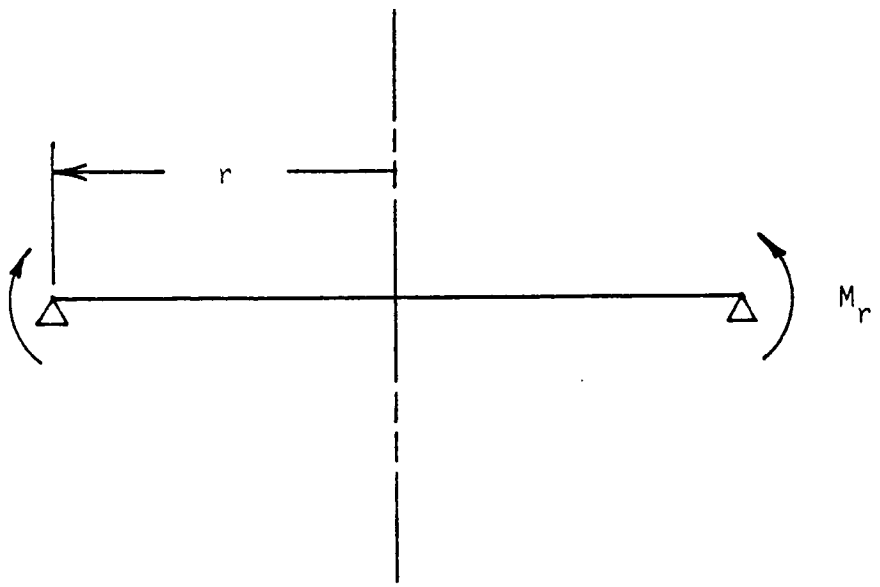


Figure 2-10. Flat circular plate subjected to edge moments.

III. PROGRAM ORGANIZATION

Subroutine Functions

ALSHEL is a series of subroutines which will calculate primary displacements and influence coefficients, assemble them into a global system of equations and solve for discontinuity forces. The geometries included are circular cylindrical shells, open and closed hemispherical shells, closed ellipsoidal shells, open and closed conical shells, stiffening rings, and flat circular plates. GRAPH takes the output from ALSHEL and plots the meridional deflection and rotation, and the applicable force resultants as a function of the respective meridional length.

All subroutines in ALSHEL and GRAPH are named for the function that they perform. For example, subroutine CYLIND calculates primary displacements and influence coefficients for a circular cylindrical shell and subroutine ASSEMB assembles the individual displacement vectors and influence coefficients into a global system of equations. The following is a complete list of the subroutines in ALSHEL and their function:

(1) MAIN — Reads the necessary data to describe a structure and determines the order of operations.

(2) CYLIND — Calculates influence coefficients and primary displacements caused by hydrostatic pressure and uniform pressure loads for circular cylindrical shells.

(3) SPHERE Calculates influence coefficients and primary displacements due to uniform pressure for open crown spherical shells.

(4) HEMI Calculates influence coefficients and primary displacements due to uniform pressure, horizontal projected and weight loadings.

(5) CLSCON Calculates influence coefficients and primary displacements due to uniform pressure for closed conical shells.

(6) OPNCON Calculates influence coefficients and primary displacements due to uniform pressure for open conical shells.

(7) PLATE Calculates influence coefficients and primary displacements with uniform pressure loading for flat circular plates.

(8) ELLIPS Calculates influence coefficients and primary displacements with uniform pressure loading for flat circular plates.

(9) ASSEMB Assembles the element influence coefficients and primary displacements into a global system of equations. ASSEMB is called by subroutines (2.)- (8.).

(10) INVERT Inverts the global flexibility matrix and multiplies it by the global displacement vector.

(11) RITEIT Writes the results into RESULT.DAT and GRAPH.DAT.

(12) GAUSS Evaluates the elliptic integral of the second kind by the six point Gaussian Quadrature method.

(13) KELVIN Evaluates the Thomson functions. KELVIN is called by subroutines SPHERE, HEMI, and ELLIPS.

All subroutines which calculate influence coefficients and primary displacements write the element results to files with the same name as the subroutine with a .DAT extension.

Program Usage

A structure's geometrical and loading configuration is described by four letter words in a file called ALSHEL.DAT. This file is used by the main parts of ALSHEL and GRAPH to determine the order of necessary operations.

A minimum of nine (maximum of ten is available) four letter words is necessary to describe a structure. They are followed by a line of data which describes that particular feature of a shell. The following list gives the four letter words and their duty (they also appear in the recommended order in which they should be in ALSHEL.DAT):

(1) TITL - job or structure title

(2) NUMB - total number of individual elements

(3) TYPE - the type of elements that comprise the structure designated as

1 - open spherical shells

2 - circular cylindrical shells

3 - closed hemispherical heads

4 - open conical sections

5 - flat circular plates

6 - closed conical heads

7 - closed elliptical heads

8 - stiffening rings

(4) RAD1 - larger radii measured perpendicular to the axis of revolution

(5) RAD2 - smaller radii measured perpendicular to the axis of revolution (note that this module is necessary only for open spherical and conical shells)

(6) LENT - axial length of a shell element

(7) THIC - shell thickness

(8) LOAD - type of loading designated as

1 - uniform pressure

2 - hydrostatic pressure

3 - weight

4 - horizontal uniform load

(Note 1 is applicable to all shell elements while 2 is only applicable to cylindrical shells; and 3 and 4 are only good for closed hemispherical heads)

(9) PRES - maximum pressure

(10) MATR - is followed by two lines of data, the first of which is Young's modulus and the second is Poisson's ratio.

Each of the data is stored in an array of the same name and the results are contained in RESULT.DAT. Several examples have been provided for clarification.

Example Problem

Example 1. Determine the generalized discontinuity forces shown in Fig. 3-1. The vessel is all steel with a Young's Modulus of 30 X 10^6 psi, Poisson's ratio of .3 and a uniform internal pressure of 500 psi.

Solution: The vessel is comprised of seven different types of shells. If the axes of revolution were horizontal, they would be input into ALSHEL from left to right, but since they are vertical, they must be input from bottom to top. The following is the required data file:

```
TY ALSHEL.DAT
TITL
**********EXAMPLE NO. 1*********
NUMB
7
TYPE
7 2 1 2 4 2 6
RAD1
24 24 24 12 24 24 24
RAD2
12
12
LENT
12 50 20.8 30 30 50 15
THIC
.75 1.5 .75 1 1.25 1.5 1.875
LOAD
1 1 1 1 1 1 1
PRES
500 500 500 500 500 500 500
MATR
30E06,30E06,30E06,30E06,30E06,30E06,30E06
.3 .3 .3 .3 .3 .3 .3
```

Note that only two values are needed for the RAD2 modulus for the open conical and hemispherical shells and are input in a column instead of a row.

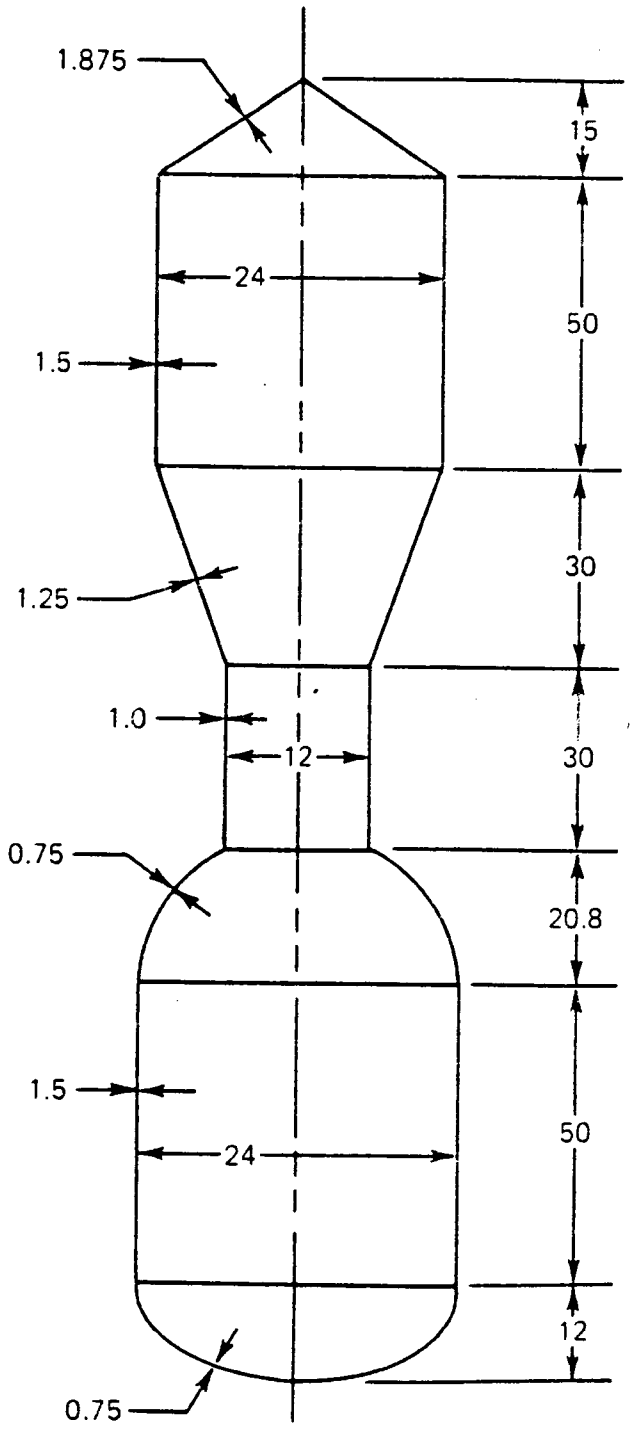


Figure 3-1. Pressure vessel comprised of various shell geometries.

The following are the results calculated by ALSHEL:

```
TY RESULT.DAT
*******EXAMPLE NO. 1**********
ALL MOMENTS AND SHEARS ARE IN GLOBAL X-Y COORDINATES
MOMENTS ARE IN (IN-LB/IN) AND SHEARS ARE IN (LB/IN).
                ELLIPTICAL END
          MOMENT              SHEAR
          940.7               -434.6
               CYLINDRICAL SHELL
          END 1               END 2
     MOMENT   SHEAR       MOMENT  SHEAR
     -940.7   434.6         73.7   68.0
          OPEN CROWN SPHERICAL SECTION
          END 1               END 2
      (LARGE END)          (SMALL END)
     MOMENT   SHEAR       MOMENT  SHEAR
     -73.7    -68.0       -4042.3 -6412-4
               CYLINDRICAL SHELL
          END 1               END 2
     MOMENT   SHEAR       MOMENT  SHEAR
     4042.3   6412.4        89.2   -2.2
```

OPEN CONICAL SECTION

END 1 (SMALL END)		END 2 (LARGE END)	
MOMENT	SHEAR	MOMENT	SHEAR
-89.2	2.2	-127.0	-107.4

CYLINDRICAL SHELL

END 1		END 2	
MOMENT	SHEAR	MOMENT	SHEAR
127.0	107.4	1090.1	99.5

CLOSED CONICAL HEAD

MOMENT	SHEAR
-1090.1	-99.5

Figure 3-2 shows an exploded view of the structure with the moments and forces acting in the proper directions. The maximum shear and moment is at the junction of the small end of the open crown spherical shell and the cylindrical shell. Figures 3-3 through 3-15 show the effects of the discontinuity forces for the two respective shells. These plots are available for all geometries except ellipsoidal heads.

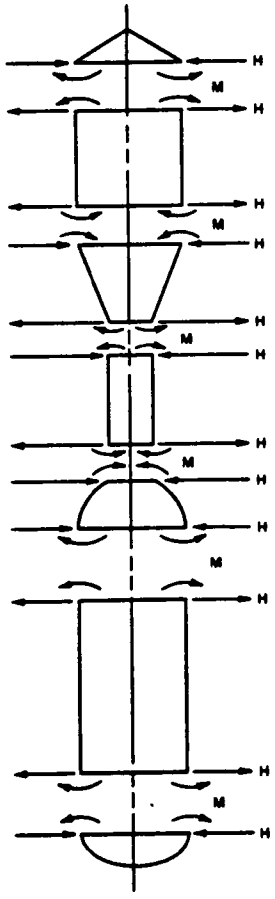


Figure 3-2. Free body diagram of example one.

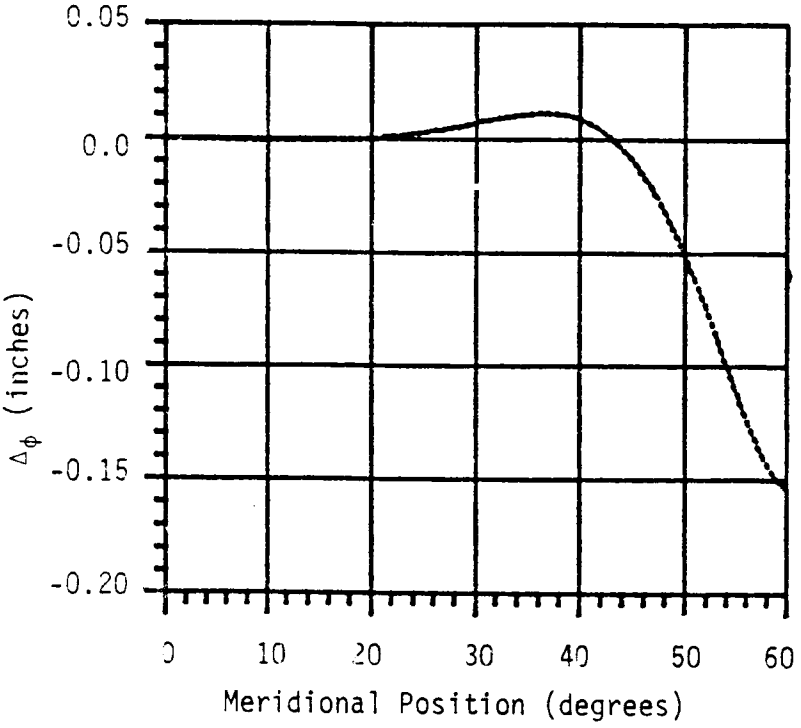


Figure 3-3. Radial deflection of the open crown hemisphere due to discontinuity forces.

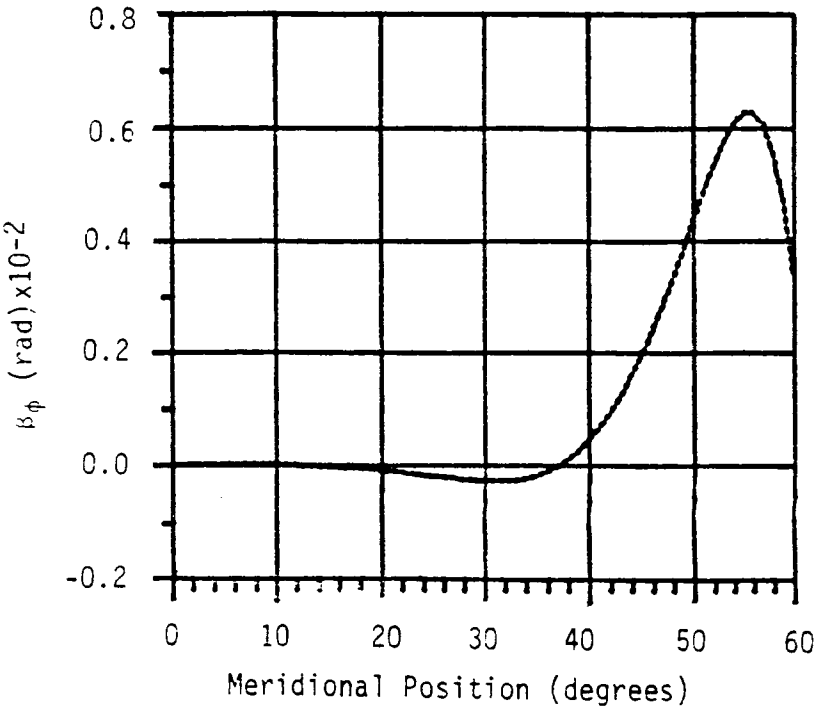


Figure 3-4. Meridional rotation of the open crown hemisphere due to discontinuity forces.

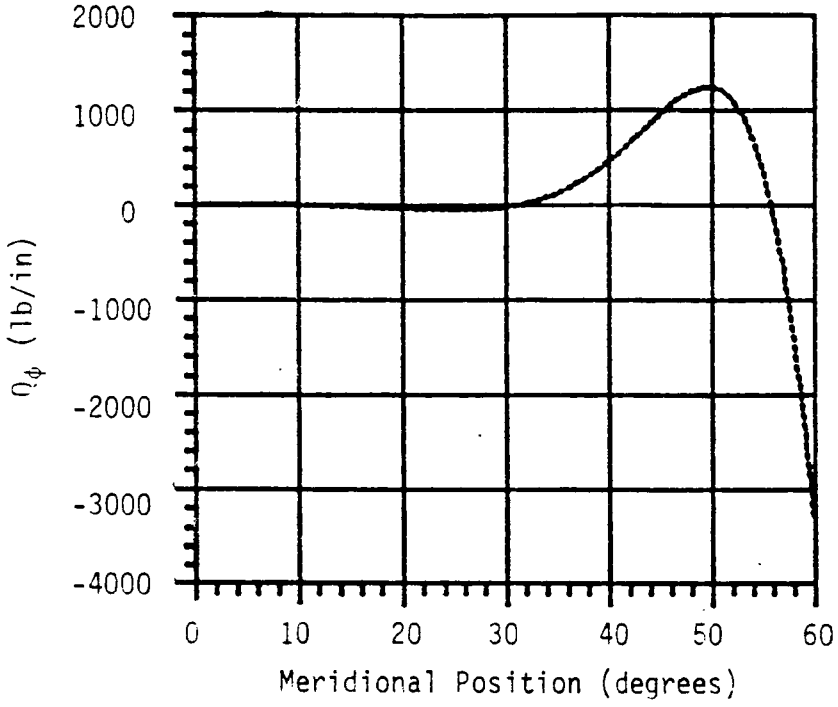


Figure 3-5. Shear stress resultant caused by discontinuity forces.

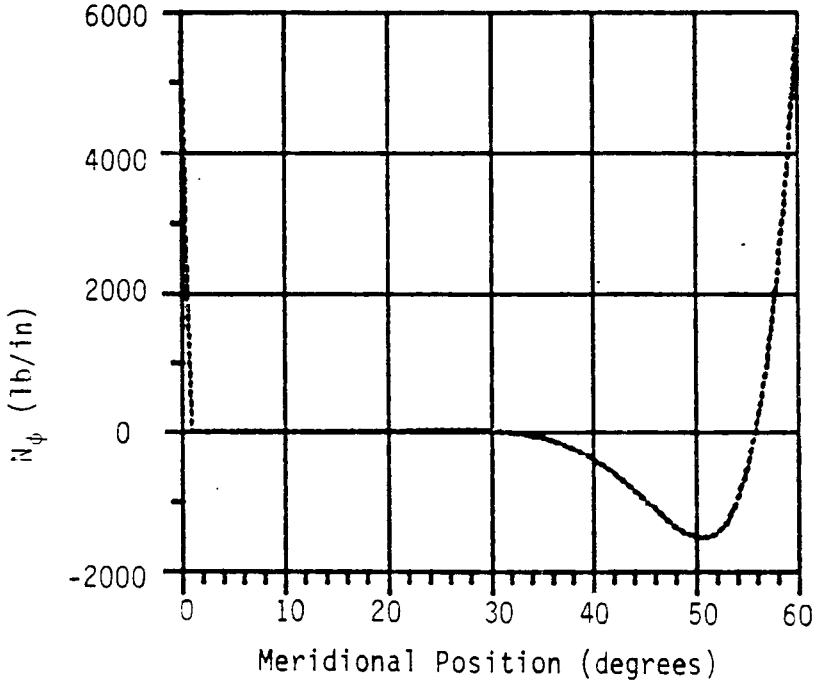


Figure 3-6. Meridional membrane stress resultant caused by discontinuity forces.

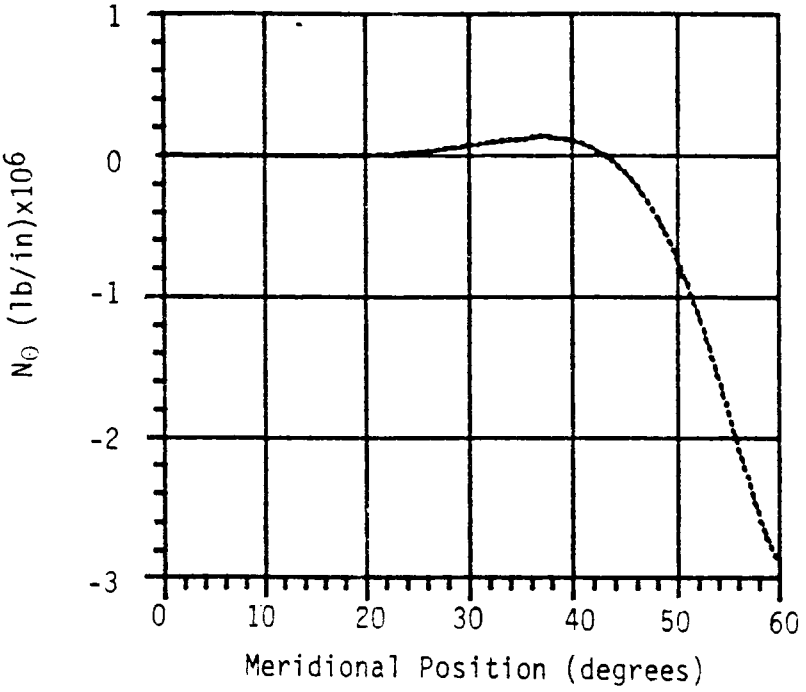


Figure 3-7. Tangential membrane stress resultant due to discontinuity forces.

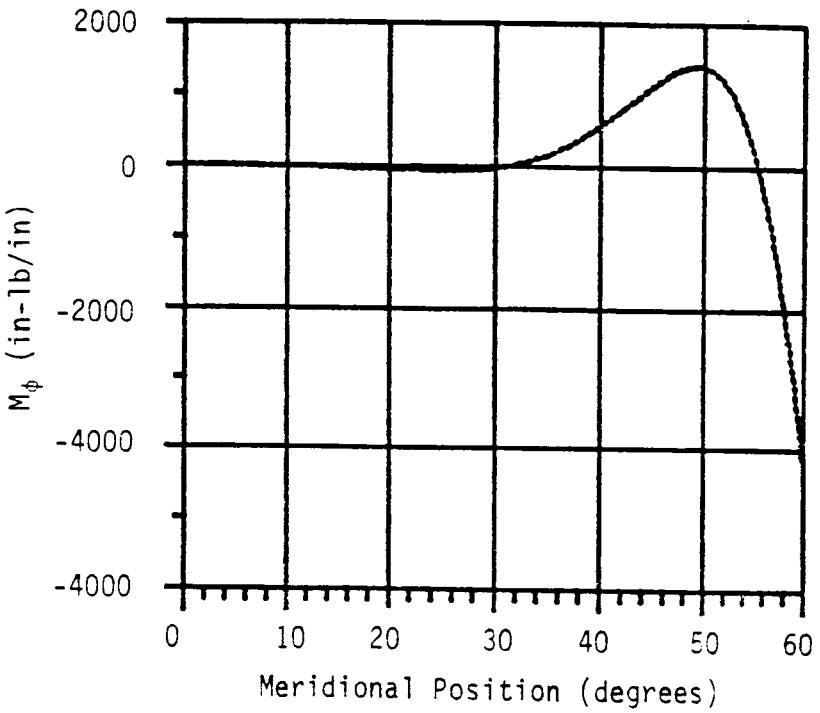


Figure 3-8. Meridional moment due to discontinuity forces.

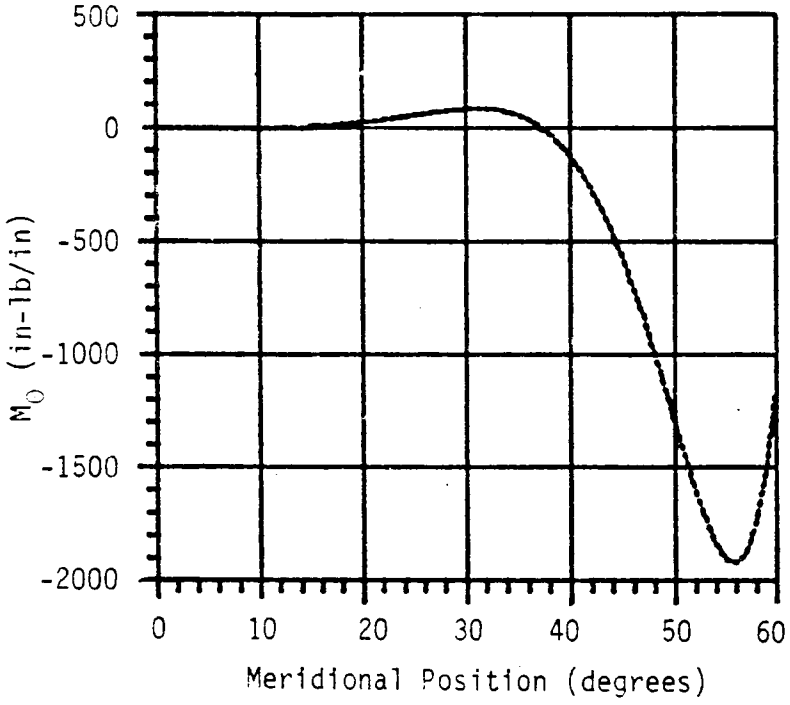


Figure 3-9. Tangential moment of the open crown hemisphere caused by discontinuity forces.

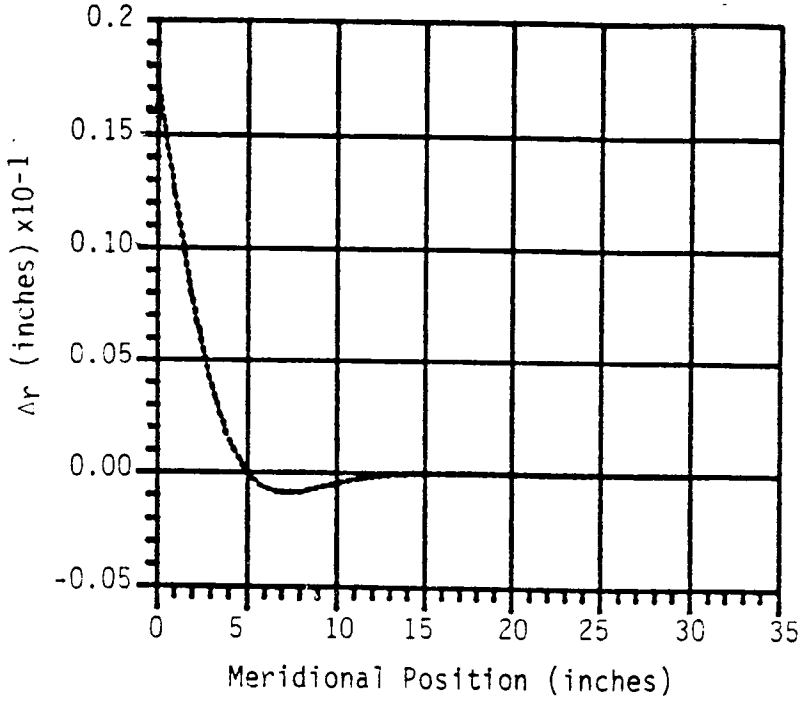


Figure 3-10. Radial deflection of the small cylinder caused by discontinuity forces.

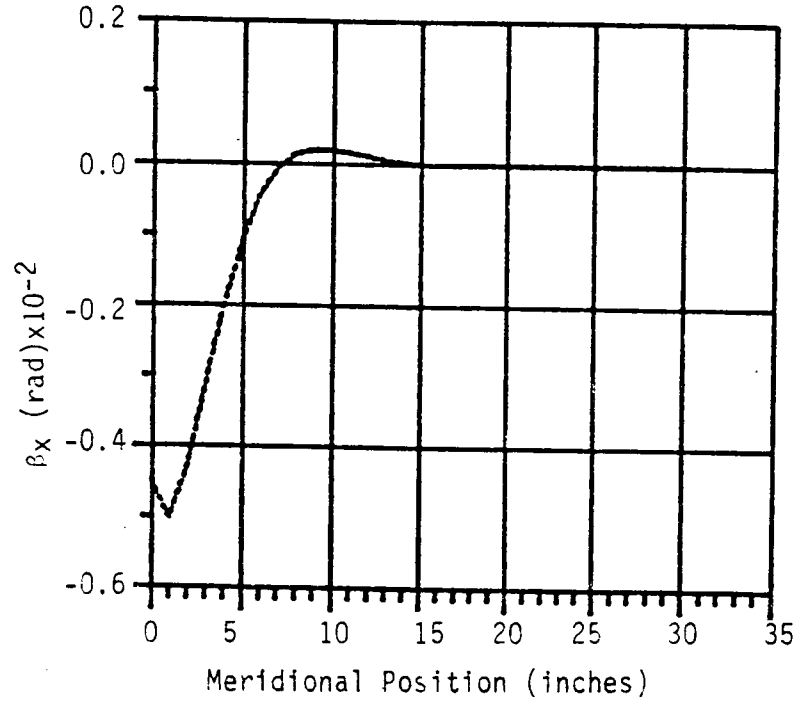


Figure 3-11. Meridional rotation of the small cylinder due to discontinuity forces.

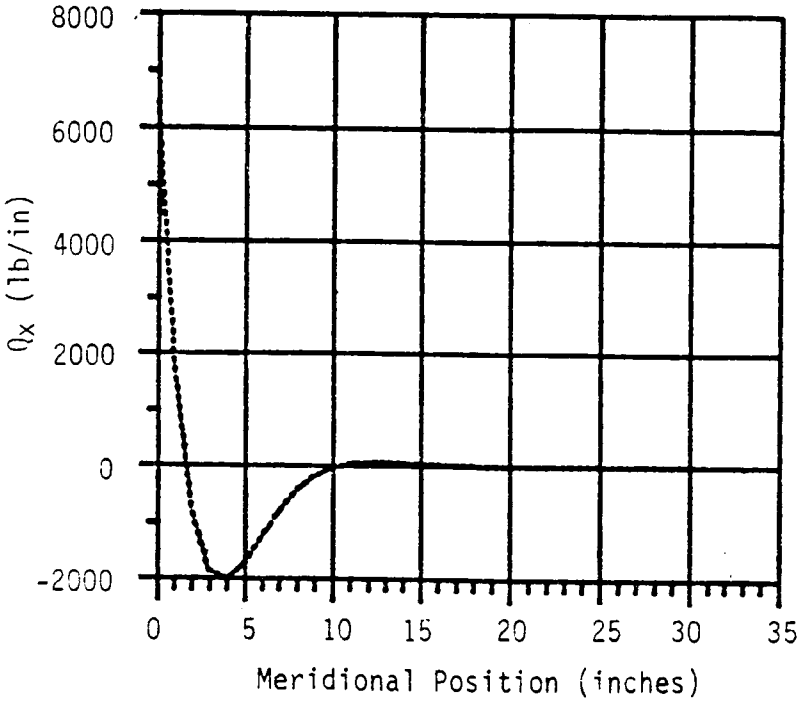


Figure 3-12. Transverse shear stress resultant of the small cylinder caused by discontinuity forces.

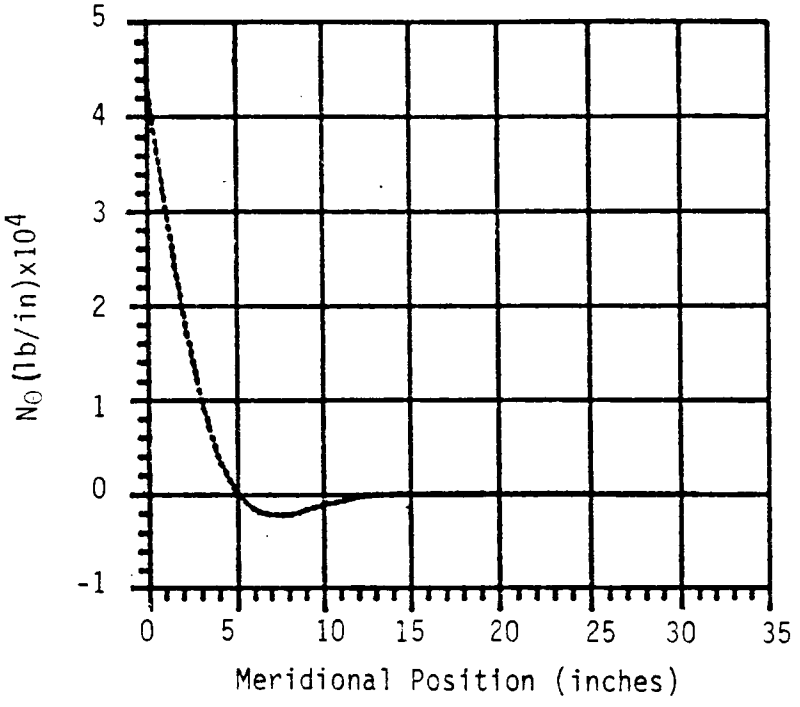


Figure 3-13. Tangential membrane stress resultant for the small cylinder caused by discontinuity forces.

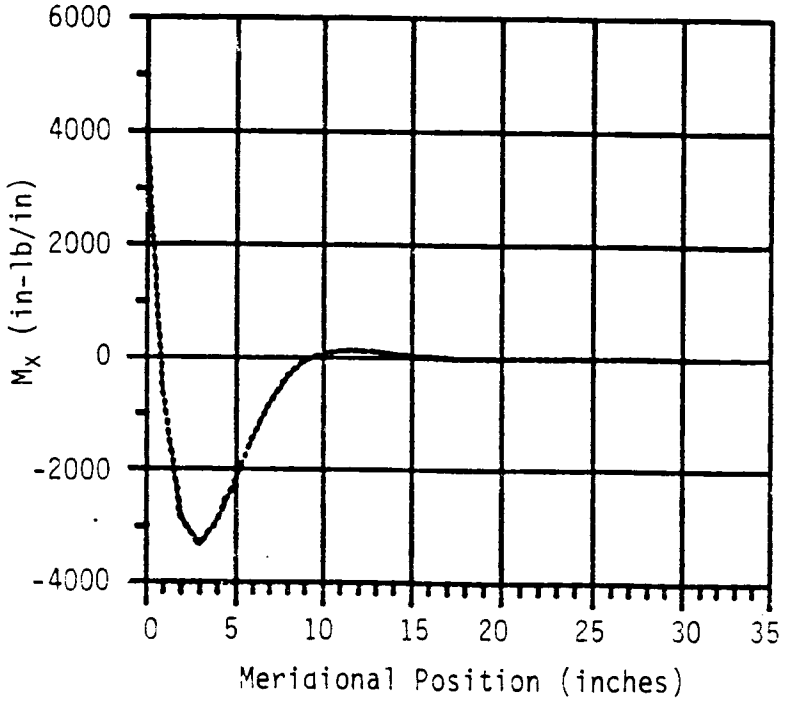


Figure 3-14. Meridional moment for the small cylinder due to discontinuity forces.

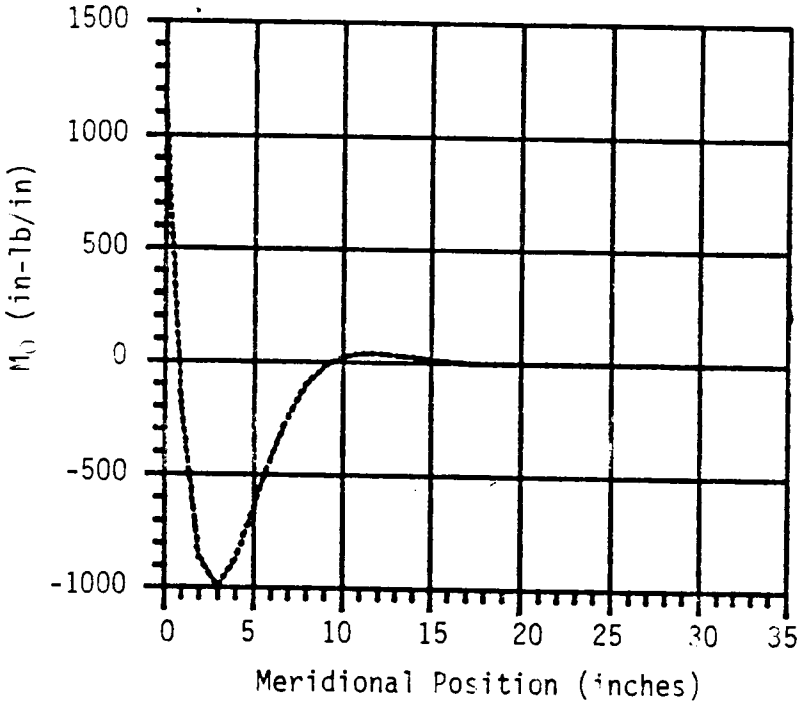


Figure 3-15. Tangential moment for the small cylinder due to discontinuity forces.

Example 2. Determine the discontinuity forces in the water storage tank shown in Fig. 3-16. All components are steel with a Young's Modulus of 29 X 10^6 psi and a Poisson's Ratio of .3.

Solution: The vessel is comprised of a cylindrical shell connected to a stiffening ring and closed hemispherical head. The primary loads are a distributed weight over the hemispherical head which is reacted to the stiffening ring and a linearly varying water pressure acting on the cylinder. The weight of the hemispherical head is 1000 lb/ft^2. The following is the required data file for ALSHEL (note the units and the load type number):

```
TY ALSHEL.DAT

TITL

*******EXAMPLE NO. 2*******

NUMB
3
2 8 3
RAD1
531 531 531
LENT
264 17.5 132
THIC
5 8 6
LOAD
2 2 3
PRES
9.53 6.94 6.94
MATR
29E06 29E06 29E06
.3 .3 .3 .3 .3 .3 .3
```

Figure 3-17 illustrates how ALSHEL models the dome-ring-cylinder joint.

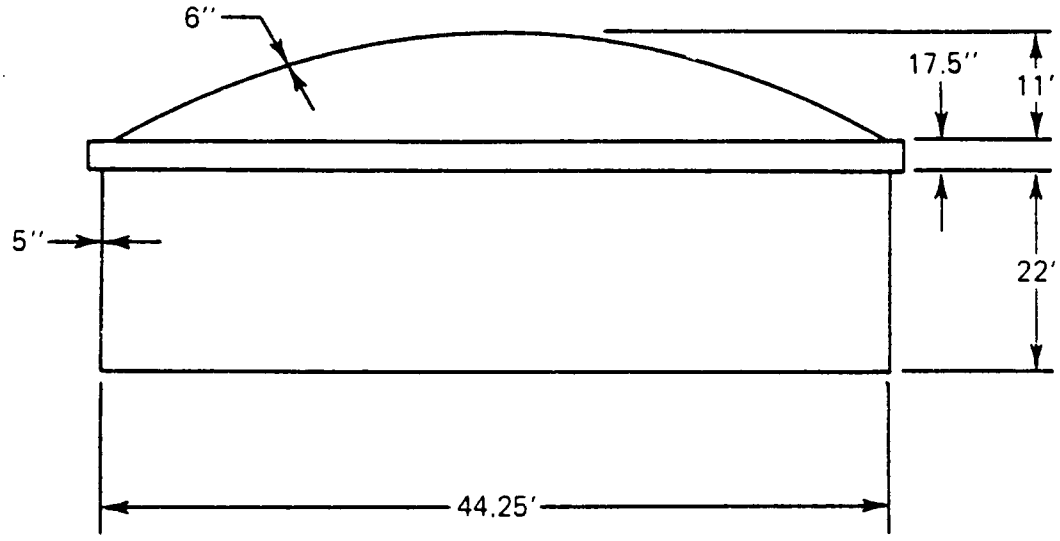


Figure 3-16. Storage tank for example two.

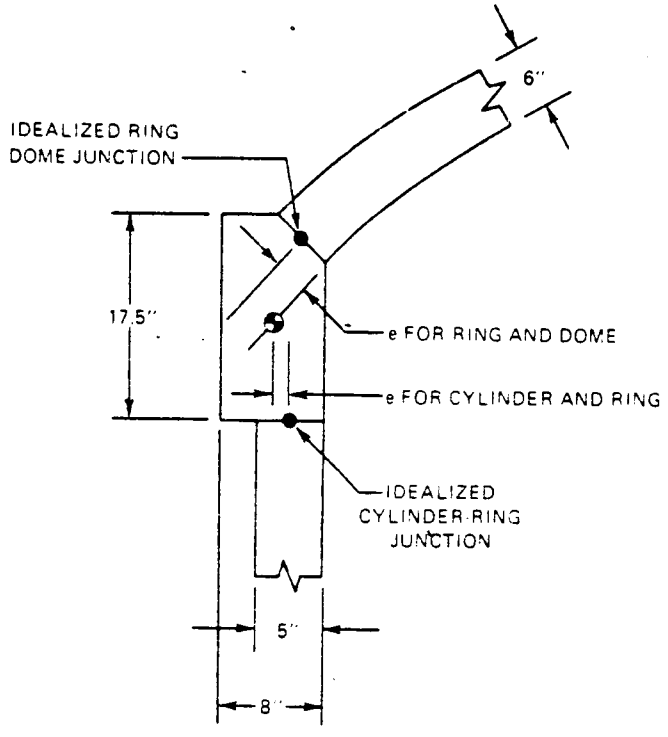


Figure 3-17. Idealized cylinder-ring-dome junction.

The following results were calculated by ALSHEL:

TY RESULT.DAT

*******EXAMPLE NO. 1*******

ALL MOMENTS AND SHEARS ARE IN GLOBAL X-Y COORDINATES

MOMENTS ARE IN (IN-LB/IN) AND SHEARS ARE IN (LB/IN)

CYLINDRICAL SHELL

END 1		END 2	
MOMENT	SHEAR	MOMENT	SHEAR
-8876.2	-413.0	6549.4	-136.6

STIFFENING RING

END 1		END 2	
MOMENT	SHEAR	MOMENT	SHEAR
-6549.4	136.6	16019.1	-1717.0

HEMISPHERICAL HEAD

MOMENT	SHEAR
-16019.1	1717.0

IV. CONCLUSION

ALSHEL AND GRAPH have proved that they can determine the discontinuity forces in a wide variety of multishell structures. However, several improvements could be made that would improve their computational efficiency and applicability.

The most obvious addition would be to expand their geometric and loading capabilities. For instance, open crown ellipsoidal shells, plates with holes in the center and parabolic shells are commonly used in practice. The ability to allow for more than two shells at a junction would also be helpful. Additional loading configurations might include wind loading, thermal loading, support considerations, and the ability to perform dynamic analysis.

To improve computational efficiency, the method described by Baker and Cline could be incorporated for cylindrical and conical shells. This would decrease the number of subroutines. Also, the ability to analyze shells of variable thickness could be easily added. The ability to input multiple data in a single statement would also be valuable.

LIST OF REFERENCES

LIST OF REFERENCES

1. Timoshenko, S. Theory of Plates and Shells, New York: MacGraw-Hill Book Company, 1957.

2. Flugge, W. Stresses in Shells. Berlin: Springer-Verlag, 1960.

3. Baker, B.R. and G. B. Cline, Jr. "Influence Coefficients for Thin Smooth Shells of Revolution Subjected to Symmetric Loads", ASME J. Appl. Mechanics. Volume 29, No. 2, June 1962.

4. Billington, D.P. Thin Shell Concrete Structures. New York: MacGraw-Hill Book Company, 1965.

5. Galletly, G.D. "Influence Coefficients for Open-Crown Hemispheres," ASME J. Eng. Ind., Vol. 82, January 1963.

6. Baker, E. H, L. Kovalevsky, F. L. Rish. Structural Analysis of Shells. New York: Robert E. Kriger Publishing Company, 1981.

7. Roark, Raymond J. Formulas for Stress and Strain. Fourth Edition. New York: Macgraw-Hill Book Company, 1965.

8. Tuma, J. J. Engineering Mathematics Handbook. New York: MacGraw-Hill Book Company, 1970.

9. Akin, J. E. Application and Implementation of Finite Element Method. New York: Academic Press, 1982.

10. Stark, P. A. Introduction to Numerical Methods. New York: Macmillan Publishing Company, 1965.

11. Bednar, H. H. Pressure Vessel Design Handbook. New York: Van Nostrand Rheinhold Company, 1981.

APPENDICES

APPENDIX A
EQUATIONS FOR PROGRAM GRAPH

The equations incorporated into program GRAPH are based on the solution to Eq. (2-70). With the appropriate change of variable [6], these can be applied to all geometries with sufficient accuracy for plotting purposes. The general solution to Eq. (2-70) can be written as combinations of trigonometric and hyperbolic functions. The necessary combinations have been tabulated for convenience in Table A.1.

Table A.1. Trigonometric and Hyperbolic Functions

i	F_i
1	sinh (n) - sin (n)
2	sinh (n) + sin (n)
3	sinh(n)cosh(n) + sin(n)cos(n)
4	sinh(n)cosh(n) - sin(n)cos(n)
5	$\sin^2$ (n)
6	$\sinh^2$ (n)
7	cosh(n)cos(n)
8	sinh(n)sin(n)
9	cosh(n)sin(n) - sinh(n)cos(n)
10	cosh(n)sin(n) + sinh(n)cos(n)
11	sin(n)cos(n)
12	sinh(n)cosh(n)
13	cosh(n)cos(n) - sinh(n)sin(n)
14	cosh(n)cos(n) + sinh(n)sin(n)
15	cosh(n)sin(n)
16	sinh(n)cos(n)
17	exp(- n cos(n))
18	exp(- n sin(n))
19	exp(- n(cos(n) + sin(n)))
20	exp(- n(cos(n) - sin(n)))

Cylindrical Shells

Equations (A.1) through (A.11) show the appropriate solutions for circular cylindrical shells with positive edge disturbances, Figs. A.1 (a) and (b). For cylindrical shells

$$n = x\beta \tag{A.1}$$

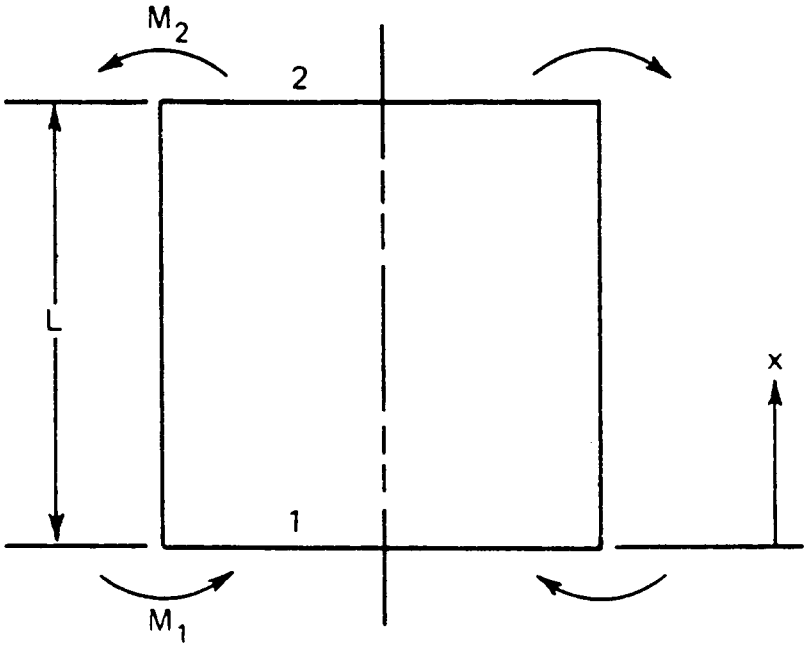
where x = 0 at edge one and x = L at edge two. When the trigonometric functions are not shows as a function of n, then x = L has been substituted.

For loads applied at edge one

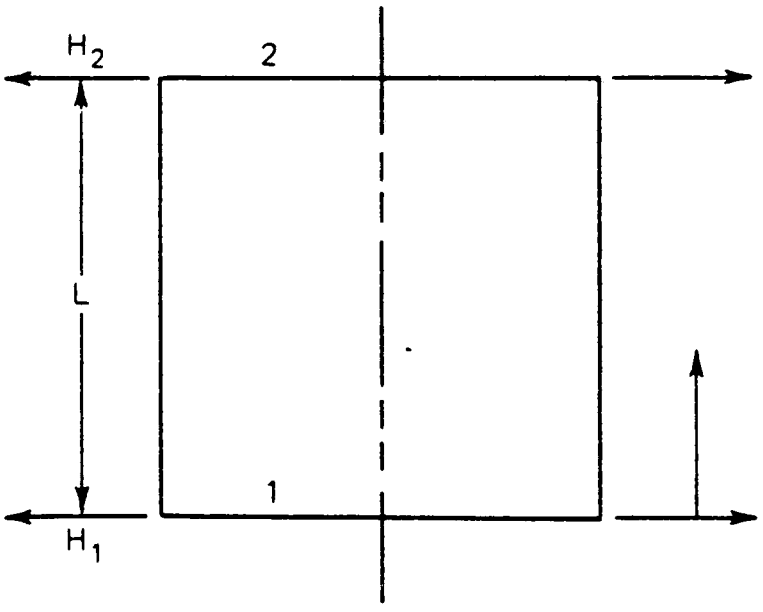
$$N_\theta = -\frac{M_1\beta^2 r}{F_1}\left[-F_2F_7(n) + F_3F_{10}(n) - \frac{F_8(n)}{F_1}\right] + \frac{H_1 2\beta r}{F_1}\left[F_4F_7(n) - F_5F_{15}(n) - F_6F_{16}(n)\right] \tag{A.2}$$

$$M_x = \frac{M_1}{F_1}\left[F_2F_8(n) - F_3F_9(n) - \frac{F_7(n)}{F_1}\right] + \frac{H_1}{F_1}\left[-F_4F_{10}(n) - F_5F_{13}(n) + F_6F_{15}(n)\right] \tag{A.3}$$

$$Q_x = \frac{\beta M_1}{F_1}\left[F_2F_{10}(n) - 2F_3F_8(n) + F_8(n)\right] + \frac{H}{F_1}\left[F_4F_{10}(n) + F_5F_{13}(n) - F_6F_{14}(n)\right] \tag{A.4}$$



(a)



(b)

Figure A-1. Circular cylinder with positive discontinuity forces.

$$\Delta_x = -\frac{M_1}{2D\,\beta^2 F_1}\left[-F_2F_7(n) + F_3F_{10}(n) - \frac{F_8(n)}{F_1}\right] \qquad (A.5)$$

$$+\frac{H_1}{2D\,\beta^3 F_1}\left[F_4F_7(n) - F_5F_{15}(n) - F_6F_{16}(n)\right]$$

$$\theta_x = \frac{M_1}{2D\,\beta F_1}\left[F_2F_9(n) + 2\,F_3F_7(n) - \frac{F_{10}(n)}{F_1}\right] \qquad (A.6)$$

$$-\frac{H_1}{2D\,\beta^2 F_1}\left[F_4F_9(n) + F_5F_{14}(n) + F_6F_{13}(n)\right]$$

For loads applied at edge two

$$N_\theta = \frac{M_2\,2\beta^2 r}{F_1}\left[-2\,F_8F_7(n) + F_{10}F_{10}(n)\right] \qquad (A.7)$$

$$+\frac{H_2\,2\beta r}{F_1}\left[-F_9F_7(n) + F_8F_{10}(n)\right]$$

$$M_x = \frac{M_2}{F_1}\left[2\,F_8F_8(n) - F_{10}F_9(n)\right] \qquad (A.8)$$

$$\frac{H_2}{F_1}\left[F_9F_8(n) - F_8F_9(n)\right]$$

$$Q_x = -\frac{M_2\beta}{F_1}\left[2\,F_8F_{10}(n) - 2\,F_{10}F_8(n)\right] \qquad (A.9)$$

$$-\frac{H_2}{F_1}\left[F_9F_{10}(n) - 2\,F_8F_8(n)\right]$$

$$\Delta_x = \frac{M_2}{2D\,\beta^2 F_1}\left[-2\,F_8F_7(n) + F_{10}F_{10}(n)\right] \qquad (A.10)$$

$$\frac{H_2}{2D\,\beta^3 F_1}\left[-F_9F_7(n) + F_8F_{10}(n)\right]$$

$$\theta_x = \frac{M_2}{2D\ \beta F}\left[2\ F_8F_9(n) + 2\ F_{10}F_7(n)\right] \qquad (A.11)$$

$$\frac{H_2}{2D\beta^2F_1}\left[F_9F_9(n) + 2\ F_8F_7(n)\right]$$

Open and Closed Hemispherical Shells

Equations (A.16) through (A.33) are the appropriate solutions for open and closed hemispherical shells with positive edge disturbances (Figs. A.2-a and A.2-b). For spherical shells

$$\beta^4 = 3(1-\mu^2)(r/t)^2 \tag{A.12}$$

$$n = \beta\alpha \tag{A.13}$$

and

$$\alpha = \phi_1 - \phi \ . \tag{A.14}$$

For factors not written as a function of n

$$\alpha = \phi_1 - \phi_2 \ . \tag{A.15}$$

For loads applied at edge one

$$N_\phi = \frac{H_1 \sin\phi_1 \cot\phi}{F_1} \left[F_1F_7(n) - F_4F_{10}(n) + F_2F_8(n)\right] \tag{A.16}$$

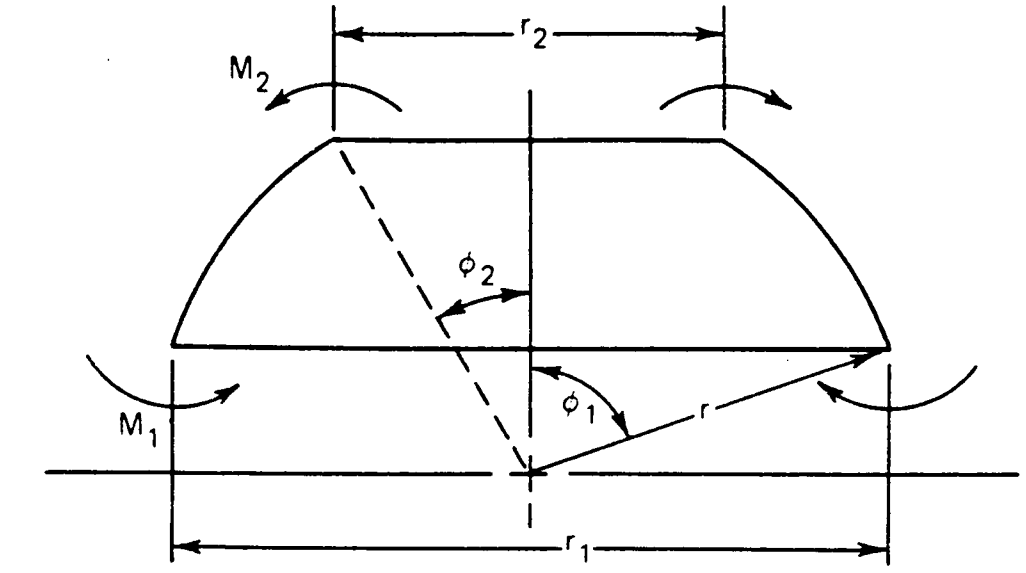
$$+ \frac{M_1 2\beta \cot\phi}{rF_1} \left[F_6F_{15}(n) + F_5F_{16}(n) - F_3F_8(n)\right].$$

$$N_\theta = \frac{H_1 \beta \sin\phi_1}{F_1} \left[- F_1F_9(n) - 2F_4F_7(n) + F_2F_{10}(n)\right] \tag{A.17}$$

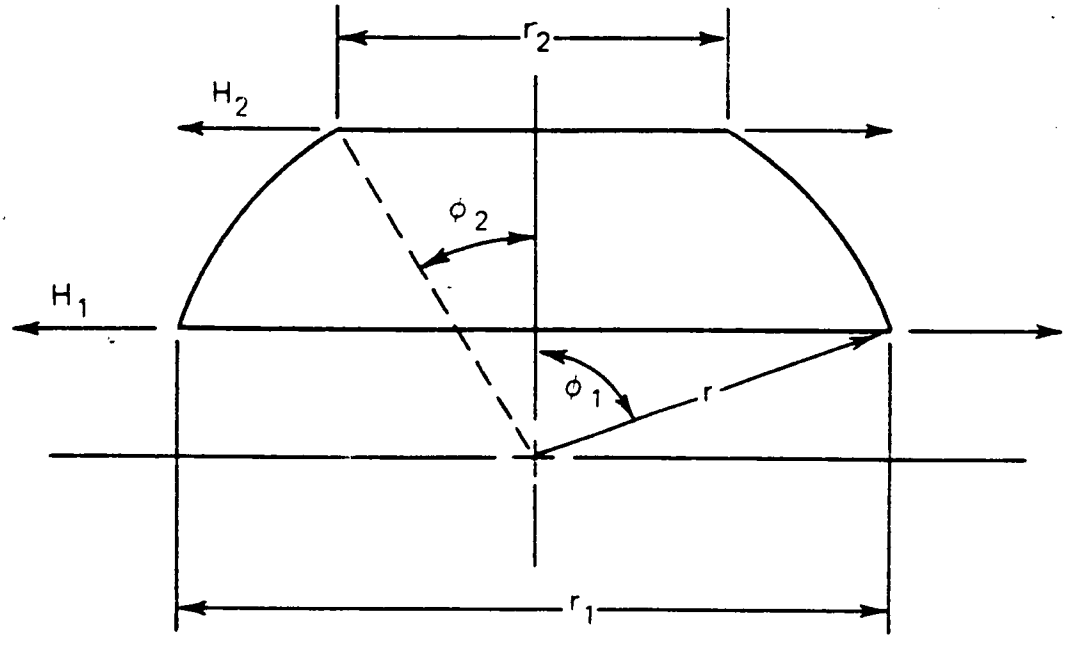
$$+ \frac{M2\beta^2}{rF_1} \left[F_6F_{14}(n) + F_5F_{13}(n) - F_3F_{10}(n)\right].$$

$$Q_\phi = \frac{H_1 \sin\phi_1}{F_1} \left[F_1F_7(n) - F_4F_{10}(n) + F_2F_8(n)\right] \tag{A.18}$$

$$+ \frac{M_1 2\beta}{rF_1} \left[F_6F_{15}(n) + F_5F_{16}(n) - F_3F_8(n)\right].$$



(a)



(b)

Figure A-2. Open crown hemisphere subjected to positive discontinuity forces.

$$M_\phi = \frac{H_1 r \sin\phi_1}{2\beta F_1}\left[- F_1F_{10}(n) + 2\ F_4F_8(n) - F_2F_9(n)\right] \qquad (A.19)$$

$$+ \frac{M_1}{F_1}\left[F_6F_{13}(n) - F_5F_{14}(n) + F_3F_9(n)\right]$$

$$M_\theta = \nu M\phi \qquad (A.20)$$

$$\Delta\phi = \frac{H_1 r \sin\phi\ \sin\phi_1 \beta}{EtF_1}\left[F_1F_9(n) + 2\ F_4F_7(n) - F_2F_{10}(n)\right] \qquad (A.21)$$

$$+ \frac{M_1 2\beta^2\ \sin\phi}{EtF_1}\left[F_6F_{14}(n) + F_5F_{13}(n) - F_3F_{10}(n)\right]$$

$$\theta\phi = \frac{H_1 2\beta^2\ \sin\phi_1}{EtF_1}\left[F_1F_8(n) - F_{5yF15}(n) - F_3F_7(n)\right] \qquad (A.22)$$

$$+ \frac{M4\beta^3}{EtrF_1}\left[F_6F_{16}(n) - F_5F_{15}(n) - F_3F_7(n)\right]$$

For loads applied at edge two

$$N_\phi = \frac{M_2 2\ \cot\phi}{rF_1}\left[F_8F_{10}(n) - F_{10}F_8(n)\right] \qquad (A.23)$$

$$- \frac{H_2\ \cot\phi\ \sin\phi_2}{F_1}\left[F_9F_{10}(n) - 2\ F_8F_8(n)\right]$$

$$N_\theta = \frac{-M_2 2\beta^2}{rF_1}\left[- 2F_8F_7(n) + F_{10}F_{10}(n)\right] \qquad (A.24)$$

$$+ \frac{H_2 2\beta \sin\phi_2}{F_1}\left[- F_9F_7(n) + F_8F_{10}(n)\right]$$

$$Q_\phi = + \frac{M_2 2\beta}{rF_1}\left[- F_8F_{10}(n) + F_{10}F_8(n)\right] \qquad (A.25)$$

$$- \frac{H_2 \sin\phi_2}{F_1}\left[F_9F_{10}(n) - 2\ F_8F_8(n)\right]$$

$$M_\phi = \frac{M_2}{F_1}\left[2\ F_8F_8(n) - F_{10}F_9(n)\right] \qquad (A.26)$$

$$+ \frac{H_2\ \sin\phi_2}{F_1\beta}\left[F_9F_8(n) - F_8F_9(n)\right]$$

$$M_\theta = \nu M\phi \qquad (A.27)$$

$$\Delta\phi = - \frac{M_2 2\beta^2 \sin\phi}{EtF_1}\left[- 2\ F_8F_7(n) + F_{10}F_{10}(n)\right] \qquad (A.28)$$

$$+ \frac{2H_2\ \sin\phi_2 r\ \beta\sin\phi}{EtF_1}\left[F_9F_7(n) - F_8F_{10}(n)\right]$$

$$\theta\phi = \frac{M_2 4\ \beta^3}{EtrF_1}\left[F_8F_9(n) + F_{10}F_7(n)\right] \qquad (A.29)$$

$$- \frac{H_2 \sin\phi_2 2\ \beta^2}{EtF_1}\left[F_9F_9(n) + 2\ F_8F_7(n)\right]$$

Open and Closed Conical Shells

The solutions for open and closed conical sections are given by Eqs. A.30 through A.42. For conical shells, n is given by Eq. (2-82) and

$$n = \bar{x}\beta \tag{A.30}$$

Again, factors not shown as a function of n have been evaluated at $\bar{x} = \bar{L}$ (see Fig. A.3).

$$N_{\bar{x}} = + \frac{M_1 2\ \beta \cot\alpha}{F_1} \left[F_6F_{15}(n) + F_5F_{16}(n) - F_3F_8(n)\right] \tag{A.31}$$

$$- \frac{H_1 \cos\alpha}{F_1} \left[F_1F_7(n) - F_4F_{10}(n) + F_2F_8(n)\right]$$

$$N_\phi = - \frac{M_1 2\beta^2 \bar{x}_1 \cot\alpha}{F_1} \left[F_6F_{14}(n) + F_5F_{13}(n) - F_3F_{10}(n)\right] \tag{A.32}$$

$$+ \frac{H_1 \bar{x}_1 \cos\alpha}{F_1} \left[F_1F_9(n) + 2\ F_4F_7(n) - F_2F_{10}(n)\right]$$

$$M_{\bar{x}} = \frac{M_1}{F_1} \left[F_6F_{13}(n) - F_5F_{14}(n) + F_3F_9(n)\right] \tag{A.33}$$

$$+ \frac{H_1 \sin\alpha}{2\beta F_1} \left[F_1F_{10}(n) - 2\ F_4F_8(n) + F_2F_9(n)\right]$$

$$Q_{\bar{x}} = + \frac{M_1 2\beta}{F_1} \left[F_6F_{15}(n) + F_5F_{16}(n) - F_3F_8(n)\right] \tag{A.34}$$

$$- \frac{H_1 \sin\alpha}{F_1} \left[F_1F_7(n) - F_4F_{10}(n) + F_2F_8(n)\right]$$

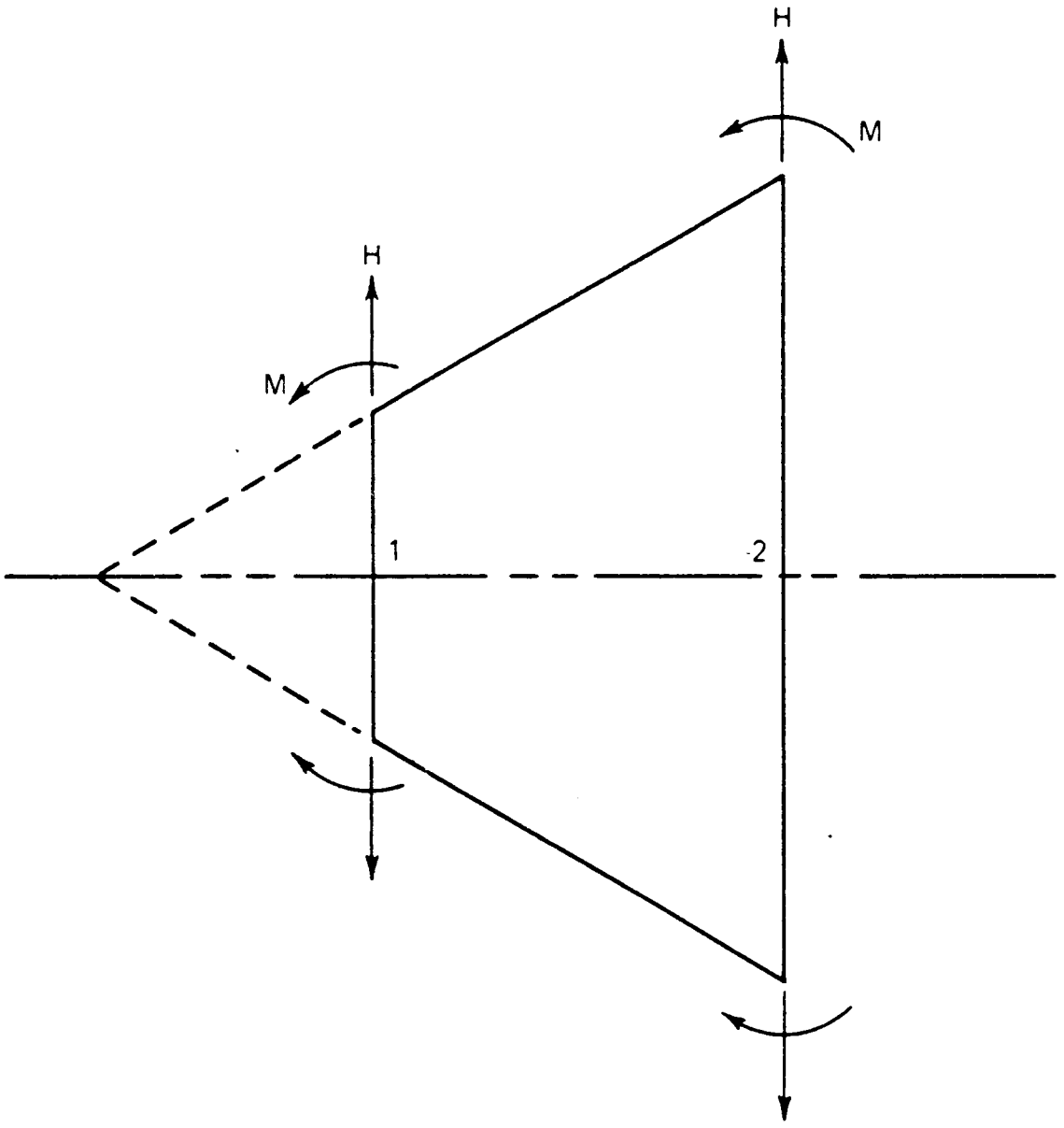


Figure A-3. Conical shell subjected to positive discontinuity forces.

$$\Delta_{\bar{x}} = -\frac{M_1 \sin\alpha}{2D\beta^2 F_1}\left[F_6F_{14}(n) + F_5F_{13}(n) - F_3F_{10}(n)\right] \qquad (A.35)$$

$$+\frac{H_1 \sin^2\alpha}{4D\beta^3 F_1}\left[F_1F_9(n) + 2\,F_4F_7(n) - F_2F_{10}(n)\right]$$

$$\theta_{\bar{x}} = +\frac{M_1}{D\beta F_1}\left[F_6F_{16}(n) - F_5F_{15}(n) - F_3F_7(n)\right] \qquad (A.36)$$

$$-\frac{H_1 \sin\alpha}{2D\beta^2 F_1}\left[-F_1F_8(n) + F_4F_9(n) + F_2F_7(n)\right]$$

For loads applied at edge two

$$N_{\bar{x}} = +\frac{M_2 2\beta\cot\alpha}{F_1}\left[F_8F_{10}(n) - F_{10}F_8(n)\right] \qquad (A.37)$$

$$+\frac{H_2 \cos\alpha}{F_1}\left[-F_9F_{10}(n) + 2\,F_8F_8(n)\right]$$

$$N_{\phi} = \frac{M_2 2\,\beta^2\,\bar{x}_2\cot\alpha}{F_1}\left[2\,F_8F_7(n) - F_{10}F_{10}(n)\right] \qquad (A.38)$$

$$+\frac{2H_2\,\bar{x}_2\cos\alpha}{F_1}\left[-F_9F_7(n) + F_8F_{10}(n)\right]$$

$$M_{\bar{x}} = \frac{M_2}{F_1}\left[2\,F_8F_8(n) - F_{10}F_9(n)\right] \qquad (A.39)$$

$$+\frac{H_2 \sin\alpha}{F_1}\left[F_9F_8(n) - F_8F_9(n)\right]$$

$$Q_{\bar{x}} = \frac{M_2 2\beta}{F_1}\left[F_8F_{10}(n) - F_{10}F_8(n)\right] \qquad (A.40)$$

$$+\frac{H_2 \sin\alpha}{F_1}\left[-F_9F_{10}(n) + 2\,F_8F_8(n)\right]$$

$$\Delta_{\overline{x}} = -\frac{M_2 \sin\alpha}{2D\beta^2 F_1}\left[2 F_8F_7(n) - F_{10}F_{10}(n)\right] + \frac{H_2 \sin^2\alpha}{2\beta^3 DF_1}\left[- F_9F_7(n) + F_8F_{10}(n)\right] \quad (A.41)$$

$$\theta_{\overline{x}} = -\frac{M_2}{D\beta F_1}\left[F_8F_9(n) + F_{10}F_7(n)\right] + \frac{H_2 \sin\alpha}{2D\beta^2 F_1}\left[F_9F_9(n) + 2 F_8F_7(n)\right] \quad (A.42)$$

APPENDIX B

THOMPSON FUNCTIONS

Equations (2-46) and (2-47) can be written in the following form [2]:

$$L[L(U) + i\mu^2 U] - i\mu^2[L(U) + i\mu^2 U] = 0 \tag{B.1}$$

$$L[L(U) - i\mu^2 U] + i\mu^2[L(U) - i\mu^2 U] = 0 \tag{B.2}$$

which indicates that solutions to the second order equations

$$L(U) + i\mu^2 U = 0 \tag{B.3}$$

$$L(U) - i\mu^2 U = 0 \tag{B.4}$$

are also solutions to (B.1) and (B.2).

Introducing the variables

$$n = y\sqrt{i} \quad = \quad 2\mu\sqrt{\frac{ix}{r}} \tag{B.5}$$

$$m = U\sqrt{\frac{x}{r}} \tag{B.6}$$

Eq. (B.3) can be written as

$$n^2 \frac{d^2 m}{dn^2} + n \frac{dn}{dm} + (n^2 - 1)m = 0 \cdot \tag{B.7}$$

The solutions are Bessel's functions of the first order of m

$$m = AJ_1(n) + BH_1^{(1)}(n) \cdot \tag{B.8}$$

These are related to the zero order functions by

$$J_1(n) = \frac{d\ Jo(n)}{dn} \tag{B.9}$$

$$H_1^{(1)}(n) = \frac{d\ Ho^{(1)}(n)}{d(n)} \tag{B.10}$$

The real and imaginary parts of Jo and $Ho^{(1)}$ can be considered as real functions of y, or

$$Jo(n) = Jo(y\ \ i) = ber(y) - i\ bei(y) \tag{B.11}$$

$$Ho^{(1)}(n) = Ho^{(1)}(y\ \ i) = -\frac{2}{\pi}(kei(y) + i\ ker(y) \tag{B.12}$$

Differentiating with respect to n and separating real and imaginary parts gives

$$J_1(n) = \frac{1}{2}(bei'(y) - ber'(y)) + i(bei'(y) + ber'(y)) \tag{B.13}$$

$$H_1^{(1)}(n) = \frac{2}{\pi}(ker'(y) + kei'(y)) + i(ker'(y) - kei'(y)) \tag{B.14}$$

where bei, ber, ker, and kei are the Thomson functions and the prime denotes differentiating with respect to y.

The Thomson functions can be represented by the following series

$$\text{ber}(y) = 1 - \frac{(g)^4}{(2!)^2} + \frac{(g)^8}{(4!)^2} - \frac{(g)^{12}}{(6!)^2} + \cdots \quad \text{(B.15)}$$

$$\text{bei}(y) = \frac{(g)^2}{(1!)^2} - \frac{(g)^6}{(3!)^2} + \frac{(g)^{10}}{(5!)^2} - \cdots \quad \text{(B.16)}$$

$$\text{ker}(y) = -(\ln(g) + C)\,\text{ber}(y) + \frac{\pi}{4}\,\text{bei}(y) - \lambda_2 + \lambda_4 + \cdots \quad \text{(B.17)}$$

$$\text{kei}(y) = -(\ln(g) + C)\,\text{bei}(y) - \frac{\pi}{4}\,\text{ber}(y) + \lambda_1 - \lambda_3 - \cdots \quad \text{(B.18)}$$

$$\text{ber}'(y) = -\frac{2(g)^3}{(2!)^2} + \frac{4(g)^7}{(4!)^2} - \frac{6(g)^{11}}{(6!)^2} + \cdots \quad \text{(B.19)}$$

$$\text{bei}'(y) = \frac{(g)}{(1!)^2} - \frac{3(g)^5}{(3!)^2} + \frac{5(g)^9}{(5!)^2} - \cdots \quad \text{(B.20)}$$

$$\text{ker}'(y) = -(\ln(g) + C)\text{ber}'(y) - \left(\frac{1}{y}\right)\text{ber}(y) + \frac{\pi}{4}\,\text{bei}(y) - \lambda'_2 + \lambda'_4 - \cdots \quad \text{(B.21)}$$

$$\text{kei}'(y) = -(\ln(g) + C)\,\text{bei}'(y) - \left(\frac{1}{y}\right)\text{bei}(y) - \frac{\pi}{4}\,\text{ber}(y) + \lambda'_1 - \lambda'_3 + \cdots \quad \text{(B.22)}$$

where

$$\lambda_k = \frac{(g)^{2k}}{(k!)^2}\left(1 + \frac{1}{2} + \frac{1}{3} + \cdots \frac{1}{k}\right) \qquad \text{(B.23)}$$

$$\lambda'_k = \frac{d\lambda_k}{dy}$$

$$C = 0.57721$$

$$g = \frac{y}{2}$$

Figures B.1 through B.4 show the characteristics of the Thomson function. This series is taken out to K = 8 in subroutine KELVIN.

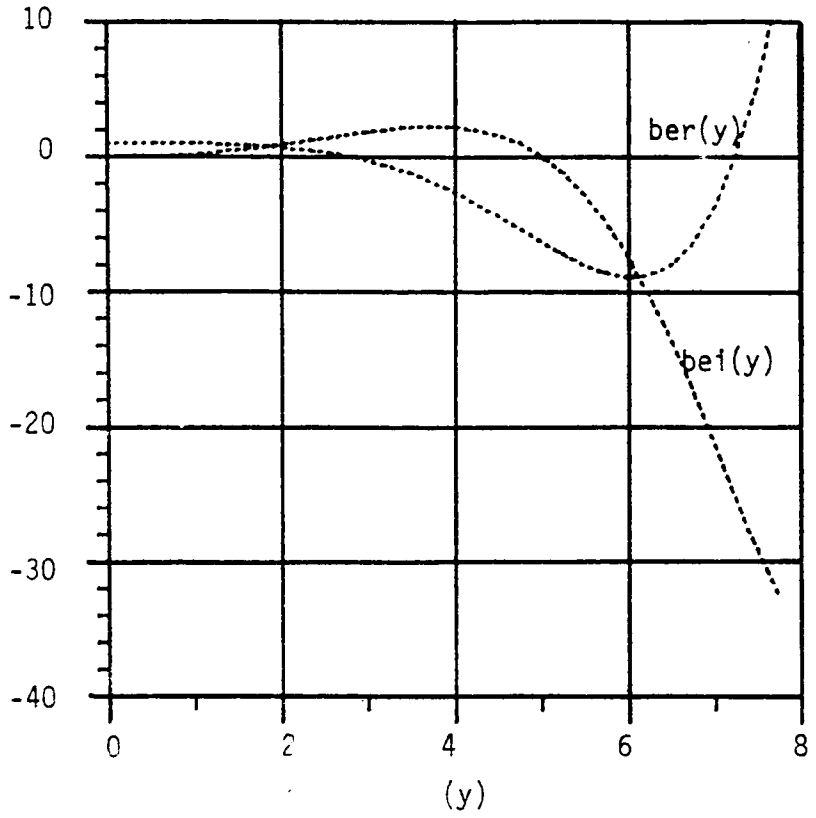


Figure B-1. Characteristics of ber(y) and bei(y).

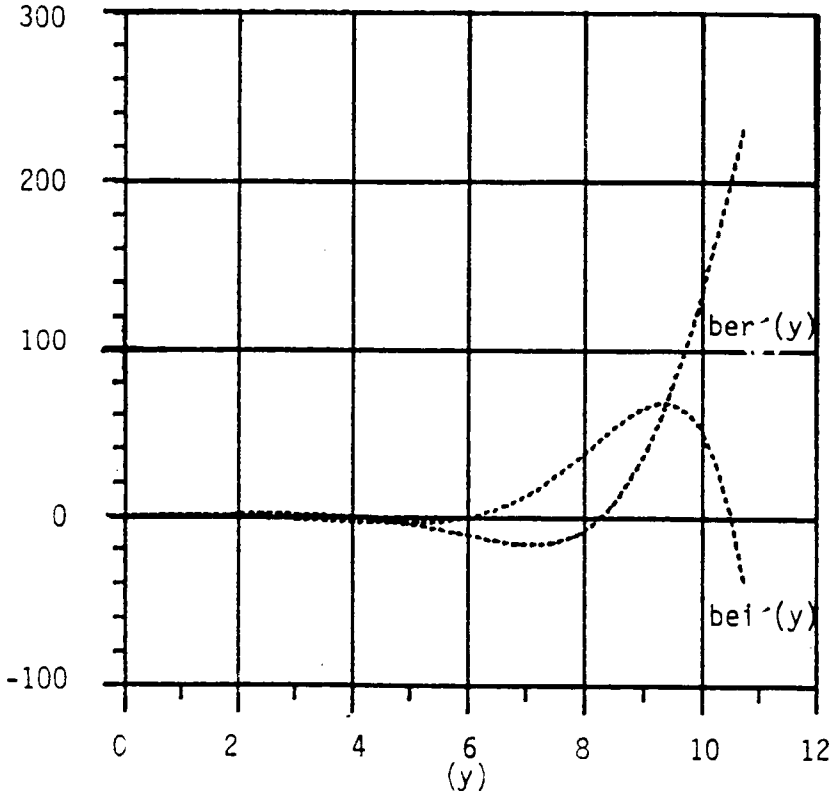


Figure B-2. Characteristics of ber´(y) and bei´(y).

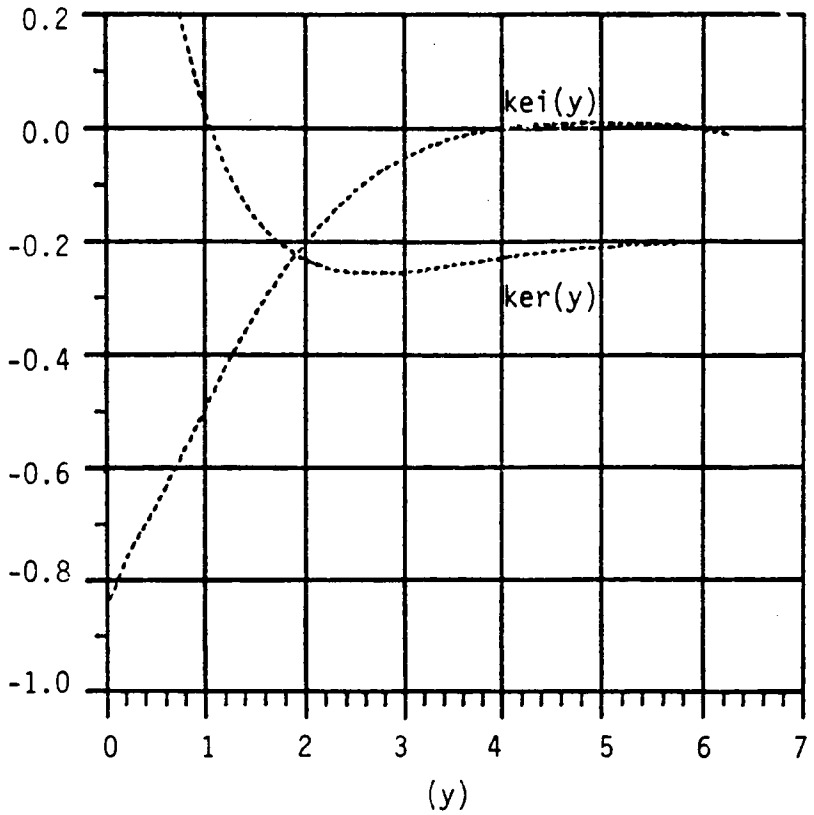


Figure B-3. Characteristics of ker(y) and kei(y)

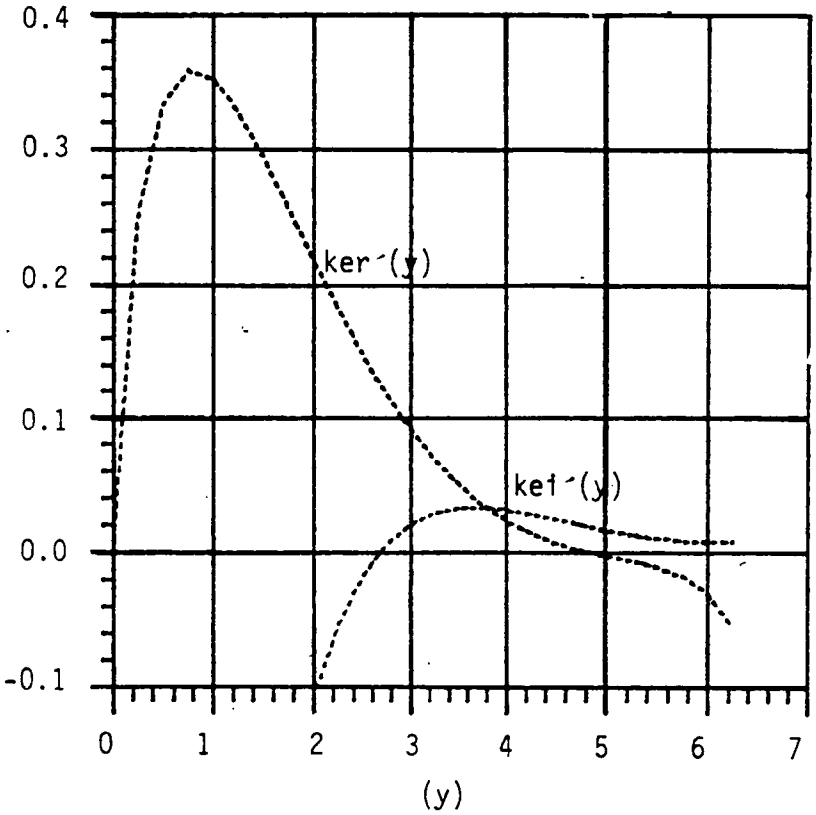


Figure B-4. Characteristics of ker'(y) and kei'(y).

APPENDIX C

NUMERICAL INTEGRATION FOR ELLIPTIC INTEGRALS

In the solution for the influence coefficients of ellipsoidal shells, evaluation of the elliptic integral of the second kind is required. One of the most accurate and simplest numerical schemes is the method of Gaussian Quadratures. Instead of dividing the region into several intervals and approximating the area of each interval, Gauss considered the problem of determining which values of an integral should be chosen to give the greatest possible accuracy. For example, for a given definite integral

$$I = \int_a^b f(X)dX \qquad C.1$$

what values of X should be chosen and at what intervals to give the greatest accuracy? Gauss determined that if "n" points were chosen, they should not be equally spaced but should be symmetrically placed with respect to the mid-point of the region. If the region is the symmetric, then the ratio of the spacing would also be symmetric. Therefore, the limits of integration are taken from -1 to +1 for any function. For instance, if a change of variable of

$$x(r) = 1/2(b-a)r + 1/2\ (b-a) \qquad C.2$$

is introduced, then

$$f(x) = f\left[1/2(b-a)r + 1/2(b-a)\right] \quad (r) \qquad C.3$$

and noting from C.2

$$dx = 1/2(b-a)\ dr \qquad C.4$$

then C.1 becomes

$$I = 1/2(b-a)\int_{-1}^{+1} f(x(r))dr\ . \qquad C.5$$

Gauss showed that

$$\int_{-1}^{+1} f(x(r))dr = \sum_{i=1}^{n} W_i \, \emptyset \, (r_i) \qquad C.6$$

where W_i are weight functions (or ratios) and r_i are the associated abscissae for n points in the interval (-1, +1). For n = 6

$$-r_1 = r_6 = 0.9324695142$$

$$-r_2 = r_5 = 0.6612093865$$

$$-r_3 = r_4 = 0.2386191861$$

and

$$W_1 = W_6 = 0.1713244924$$

$$W_2 = W_5 = 0.3607615730$$

$$W_3 = W_4 = 0.4679139346$$

The elliptic integral of the second kind is

$$E(k,\emptyset) = \int_0^{\phi} 1-k^2\sin^2\emptyset \; d\emptyset \qquad C.7$$

where

$$k = \sin\omega \; . \qquad C.8$$

If the change of variable

$$\phi = \frac{(X+1)\alpha}{2} \qquad C.9$$

is introduced, Eq. C.7 becomes

$$E(k,0) = \int_{-1}^{1} 1-k^2 \sin \frac{(X+1\alpha)}{2} \frac{\alpha}{2} \, dX \qquad C.10$$

which is in suitable form for Gaussian Quadrature.

APPENDIX D

FORTRAN LISTING OF ALSHEL

AND GRAPH

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                          C
C                                                          C
C   PROGRAM ALSHEL                  BY: BRIAN G. SMITH     C
C                                                          C
C   ALSHEL IS A MULTY PURPOSE PROGRAM TO DETERMINE         C
C   DISCONTINUITY FORCES IN MULTISHELL STRUCTURES.         C
C                                                          C
C                                                          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
      DIMENSION STFMAT(50,50),DEFLEC(50),IP(50),IN(50,2),PIE(50)
      COMMON/ARRAYS/ITYPE(10),RAD1(10),RAD2(10),XLENT(10),LOAD(10),
     $              PRES(10),THIC(50),XNU(10),E(10),ELDELF(4),C(50),
     $              FLEX(4,4)
      COMMON/CONTS/PI,IORDER,NUMELE,COUNT,IDIM
      DIMENSION NAMES(10),TITLE(20)
      DATA NAMES/4HTITL,4HNUMB,4HTYPE,4HRAD1,4HRAD2,
     $           4HLENT,4HTHIC,4HLOAD,4HPRES,4HMATR/
      OPEN(UNIT=20,FILE='ALSHEL.DAT',DEVICE='DSK')
      OPEN(UNIT=12,FILE='CYLIND.DAT',DEVICE='DSK')
      OPEN(UNIT=11,FILE='SPHERE.DAT',DEVICE='DSK')
      OPEN(UNIT=21,FILE='ASSEMB.DAT',DEVICE='DSK')
      OPEN(UNIT=23,FILE='HEMI.DAT',DEVICE='DSK')
      OPEN(UNIT=24,FILE='INVERT.DAT',DEVICE='DSK')
      OPEN(UNIT=25,FILE='UPNCON.DAT',DEVICE='DSK')
      OPEN(UNIT=26,FILE='PLATE.DAT',DEVICE='DSK')
      OPEN(UNIT=27,FILE='CLSCON.DAT',DEVICE='DSK')
      OPEN(UNIT=28,FILE='RING.DAT',DEVICE='DSK')
      OPEN(UNIT=30,FILE='RESULT.DAT',DEVICE='DSK')
      OPEN(UNIT=31,FILE='GRAPH.DAT',DEVICE='DSK')
      OPEN(UNIT=32,FILE='ELLIPS.DAT',DEVICE='DSK')
      OPEN(UNIT=33,FILE='GAUSS.DAT',DEVICE='DSK')
      OPEN(UNIT=34,FILE='KELVIN.DAT',DEVICE='DSK')
      PI=3.141592654
      COUNT=0.0
      DO 2000 I=1,2
      READ(20,5000) NAME
      IF(NAME.NE.NAMES(I)) GOTO 2010
      GOTO(2020,2030),I
C
```

```
C
C
C       READING JOB TITLE
C
C
C
 2020   READ(20,5010) TITLE
        WRITE(24,5020) NAME
        WRITE(30,5030) TITLE
C
C
C
C       READING NUMBER OF SHELLS
C
C
C
 2030   CONTINUE
 2030   READ(20,*) NUMELE
        IDIM=2.*NUMELE+2.
        NN=7
        K=0
 1       K=K+1
        IF(K.GT.NN) GOTO 2140
        READ(20,5000) NAME
        DO 2040 II=3,10
        IF(NAME.EQ.NAMES(II)) GOTO 2050
 2040   CONTINUE
 2050   M=II-2
        GOTO(2060,2070,2080,2090,2100,2110,2120,2130),M
C
C
C       (1.)  READING THE SHELL TYPES
C
 2060   CONTINUE
        READ(20,*),(ITYPE(J),J=1,NUMELE)
        GOTO 1
C
C
C       (2.)  READING THE PRIMARY RADII (RAD1)
C
 2070   CONTINUE
        READ(20,*),(RAD1(J),J=1,NUMELE)
        GOTO 1
C
C
C       (3.)  READING ANY SECONDARY RADII (RAD2)
C
 2080   CONTINUE
        NN=8
 3      DO 2 J=1,NUMELE
        IF(ITYPE(J).EQ.1.OR.ITYPE(J).EQ.4) READ(20,*) RAD2(J)
 2      CONTINUE
        GOTO 1
C
C
```

```
C      (4.)  READING AXIAL LENGTH OF THE SHELLS
C
2090   CONTINUE
       READ(20,*),(XLENT(J),J=1,NUMELE)
       GOTO 1
C
C
C      (5.)  READING THE SHELL THICKNESSES
C
2100   CONTINUE
       READ(20,*),(THIC(J),J=1,NUMELE)
       GOTO 1
C
C
C     (6.)  READING THE TYPE OF EXTERNAL LOADING
C
2110   CONTINUE
       READ(20,*),(LOAD(J),J=1,NUMELE)
       GOTO 1
C
C
C      (7.)  READING THE MAXIMUM PRESSURE
C
2120   CONTINUE
       READ(20,*),(PRES(J),J=1,NUMELE)
       GOTO 1
C
C
C      (8.)  READING MATERIAL PROPERTIES I.E. E AND NU
C
2130   CONTINUE
       READ(20,*),(E(J),J=1,NUMELE)
       READ(20,*),(XNU(J),J=1,NUMELE)
       GOTO 1
2140   CONTINUE
       IORDER=0
       DO 15 I=1,NUMELE
       IORDER=IORDER+1
       KK=ITYPE(I)
       GOTO(100,200,300,400,500,600,700,800),KK
100    CALL SPHERE(STFMAT,DEFLEC,IP,IN,PIE)
       GOTO 15
200    CALL CYLIND(STFMAT,DEFLEC,IP,IN,PIE)
       GOTO 15
300    CALL HEMI(STFMAT,DEFLEC,IP,IN,PIE)
       GOTO 15
400    CALL CONCON(STFMAT,DEFLEC,IP,IN,PIE)
       GOTO 15
500    CALL PLATE(STFMAT,DEFLEC,IP,IN,PIE)
       GOTO 15
600    CALL CLSCON(STFMAT,DEFLEC,IP,IN,PIE)
       GOTO 15
700    CALL ELLIPS(STFMAT,DEFLEC,IP,IN,PIE)
       GOTO 15
800    CALL RING(STFMAT,DEFLEC,IP,IN,PIE)
```

```
      GOTO 15
15    CONTINUE
      GOTO 1000
5000  FORMAT(A4)
5010  FORMAT(20A4)
5020  FORMAT(/1X,A4)
5030  FORMAT(/1X,20A4)
2010  WRITE(5,5040)
5040  FORMAT(//2X,'DATA MODULES ARE OUT OF ORDER')
1000  CONTINUE
      STOP
      END
C
C
C
C
C
C
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                    C
C                                                                    C
C        SUBROUTINE CYLIND CALCULATES THE INFLUENCE                  C
C        FOR SHORT AND LONG THIN CIRCULAR CYLINDRICAL SHELLS         C
C        THIN SHELLS HAVE RADIUS TO THICKNESS RATIO LARGER           C
C        THAN 10.  LONG SHELLS ARE DEFINED AS BETA*LENGTH>5.         C
C        ALL ENDS ARE ASSUMED TO BE FIXED.                            C
C                                                                    C
C                                                                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C
C
C
C
C
      SUBROUTINE CYLIND(STFMAT,DEFLEC,IP,IN,PIE)
      COMMON/ARRAYS/ITYPE(10),RAD1(10),RAD2(10),XLENT(10),LOAD(10),
     $              PRES(10),THIC(50),XNU(10),E(10),ELDELF(4),U(50),
     $              FLEX(4,4)
      COMMON/CONTS/PI,IORDER,NUMELE,COUNT,IDIM
      DIMENSION STFMAT(IDIM,IDIM),DEFLEC(IDIM),IP(IDIM),IN(IDIM,2)
      DIMENSION PIE(IDIM)
C
C
C     INITIALIZING FLEX
      DO 11 NI=1,4
      DO 11 NJ=1,4
11    FLEX(NI,NJ)=0.0
C
C
C
```

```
C       CALCULATING  GEOMETRIC AND MATERIAL CONSTANTS
C
C
      XK=((3.*(1.-XNU(IORDER)**2.))**.25)/(RAD1(IORDER)*THIC
     $(IORDER))**.5
      BETA=XK*XLENT(IORDER)
      D=(E(IORDER)*THIC(IORDER)**3.)/(12.*(1.-XNU(IORDER)**2.))
      IF(LOAD(IORDER).EQ.1) GOTO 5
      IF(LOAD(IORDER).EQ.2) GOTO 15
C
C     PRIMARY SOLUTIONS
C
C
C
C
C     UNIFORM PRESSURE SOLUTIONS
C
5     CONTINUE
      ELDELF(1)=0.0
      ELDELF(2)=(PRES(IORDER)*RAD1(IORDER)**2.)/(E(IORDER)*THIC(IORDER)
     $)*(1.-(XNU(IORDER)/2.))
      ELDELF(3)=0.0
      ELDELF(4)=(PRES(IORDER)*RAD1(IORDER)**2.)/(E(IORDER)*THIC(IORDER)
     $)*(1.-(XNU(IORDER)/2.))
      GOTO 16
C
C
C     HYDROSTATIC PRESSURE SOLUTIONS
C
15    CONTINUE
      ELDELF(1)=(PRES(IORDER)*RAD1(IORDER)**2)/(E(IORDER)*THIC(IORDER)
     $*XLENT(IORDER))
      ELDELF(2)=(PRES(IORDER)*RAD1(IORDER)**2)/(E(IORDER)*THIC(IORDER))
      ELDELF(3)=(PRES(IORDER)*RAD1(IORDER)**2)/(E(IORDER)*THIC(IORDER)
     $*XLENT(IORDER))
      ELDELF(4)=0
C
C
C
C
C     SECONDARY SOLUTIONS
16    CONTINUE
      F1=SINH(BETA)**2.-SIN(BETA)**2.
      F2=SINH(BETA)**2.+SIN(BETA)**2.
      F3=SINH(BETA)*COSH(BETA)+SIN(BETA)*COS(BETA)
      F4=SINH(BETA)*COSH(BETA)-SIN(BETA)*COS(BETA)
      F8=SINH(BETA)*SIN(BETA)
      F9=COSH(BETA)*SIN(BETA)-SINH(BETA)*COS(BETA)
      F10=COSH(BETA)*SIN(BETA)+SINH(BETA)*COS(BETA)
      FACT1=(2.*RAD1(IORDER)**2.*XK)/(E(IORDER)*THIC(IORDER))
      FACT2=(2.*RAD1(IORDER)**2.*XK**2.)/(E(IORDER)*THIC(IORDER))
      FACT3=(2.*RAD1(IORDER)**2.*XK**3.)/(E(IORDER)*THIC(IORDER))
      FLEX(1,1)=FACT3*(2.*F3/F1)
      FLEX(1,2)=FACT2*(F2/F1)
```

```
      FLEX(1,3)=FACT3*(2.*F10/F1)
      FLEX(1,4)=FACT2*(2.*F8/F1)
      FLEX(2,1)=FLEX(1,2)
      FLEX(2,2)=FACT1*(F4/F1)
      FLEX(2,3)=FACT2*(2.*F8/F1)
      FLEX(2,4)=FACT1*(F9/F1)
      FLEX(3,1)=FLEX(1,3)
      FLEX(3,2)=FLEX(2,3)
      FLEX(3,3)=FACT3*(2.*F3/F1)
      FLEX(3,4)=FACT2*(F2/F1)
      FLEX(4,1)=FLEX(1,4)
      FLEX(4,2)=FLEX(2,4)
      FLEX(4,3)=FLEX(3,4)
      FLEX(4,4)=FACT1*(F4/F1)
C
C
C     ENFORCING COMPATIBILITY
C
C
      FLEX(1,1)=-FLEX(1,1)
C      FLEX(1,2)=-FLEX(1,2)
C      FLEX(2,1)=-FLEX(2,1)
      FLEX(2,2)=-FLEX(2,2)
      FLEX(3,3)=-FLEX(3,3)
      FLEX(3,4)=-FLEX(3,4)
      FLEX(4,3)=-FLEX(4,3)
      FLEX(4,4)=-FLEX(4,4)
      ELDELF(1)=ELDELF(1)
      ELDELF(2)=ELDELF(2)
      ELDELF(3)=-ELDELF(3)
      ELDELF(4)=-ELDELF(4)
21    WRITE(12,*) ITYPE(IORDER),RAD1(IORDER),XLENT(IORDER),RAD1(IORDER)
      WRITE(12,*) RAD2(IORDER),XLENT(IORDER),THIC(IORDER),PRES(IORDER)
      WRITE(12,*) XNU(IORDER),E(IORDER),PI,IORDER,NUMELE,COUNT
      WRITE(12,*) XK,BETA,ELDELF(1),ELDELF(2),ELDELF(3),ELDELF(4)
      WRITE(12,*) F1,F2,F3,F4,F8,F9,F10,FACT1,FACT2,FACT3
      DO 10 I=1,4
      WRITE(12,*) (FLEX(I,J),J=1,4)
   10 CONTINUE
      CALL ASSEMB(STFMAT,DEFLEC,IP,IN,PIE)
      RETURN
      END
C
C
C
C
C
C
C
C
C
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
```

```
C                                                                       C
C                                                                       C
C       SUBROUTINE SPHERE CALCULATES THE INFLUENCE                      C
C       FOR THIN OPEN CROWN SPHERICAL SHELLS.                           C
C        THIN SPHERICAL SHELLS HAVE A PRINCIPLE RADIUS                  C
C        /THICKNESS LARGER THAN TEN.                                    C
C                                                                       C
C                                                                       C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C
C
C
C
C
      SUBROUTINE SPHERE(STFMAT,DEFLEC,IP,IN,PIE)
      DIMENSION STFMAT(IDIM,IDIM),DEFLEC(IDIM)
      DIMENSION IP(IDIM),IN(IDIM,2),PIE(IDIM)
      COMMON/ARRAYS/ITYPE(10),RAD1(10),RAD2(10),XLENT(10),LOAD(10),
     S              PRES(10),THIC(50),XNU(10),E(10),ELDELF(4),C(50),
     S              FLEX(4,4)
      COMMON/CONTS/PI,IORDER,NUMELE,COUNT,IDIM
C
C
      DO 11 NI=1,4
      DO 11 NJ=1,4
11    FLEX(NI,NJ)=0.0
C
C
C     CALCULATING SHELL GEOMETRY
C
C
      XL=((RAD1(IORDER)-RAD2(IORDER))**2.+XLENT(IORDER)**2.)**.5
      ALPHA=ATAN((RAD1(IORDER)-RAD2(IORDER))/XLENT(IORDER))
      THETA=PI/2.-ALPHA
      XZ=.5*XL*TAN(ALPHA)
      XM=RAD2(IORDER)/(SIN(THETA))
      T=XM+XZ
      XRAD=((XL/2.)**2.+T**2.)**.5
      IF(XRAD.LE.RAD1(IORDER)) XRAD=RAD1(IORDER)
      PHIONE=ASIN(RAD1(IORDER)/XRAD)
      PHITWO=ASIN(RAD2(IORDER)/XRAD)
      DIFANG=PHIONE-PHITWO
      BETA=XK*DIFANG
      XK=(3.*(1.-XNU(IORDER)**2.)*(XRAD/THIC(IORDER))**2.)**.25
      COEF=1.-((1.+XNU(IORDER))/2.)*(1.-(SIN(PHITWO)**2./SIN(PHIONE)**2
     S))
C
C
C
C     PRIMARY SOLUTIONS
      IF(LOAD(IORDER).EQ.1) GOTO 5
      IF(LOAD(IORDER).EQ.3) GOTO 25
```

```
C
C
C
C     UNIFORM PRESSURE
C
5       CONTINUE
        ELDELF(1)=0.0
        ELDELF(3)=0.0
        ELDELF(2)=((XRAD**2*PRES(IORDER)*SIN(PHIONE))/(E(IORDER)*
     $THIC(IORDER)))*COEF
        ELDELF(4)=((XRAD**2*PRES(IORDER)*SIN(PHITWO))/(E(IORDER)*
     $THIC(IORDER)))
        IF(RAD2(IORDER).EQ.RAD1(IORDER-1)) ELDELF(2)=(XRAD**2*PRES(IORDER)
     $**SIN(PHITWO))/(E(IORDER)*THIC(IORDER))
        IF(RAD2(IORDER).EQ.RAD1(IORDER-1)) ELDELF(4)=((XRAD**2*PRES
     $(IORDER)*SIN(PHIONE))/(E(IORDER)*THIC(IORDER)))*COEF
        GOTO 16
C
C
C
C
C
C       UNIFORM LINE LOAD AROUND CROWN OPENING
C
25      CONTINUE
        ELDELF(1)=0
        ELDELF(2)=((PRES(IORDER)*XRAD)/(E(IORDER)*THIC(IORDER)))
     $*(1.+XNU(IORDER))*(SIN(PHITWO)/SIN(PHIONE))
        ELDELF(3)=0
        ELDELF(4)=((PRES(IORDER)*XRAD)/(E(IORDER)*THIC(IORDER)))
     $*(1.+XNU(IORDER))
        GOTO 16
C
C
C
C
C
C       SECONDARY SOLUTIONS
C
16      CONTINUE
        GOTO 17
C       F1=SINH(BETA)**2.-SIN(BETA)**2.
C       F2=SINH(BETA)**2.+SIN(BETA)**2.
C       F3=SINH(BETA)*COSH(BETA)+SIN(BETA)*COS(BETA)
C       F4=SINH(BETA)*COSH(BETA)-SIN(BETA)*COS(BETA)
C       F8=SINH(BETA)*SIN(BETA)
C       F9=COSH(BETA)*SIN(BETA)-SINH(BETA)*COS(BETA)
C       F10=COSH(BETA)*SIN(BETA)+SINH(BETA)*COS(BETA)
C       FACT1=(2.*XRAD*XK)/(E(IORDER)*THIC(IORDER))
C       FACT2=(2.*XK**2.)/(E(IORDER)*THIC(IORDER))
C       FACT3=(2.*XK**3.)/(E(IORDER)*THIC(IORDER)*XRAD)
CC
C       FLEX(1,1)=FACT3*(2.*F3/F1)
C       FLEX(1,2)=FACT2*SIN(PHIONE)*(F2/F1)
```

```
C       FLEX(1,3)=FACT3*(2.*F10/F1)
C       FLEX(1,4)=FACT2*SIN(PHITWO)*(2.*F8/F1)
C       FLEX(2,1)=FLEX(1,2)
C       FLEX(2,2)=FACT1*SIN(PHIONE)**2.*(F4/F1)
C       FLEX(2,3)=FACT2*SIN(PHIONE)*(2.*F8/F1)
C       FLEX(2,4)=FACT1*SIN(PHIONE)*SIN(PHITWO)*(F9/F1)
C       FLEX(3,1)=FLEX(1,3)
C       FLEX(3,2)=FLEX(2,3)
C       FLEX(3,3)=FACT3*(2.*F3/F1)
C       FLEX(3,4)=FACT2*SIN(PHITWO)*(F2/F1)
C       FLEX(4,1)=FLEX(1,4)
C       FLEX(4,2)=FLEX(2,4)
C       FLEX(4,3)=FLEX(3,4)
C       FLEX(4,4)=FACT1*SIN(PHITWO)**2.*(F4/F1)
C
C
17      CONTINUE
        XK=(E(ICRDER)*THIC(ICRDER)**2)/(12.*(1-XNU(ICRDER)**2))**.5
        ZETA=((E(ICRDER)*THIC(ICRDER)*XRAD/XK)**.5)*PHIONE
        ZETA1=ZETA
        CALL KELVIN(XKER,XKEI,BER,BEI,DXKER,DXKEI,DBER,DBEI,ZETA)
C
C
C       CAP
C
C
        D1=BER*DBEI-BEI*DBER-(((1.-XNU(ICRDER))/ZETA)*(DBEI**2
     S+DBER**2))
        XK111=(1./D1)*(BEI**2+BER**2+((2./ZETA)*(BEI*DBER-BER*
     SDBEI))+(((1.-XNU(ICRDER)**2)/ZETA**2)*(DBEI**2+DBER**2)))
        XK121=(BEI*DBEI+BER*DBER)/D1
        XK221=(DBEI**2+DBER**2)/D1
C
C
C       COMPLEMENT
C
C
        ZETA=((E(ICRDER)*THIC(ICRDER)*XRAD/XK)**.5)*PHITWO
        ZETA2=ZETA
        CALL KELVIN(XKER,XKEI,BER,BEI,DXKER,DXKEI,DBER,DBEI,ZETA)
        D2=XKER*DXKEI-XKEI*DXKER-(((1.-XNU(ICRDER))/ZETA)*(DXKEI**2+
     SDXKER**2))
        XK112=(1./D2)*(XKEI**2+XKER**2+((2./ZETA)*(XKEI*DXKER-XKER*DXKEI))
     S+(((1.-XNU(ICRDER)**2)/ZETA**2)*(DXKEI**2+DXKER**2)))
        XK122=(XKEI*DXKEI+XKER*DXKER)/D2
        XK222=(DXKEI**2+DXKER**2)/D2
C
C
C
        IF(RAD2(ICRDER).EQ.RAD1(ICRDER-1)) GOTO 19
        FLEX(1,1)=((E(ICRDER)*THIC(ICRDER)*XRAD)**.5/(XK**1.5))*XK221
        FLEX(1,2)=(XRAD*SIN(PHIONE)*XK121)/XK
        FLEX(2,1)=FLEX(1,2)
        FLEX(2,2)=(XRAD**1.5*SIN(PHIONE)**2*XK111)/((E(ICRDER)*
```

```
     $THIC(IORDER)*XK)**.5)
      FLEX(3,3)=((E(IORDER)*THIC(IORDER)*XRAD)**.5/(XK**1.5))*XK222
      FLEX(3,4)=(XRAD*SIN(PHITWO)*XK122)/XK
      FLEX(4,3)=FLEX(3,4)
      FLEX(4,4)=(XRAD**1.5*SIN(PHITWO)**2*XK112)/((E(IORDER)*
     $THIC(IORDER)*XK)**.5)
      GOTO 18
19    CONTINUE
      FLEX(1,1)=((E(IORDER)*THIC(IORDER)*XRAD)**.5/(XK**1.5))*XK222
      FLEX(1,2)=(XRAD*SIN(PHITWO)*XK122)/XK
      FLEX(2,1)=FLEX(1,2)
      FLEX(2,2)=(XRAD**1.5*SIN(PHITWO)**2*XK112)/(E(IORDER)*
     $THIC(IORDER)*XK)**.5
      FLEX(3,3)=((E(IORDER)*THIC(IORDER)*XRAD)**.5/(XK**1.5))*XK221
      FLEX(3,4)=(XRAD*SIN(PHIONE)*XK121)/XK
      FLEX(4,3)=FLEX(3,4)
      FLEX(4,4)=(XRAD**1.5*SIN(PHIONE)**2*XK111)/(E(IORDER)*
     $THIC(IORDER)*XK)**.5
15    CONTINUE
C
C
C     ENFORCING COMPATIBILITY
C
C
      FLEX(1,1)=-FLEX(1,1)
C      FLEX(1,2)=-FLEX(1,2)
C      FLEX(2,1)=-FLEX(2,1)
      FLEX(2,2)=-FLEX(2,2)
      FLEX(3,3)=-FLEX(3,3)
      FLEX(3,4)=-FLEX(3,4)
      FLEX(4,3)=-FLEX(4,3)
      FLEX(4,4)=-FLEX(4,4)
      ELDELF(1)=ELDELF(1)
      ELDELF(2)=ELDELF(2)
      ELDELF(3)=-ELDELF(3)
      ELDELF(4)=-ELDELF(4)
C
C
      WRITE(11,*) ITYPE(IORDER),RAD1(IORDER),RAD2(IORDER)
      WRITE(11,*) XLENT(IORDER),THIC(IORDER),PRES(IORDER),XNU(IORDER)
      WRITE(11,*) E(IORDER),PI,NUMELS,XLENT(IORDER),IDIM
      WRITE(11,*) COUNT,IORDER
      WRITE(11,*) XL
      WRITE(11,*) ALPHA,THETA,XZ,XM,T,XRAD,PHIONE,PHITWO,XK,COEF
      WRITE(11,*) ELDELF(1),ELDELF(2),ELDELF(3),ELDELF(4)
      WRITE(11,*) DIFANG,BETA
      WRITE(11,*) F1,F2,F3,F4,F8,F9,F10
      WRITE(11,*) FACT1,FACT2,FACT3
      DO 10 I=1,4
      WRITE(11,*) (FLEX(I,J),J=1,4)
10    CONTINUE
      WRITE(11,*)ZETA1,ZETA2
      WRITE(11,*) D1,D2
      WRITE(11,*) XK111,XK121,XK221
      WRITE(11,*) XK112,XK122,XK222
```

```
      CALL ASSEMB(STFMAT,DEFLEC,IP,IN,PIE)
      RETURN
      END
C
C
C
C
C
C
C
C
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                                                                      C
C          SUBROUTINE ASSEMB ASSEMBLES THE INDIVIDUAL                  C
C          FLEXIBILITY MATRICIES AND DISPLACEMENT VECTORS              C
C          INTO A GLOBAL FLEXIBILITY MATRIX AND                        C
C          DISPLACEMENT VECTOR.                                        C
C          IT IS CALLED BY SUBROUTINES CYLIND, SPHERE,HEMI             C
C          CONE, PLATE, RING, ELLIPS.                                  C
C                                                                      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C
      SUBROUTINE ASSEMB(STFMAT,DEFLEC,IP,IN,PIE)
      DIMENSION STFMAT(IDIM,IDIM),DEFLEC(IDIM)
      DIMENSION IP(IDIM),IN(IDIM,2),PIE(IDIM)
      COMMON/ARRAYS/ITYPE(10),RAD1(10),RAD2(10),XLENT(10),LOAD(10),
     $               PRES(10),THIC(50),XNU(10),E(10),ELDELF(4),C(50),
     $               FLEX(4,4)
      COMMON/CONTS/PI,IORDER,NUMELE,COUNT,IDIM
      NUMJON=IORDER
C
C
      DO 10 I=1,2
        K=2.*(NUMJON-1)+I
        KK=2.*(NUMJON)+I
C
        DO 20 J=1,2
          L=2.*(NUMJON-1)+J
          LL=2.*(NUMJON)+J
          STFMAT(K,LL)=FLEX(I,J+2)+STFMAT(K,LL)
          STFMAT(LL,K)=FLEX(J+2,I)+STFMAT(LL,K)
          STFMAT(K,L)=FLEX(I,J)+STFMAT(K,L)
          STFMAT(KK,LL)=FLEX(I+2,J+2)+STFMAT(KK,LL)
   20   CONTINUE
      DEFLEC(K)=ELDELF(I)+DEFLEC(K)
      DEFLEC(KK)=ELDELF(I+2)+DEFLEC(KK)
   10 CONTINUE
```

```
      COUNT=COUNT+1.
      IF(COUNT.LT.NUMELE) GOTO 50
      DO 30 I=1,IDIM
      WRITE(21,*) COUNT
      WRITE(21,*) (STFMAT(I,J),J=1,IDIM)
 30   CONTINUE
      DO 1 I=1,IDIM
      WRITE(21,*) DEFLEC(I)
1     CONTINUE
C      IDIM=(IDIM-2.)/2.
C
C
      CALL INVERT(STFMAT,DEFLEC,IP,IN,PIE)
50    RETURN
      END
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                              C
C                                                              C
C         SUBROUTINE HEMI CALCULATES THE INFLUENCE             C
C         COEFFICIENTS FOR NONSHALLOW CLOSED HEMI-             C
C         SPHERICAL HEADS.  NONSHALLOW MEANS THAT THE          C
C         DISTURBANCES AT THE EDGES DO NOT INTERACT WITH       C
C         EACH OTHER.                                          C
C                                                              C
C                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
      SUBROUTINE HEMI(STFMAT,DEFLEC,IP,IN,PIE)
      COMMON/ARRAYS/ITYPE(10),RAD1(10),RAD2(10),XLENT(10),LOAD(10),
     $              PRES(10),THIC(50),XNU(10),E(10),ELDELF(4),C(50),
     $              FLEX(4,4)
      COMMON/CONTS/PI,IORDER,NUMELE,COUNT,IDIM
```

```
      DIMENSION STFMAT(IDIM,IDIM),DEFLEC(IDIM)
      DIMENSION IP(IDIM),IN(IDIM,2),PIE(IDIM)
C
C
C     INITIALIZING FLEX
      DO 117 NI=1,4
      DO 117 NJ=1,4
117    FLEX(NI,NJ)=0.0
C
C
C
C     CALCULATING SHELL GEOMETRY
C
C
      XL=(XLENT(IORDER)**2.+RAD1(IORDER)**2.)**.5
      ALPHA=ATAN(XLENT(IORDER)/RAD1(IORDER))
      XM=(XL/2.)/COS(ALPHA)
      XZ=RAD1(IORDER)-XM
      T=COTAN(ALPHA)*XZ
      XRAD=(T**2.+RAD1(IORDER)**2.)**.5
      PHIONE=ASIN(RAD1(IORDER)/XRAD)
      XK=(3.*(1.-XNU(IORDER)**2.)*(XRAD/THIC(IORDER))**2.)**.25
C
C
C
C     PRIMARY SOLUTIONS
C
      IF(LOAD(IORDER).EQ.1) GOTO 5
      IF(LOAD(IORDER).EQ.2) GOTO 15
      IF(LOAD(IORDER).EQ.3) GOTO 25
      IF(LOAD(IORDER).EQ.4) GOTO 35
C
C
C
C     UNIFORM PRESSURE
C
C
5     CONTINUE
      ELDELF(1)=0.0
      ELDELF(2)=((XRAD**2.*PRES(IORDER))/(2.*E(IORDER)*THIC(IORDER)))*(
     $1.-XNU(IORDER))*SIN(PHIONE)
      ELDELF(3)=0.0
      ELDELF(4)=ELDELF(2)
      GOTO 16
C
C
C
C     HYDROSTATIC PRESSURE
C
C
15    CONTINUE
      ELDELF(1)=-((PRES(IORDER)*XRAD**2)/(E(IORDER)*THIC(IORDER)))
```

```
     $*SIN(PHIONE)
      ELDELF(2)=-((PRES(IORDER)*XRAD**3)/(6.*E(IORDER)*THIC(IORDER)))
     $*SIN(PHIONE)*(2.-3.*COS(PHIONE)+2.*(COS(PHIONE)**2/(1.+COS
     $(PHIONE))))*(1.+XNU(IORDER))
      ELDELF(3)=ELDELF(1)
      ELDELF(4)=ELDELF(2)
      GOTO 16
C
C
C
C
C
C
C       WEIGHT LOAD
C
C
C
30    CONTINUE
      ELDELF(1)=-(XRAD*PRES(IORDER)*(2.+XNU(IORDER))*SIN(PHIONE))
     $/(E(IORDER)*THIC(IORDER))
      ELDELF(2)=((XRAD**2*PRES(IORDER)*SIN(PHIONE))/(E(IORDER)*
     $THIC(IORDER)))*(-COS(PHIONE)+((1.+XNU(IORDER))/(1.+COS(PHIONE))))
      ELDELF(3)=ELDELF(1)
      ELDELF(4)=ELDELF(2)
      GOTO 16
C
C
C
C
C       HORIZONTALLY PROJECTED LOAD
C
C
C
25    CONTINUE
      ELDELF(1)=((XRAD*PRES(IORDER))/(E(IORDER)*THIC(IORDER)))
     $*(2.+XNU(IORDER)*SIN(PHIONE))
      ELDELF(2)=((PRES(IORDER)*XRAD**2*SIN(PHIONE))/(E(IORDER)*
     $THIC(IORDER)))*(-COS(PHIONE)**2+((1.+XNU(IORDER))/2.))
      ELDELF(3)=ELDELF(1)
      ELDELF(4)=ELDELF(2)
      GOTO 16
C
C
C
C
C
C       SECONDARY SOLUTIONS
C
C
C
16    CONTINUE
      GOTO 17
      FLEX(1,1)=(4.*XK**3.)/(E(IORDER)*THIC(IORDER)*XRAD)
      FLEX(1,2)=((2.*XK**2.)/(E(IORDER)*THIC(IORDER)))*SIN(PHIONE)
      FLEX(1,3)=0.0
      FLEX(1,4)=0.0
      FLEX(2,1)=FLEX(1,2)
      FLEX(2,2)=((2.*XRAD*XK)/(E(IORDER)*THIC(IORDER)))*SIN(PHIONE)**2.
```

```
      FLEX(2,3)=0.0
      FLEX(2,4)=0.0
      FLEX(3,1)=FLEX(1,3)
      FLEX(3,2)=FLEX(2,3)
      FLEX(3,3)=FLEX(1,1)
      FLEX(3,4)=SIN(PHIONE)*(2.*XK**2.)/(E(IORDER)*THIC(IORDER))
      FLEX(4,1)=0.0
      FLEX(4,2)=FLEX(2,4)
      FLEX(4,3)=FLEX(3,4)
      FLEX(4,4)=FLEX(2,2)
17    CONTINUE
C
C
      XK=(E(IORDER)*THIC(IORDER)**2)/(12*(1-XNU(IORDER)**2))**.5
       ZETA=(((E(IORDER)*THIC(IORDER)*XRAD)/XK)**.5)*PHIONE
      CALL KELVIN(XKER,XKEI,BER,BEI,DXKER,DXKEI,DBER,DBEI,ZETA)
      D=BER*DBEI-BEI*DBER-(((1-XNU(IORDER))/ZETA)*(DBEI**2
     $+DBER**2))
      XK111=(1/D)*(BEI**2+BER**2+((2/ZETA)*(BEI*DBER-BER*DBEI))
     $+(((1-XNU(IORDER)**2)/ZETA**2)*(DBEI**2+DBER**2)))
      XK121=(BEI*DBEI+BER*DBER)/D
      XK221=(DBEI**2+DBER**2)/D
      FLEX(1,1)=(((E(IORDER)*THIC(IORDER)*XRAD)**.5/(XK**1.5))*XK221
      FLEX(1,2)=(XRAD*SIN(PHIONE)*XK121)/XK
      FLEX(2,1)=FLEX(1,2)
      FLEX(2,2)=(XRAD**1.5*SIN(PHIONE)**2*XK111)/((E(IORDER)*THIC
     $(IORDER)*XK)**.5)
      FLEX(3,3)=FLEX(1,1)
      FLEX(3,4)=FLEX(1,2)
      FLEX(4,3)=FLEX(2,1)
      FLEX(4,4)=FLEX(2,2)
C
C
C     ENFORCING COMPATIBILITY
C
C
      FLEX(1,1)=-FLEX(1,1)
C      FLEX(1,2)=-FLEX(1,2)
C      FLEX(2,1)=-FLEX(2,1)
      FLEX(2,2)=-FLEX(2,2)
      FLEX(3,3)=-FLEX(3,3)
      FLEX(3,4)=-FLEX(3,4)
      FLEX(4,3)=-FLEX(4,3)
      FLEX(4,4)=-FLEX(4,4)
      ELDELF(1)=ELDELF(1)
      ELDELF(2)=ELDELF(2)
      ELDELF(3)=-ELDELF(3)
      ELDELF(4)=-ELDELF(4)
C
C
11    CONTINUE
      DO 10 I=1,4
      WRITE(23,*) (FLEX(I,J),J=1,4)
10    CONTINUE
      WRITE(23,*) ELDELF(1),ELDELF(2),ELDELF(3),ELDELF(4),T
```

```
      WRITE(23,*) RAD1(IORDER),PRES(IORDER),THIC(IORDER),XNU(IORDER)
      WRITE(23,*) XL,ALPHA,XM,XZ,T,XRAD
      WRITE(23,*) PHIONE
      B=RAD1(IORDER)**2.
      WRITE(23,*) ZETA
      WRITE(23,*) XK
      WRITE(23,*) XK111,XK121,XK221,D
      WRITE(23,*) XKER,XKEI,BER,BEI
      WRITE(23,*) DXKER,DXKEI,DBER,DBEI
      CALL ASSEMB(STFMAT,DEFLEC,IP,IN,PIE)
      RETURN
      END
C
C
C
C
C
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                           C
C                                                           C
C       SUBROUTINE INVERT INVERTS THE GLOBAL FLEXIBILITY    C
C       MATRIX AND MULTIPLIES IT BY THE GLOBAL DISPLACE-    C
C       MENT VECTOR TO DETERMINE THE DISCONTINUITY FORCES   C
C       IT IS CALLED BY SUBROUTINE ASSEMB.                  C
C                                                           C
C                                                           C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
       SUBROUTINE INVERT (STFMAT,DEFLEC,IP,IN,PIE)
      DIMENSION STFMAT(IDIM,IDIM),IP(IDIM),IN(IDIM,2),PIE(IDIM)
      DIMENSION DEFLEC(IDIM)
      COMMON/ARRAYS/ITYPE(10),RAD1(10),RAD2(10),XLENT(10),LOAD(10),
     $              PRES(10),THIC(50),XNU(10),E(10),ELDELF(4),C(50),
     $              FLEX(4,4)
      COMMON/CONTS/PI,IORDER,NUMELE,COUNT,IDIM
101    FORMAT (20H MATRIX IS SINGULAR )
C
C
C
C     BEGINNING TO INVERT
       DO 51 I=1,IDIM
C
C
C
51      WRITE(24,*) (STFMAT(I,J),J=1,IDIM)
      DO 1 J=1,IDIM
1     IP(J)=0
      DO 14 I=1,IDIM
      AM=0.
      DO 6 J=1,IDIM
      IF (IP(J)-1)2,6,2
```

```
2       DO 5 K=1,IDIM
        IF (IP(K)-1)3,5,19
3      IF (ABS(AM)-ABS(STFMAT(J,K)))4,4,5
4       IR=J
        IC=K
        AM=STFMAT(J,K)
5       CONTINUE
6       CONTINUE
        IP(IC)=IP(IC)+1
        IF (IR-IC)7,9,7
7       DO 8 L=1,IDIM
        SW=STFMAT(IR,L)
        STFMAT(IR,L)=STFMAT(IC,L)
8       STFMAT(IC,L)=SW
9       IN(I,1)=IR
        IN(I,2)=IC
        PIE(I)=STFMAT(IC,IC)
        IF (PIE(I))10,18,10
10      STFMAT(IC,IC)=1.
        DO 11 L=1,IDIM
11      STFMAT(IC,L)=STFMAT(IC,L)/PIE(I)
        DO 14 L1=1,IDIM
        IF (L1-IC) 12,14,12
12      T=STFMAT(L1,IC)
        STFMAT(L1,IC)=0.
        DO 13 L=1,IDIM
13      STFMAT(L1,L)=STFMAT(L1,L)-STFMAT(IC,L)*T
14      CONTINUE
        DO 17 I=1,IDIM
        L=IDIM+1-I
        IF (IN(L,1)-IN(L,2)) 15,17,15
15      JR=IN(L,1)
        JC=IN(L,2)
        DO 16 K=1,IDIM
        SW=STFMAT(K,JR)
        STFMAT(K,JR)=STFMAT(K,JC)
        STFMAT(K,JC)=SW
16      CONTINUE
17      CONTINUE
       DO 20 I=1,IDIM
20     WRITE(24,*) (STFMAT(I,J),J=1,IDIM)
C
C
C     MULTIPLYING BY DISPLACEMENT VECTOR
C
C
C
       DO 50 IROW=1,IDIM
       SUM=0.0
       DO 40 K=1,IDIM
40     SUM=SUM+STFMAT(IROW,K)*DEFLEC(K)
50     C(IROW)=SUM
       WRITE(24,*) (C(I),I=1,IDIM)
       CALL RITEIT
        RETURN
```

```
18      WRITE(24,101)
19      RETURN
        END
C
C
C
C
C
C
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                              C
C                                                              C
C         SUBROUTINE OPNCON CALCULATES THE INFLUENCE COEF-     C
C         FICIENTS FOR  TRUNCATED (LONG AND SHORT) CONICAL     C
C         SECTIONS.                                            C
C                                                              C
C                                                              C
C                                                              C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C
C
C
C
      SUBROUTINE OPNCON(STFMAT,DEFLEC,IP,IN,PIE)
      DIMENSION STFMAT(IDIM,IDIM),DEFLEC(IDIM),IP(IDIM),IN(IDIM)
      DIMENSION PIE(IDIM)
      COMMON/ARRAYS/ITYPE(10),RAD1(10),RAD2(10),XLENT(10),LOAD(10),
     $             PRES(10),THIC(50),XNU(10),E(10),ELDELF(4),C(50),
     $             FLEX(4,4)
      COMMON/CONTS/PI,IORDER,NUMELE,COUNT,IDIM
C
C
C     INITIALIZING FLEX
      DO 11 NI=1,4
      DO 11 NJ=1,4
11    FLEX(NI,NJ)=0.0
C
C
C
C      CALCULATING SHELL GEOMETRY
C
C
C
      ALPHA=ATAN(XLENT(IORDER)/(RAD1(IORDER)-RAD2(IORDER)))
       XL=RAD2(IORDER)*TAN(ALPHA)
      X1=(XL**2.+RAD2(IORDER)**2.)**.5
       X2=((XL+XLENT(IORDER))**2.+RAD1(IORDER)**2.)**.5
      ALPHA2=ASIN(RAD1(IORDER)/X2)
      GAMMA=(PI/2.)-ALPHA
      RADO=RAD1(IORDER)/COS(GAMMA)
```

```
      PHI=PI-ALPHA
C
C      PRIMARY SOLUTIONS
C
C
C
C
C
C
C
C
C
C      UNIFORM PRESSURE
C
C
C
      XM=X2
      IF(RAD2(IORDER).EQ.RAD1(IORDER-1)) XM=X1
      ELDELF(1)=-((PRES(IORDER)*XM*COTAN(ALPHA)**2)/(E(IORDER)*
     $THIC(IORDER)*2.))*(3.+(X1/XM)**2)
      ELDELF(2)=((PRES(IORDER)*XM**2*COS(ALPHA)*COTAN(ALPHA))
     $/(E(IORDER)*THIC(IORDER)))*(1.-((XNU(IORDER/2.)*(1.-(X1/XM)**2))))
      IF(RAD2(IORDER).EQ.RAD1(IORDER-1)) XM=X2
      IF(RAD2(IORDER).NE.RAD1(IORDER-1)) XM=X1
      IF(RAD2(IORDER).EQ.RAD1(IORDER-1)) ELDELF(1)=-ELDELF(1)
      ELDELF(3)=-((PRES(IORDER)*XM*COTAN(ALPHA)**2)/(E(IORDER)*
     $THIC(IORDER)*2.))*(3.+(X1/XM)**2)
      ELDELF(4)=((PRES(IORDER)*XM**2*COS(ALPHA)*COTAN(ALPHA))
     $/(E(IORDER)*THIC(IORDER)))*(1.-((XNU(IORDER/2.)*(1.-(X1/XM)**2))))
      IF(RAD2(IORDER).EQ.RAD1(IORDER-1)) ELDELF(3)=-ELDELF(3)
C
C
C
C
C
C
C      SECONDARY SOLUTIONS
C
C
      XM=X2
      IF(RAD2(IORDER).EQ.RAD1(IORDER-1)) XM=X1
      XK=(3.*(1.-XNU(IORDER)**2))**.25/(THIC(IORDER)*XM*COTAN(ALPHA))
     $**.5
      BETA=XK*XM
      F1=SINH(BETA)**2-SIN(BETA)**2
      F2=SINH(BETA)**2+SIN(BETA)**2
      F3=SINH(BETA)*COSH(BETA)+SIN(BETA)*COS(BETA)
      F4=SINH(BETA)*COSH(BETA)-SIN(BETA)*COS(BETA)
      F8=SINH(BETA)*SIN(BETA)
      F9=COSH(BETA)*SIN(BETA)-SINH(BETA)*COS(BETA)
      F10=COSH(BETA)*SIN(BETA)+SINH(BETA)*COS(BETA)
      D=(E(IORDER)*THIC(IORDER)**3)/(12.*(1.-XNU(IORDER)**2))
      FACT1=1./(2.*D*XK)
      FACT2=SIN(ALPHA)/(2.*D*XK**2)
      FACT3=SIN(ALPHA)**2/(2.*D*XK**3)
      FLEX(1,1)=FACT1*(2.*F3/F1)
```

```
      FLEX(1,2)=FACT2*(F2/F1)
      FLEX(2,1)=FLEX(1,2)
      FLEX(2,2)=FACT3*(F4/F1)
      FLEX(3,1)=(1./(2.*D*XK))*(2.*F10/F1)
      FLEX(3,2)=FACT2*(2.*F8/F1)
      FLEX(4,1)=FACT2*(F9/F1)
      FLEX(4,2)=FACT3*(F9/F1)
      IF(RAD2(IORDER).EQ.0) GOTO 32
      XM=X1
      IF(RAD1(IORDER).EQ.RAD1(IORDER+1)) XM=X2
32    CONTINUE
      XK=(3.*(1.-XNU(ICRDER)**2))**.25/(THIC(IORDER)*XM*COTAN(ALPHA))**
     $.5
      BETA=XK*XM
      WRITE(25,*) XM
      WRITE(25,*) X1
      WRITE(25,*) X2
      WRITE(25,*) BETA
      F1=SINH(BETA)**2-SIN(BETA)**2
      F2=SINH(BETA)**2+SIN(BETA)**2
      F3=SINH(BETA)*COSH(BETA)+SIN(BETA)*COS(BETA)
      F4=SINH(BETA)*COSH(BETA)-SIN(BETA)*COS(BETA)
      F8=SINH(BETA)*SIN(BETA)
      F9=COSH(BETA)*SIN(BETA)-SINH(BETA)*COS(BETA)
      F10=COSH(BETA)*SIN(BETA)+SINH(BETA)*COS(BETA)
      FACT1=1./(2.*D*XK)
      FACT2=SIN(ALPHA)/(2.*D*XK**2)
      FACT3=SIN(ALPHA)**2/(2.*D*XK**3)
      FLEX(1,3)=(1./(2.*D*XK))*(2.*F10/F1)
      FLEX(1,4)=FACT2*(2.*F8/F1)
      FLEX(2,3)=FACT1*(2.*F8/F1)
      FLEX(2,4)=FACT3*(F9/F1)
      FLEX(3,3)=FACT1*(2.*F3/F1)
      FLEX(3,4)=FACT2*(F2/F1)
      FLEX(4,3)=FLEX(3,4)
      FLEX(4,4)=FACT3*F4/F1
      RAD=RAD2(ICRDER)
      IF(RAD1(IORDER).EQ.RAD1(IORDER+1)) RAD=RAD1(ICRDER)
C
C
C     ENFORCING COMPATIBILITY
C
C
C
      FLEX(1,1)=-FLEX(1,1)
C      FLEX(1,2)=-FLEX(1,2)
C      FLEX(2,1)=-FLEX(2,1)
      FLEX(2,2)=-FLEX(2,2)
      FLEX(3,3)=-FLEX(3,3)
      FLEX(3,4)=-FLEX(3,4)
      FLEX(4,4)=-FLEX(4,4)
      ELDELF(1)=ELDELF(1)
      ELDELF(2)=ELDELF(2)
      ELDELF(3)=-ELDELF(3)
      ELDELF(4)=-ELDELF(4)
      DO 10 I=1,4
```

```
      WRITE(25,*) (FLEX(I,J),J=1,4)
 10   CONTINUE
      WRITE(25,*) ITYPE(IORDER),RAD1(IORDER),XLENT(IORDER),RAD1(IORDER)
      WRITE(25,*) RAD2(IORDER),XLENT(IORDER),THIC(IORDER),PRES(IORDER)
      WRITE(25,*) XNU(IORDER),E(IORDER),PI,IORDER,NUMELE,COUNT
      WRITE(25,*) XK,BETA,ELDELF(1),ELDELF(2),ELDELF(3),ELDELF(4)
      WRITE(25,*) F1,F2,F3,F4,F8,F9,F10,FACT1,FACT2,FACT3
      WRITE(25,*) ALPHA,GAMMA,RAD0,XL,X1,X2,XM,D
      WRITE(25,*) ALPHA2,Q,Q2,A
      WRITE(25,*) ALPHA,PHI
      CALL ASSEMB(STFMAT,DEFLEC,IP,IN,PIE)
      RETURN
      END
C
C
C
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                 C
C                                                                 C
C      SUBROUTINE PLATE CALCULATES THE INFLUENCE                  C
C      COEFFICIENTS AND PRIMARY DISPLACEMENTS FOR FLAT            C
C      CIRCULAR PLATES.  PLATES ARE ASSUMED TO BE SIM-            C
C      PLY SUPPORTED WHEN CALCULATING THE PRIMARY                 C
C      DISPLACEMENTS.                                             C
C                                                                 C
C                                                                 C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
      SUBROUTINE PLATE(STFMAT,DEFLEC,IP,IN,PIE)
      DIMENSION STFMAT(IDIM,IDIM),DEFLEC(IDIM),IP(IDIM),IN(IDIM)
      DIMENSION PIE(IDIM)
      COMMON/ARRAYS/ITYPE(10),RAD1(10),RAD2(10),XLENT(10),LOAD(10),
     $              PRES(10),THIC(50),XNU(10),E(10),ELDELF(4),C(50),
     $              FLEX(4,4)
      COMMON/CONTS/PI,IORDER,NUMELE,COUNT,IDIM
C
C
C
C     INITIALIZING FLEX
      DO 11 NI=1,4
      DO 11 NJ=1,4
 11   FLEX(NI,NJ)=0.0
C
C
C   PRIMARY SOLUTIONS
C
C
      D=(E(IORDER)*THIC(IORDER)**3)/(12.*(1.-XNU(IORDER)**2))
      ELDELF(2)=0
      ELDELF(1)=((PRES(IORDER)*RAD1(IORDER))/(16.*D))
```

```
C
C                                                                       C
C                                                                       C
C      S CALCULATES THE INFLUENCE COEFFICIENTS  C
C      AND PRIMARY DISPLACEMENTS FOR RECTANGULAR STIFFENING   C
C      RINGS.                                                           C
C                                                                       C
C                                                                       C
C
C
C
      SUBROUTINE RING(STFMAT,DEFLEC,IP,IN,PIE)
      DIMENSION STFMAT(IDIM,IDIM),DEFLEC(IDIM),IP(IDIM),IN(IDIM,2)
      DIMENSION PIE(IDIM)
      COMMON/ARRAYS/ITYPE(10),RAD1(10),RAD2(10),XLENT(10),LOAD(10),
     $              PRES(10),THIC(50),XNU(10),E(10),ELDELF(4),C(50),
     $              FLEX(4,4)
      COMMON/CONTS/PI,IORDER,NUMELE,COUNT,IDIM
C
C
C     INITIALIZING FLEX
      DO 11 NI=1,4
      DO 11 NJ=1,4
11    FLEX(NI,NJ)=0.0
C
C
      AREA=XLENT(IORDER)*THIC(IORDER)
      XL=(XLENT(IORDER+1)**2.+RAD1(IORDER+1)**2.)**.5
      ALPHA=ATAN(XLENT(IORDER+1)/RAD1(IORDER+1))
      XM=(XL/2.)/COS(ALPHA)
      XZ=RAD1(IORDER+1)-XM
      T=COTAN(ALPHA)*XZ
      XRAD=(T**2.+RAD1(IORDER+1)**2.)**.5
      PHIONE=ASIN(RAD1(IORDER+1)/XRAD)
      EW=(THIC(IORDER)/2.)-(THIC(IORDER-1)/2.)
      EH=(THIC(IORDER)/2.)-((THIC(IORDER+1)/2.)*SIN(PHIONE))
      YC=(XLENT(IORDER)/2.)-((THIC(IORDER+1)/2.)*COS(PHIONE))
      XHYP=YC-EH*TAN(PHIONE)
      EF=XHYP*COS(PHIONE)
      XI=(XLENT(IORDER)**3*THIC(IORDER))/12.
      EH=0
C
C
C     PRIMARY SOLUTIONS
C
      IF(LOAD(IORDER).EQ.1) GOTO 15
      IF(LOAD(IORDER).EQ.2) GOTO 16
C
C
C     UNIFORM PRESSURE
15    CONTINUE
      ELDELF(1)=0.0
```

```
     $*((3.+XNU(IORDER)**2)/(1.+XNU(IORDER)))*RAD1(IORDER)**2-
     &RAD1(IORDER)**2)
      ELDELF(3)=ELDELF(1)
      ELDELF(4)=0
      FLEX(1,1)=RAD1(IORDER)**2/(D*RAD1(IORDER)*(1.+XNU(K)))
      FLEX(1,2)=0
      FLEX(1,3)=0
      FLEX(1,4)=0
      FLEX(2,1)=0
      FLEX(2,2)=(RAD1(IORDER)/(E(IORDER)*THIC(IORDER)))
     $*(1.-XNU(IORDER))
      FLEX(2,3)=0
      FLEX(2,4)=0
      FLEX(3,1)=FLEX(1,3)
      FLEX(3,2)=FLEX(2,3)
      FLEX(3,3)=FLEX(1,1)
      FLEX(3,4)=0
      FLEX(4,1)=FLEX(1,4)
      FLEX(4,2)=FLEX(2,4)
      FLEX(4,3)=FLEX(3,4)
      FLEX(4,4)=FLEX(2,2)
      FLEX(3,3)=FLEX(1,1)
      FLEX(4,4)=FLEX(2,2)
C
C
C
C     ENFORCING COMPATABILITY
C
C
C
      FLEX(1,1)=-FLEX(1,1)
C      FLEX(1,2)=-FLEX(1,2)
C      FLEX(2,1)=-FLEX(2,1)
      FLEX(2,2)=-FLEX(2,2)
      FLEX(3,3)=-FLEX(3,3)
      FLEX(3,4)=-FLEX(3,4)
      FLEX(4,3)=-FLEX(4,3)
      FLEX(4,4)=-FLEX(4,4)
      ELDELF(1)=ELDELF(1)
      ELDELF(2)=ELDELF(2)
      ELDELF(3)=-ELDELF(3)
      ELDELF(4)=-ELDELF(4)
C
C
      WRITE(26,*) (ELDELF(J),J=1,4)
      DO 15 I=1,4
       WRITE(26,*) (FLEX(I,J),J=1,4)
15     CONTINUE
      CALL ASSEMB(STFMAT,DEFLEC,IP,IN,PIE)
      RETURN
      END
C
C
C
```

```
      ELDELF(2)=(PRES(IORDER)*RAD1(IORDER)**3.)/(E(IORDER)*AREA)
      ELDELF(3)=0.0
      ELDELF(4)=(PRES(IORDER)*RAD1(IORDER)**3.)/(E(IORDER)*AREA)
C
C
C     SOME LOADS
C
C
   10 CONTINUE
      XNORM=(XRAD*PRES(IORDER))/(1.+COS(PHIONE))
      XH=XNORM*COS(PHIONE)
      XVERT=XNORM*SIN(PHIONE)
      XMOMET=XH*YO+XVERT*(EM-EH)
      XMOMET=0
      ELDELF(1)=(XMOMET*RAD1(IORDER)**2)/(E(IORDER)*XI)
      ELDELF(2)=(RAD1(IORDER)**2/E(IORDER))*((XH/AREA)-(XMOMET*
     $XLENT(IORDER)/(XI*2)))
      ELDELF(3)=ELDELF(1)
      ELDELF(3)=0
      ELDELF(4)=(RAD1(IORDER)**2/E(IORDER))*((XH/AREA)+(XMOMET*YO/XI))
C
C
C
C     SECONDARY SOLUTIONS
C
C
C
      FLEX(1,1)=(RAD1(IORDER)**2)/(E(IORDER)*XI)
      FLEX(1,2)=-(RAD1(IORDER)**2*(XLENT(IORDER)/2.))/(E(IORDER)*XI)
      FLEX(1,3)=RAD1(IORDER)**2/(E(IORDER)*XI)
      FLEX(1,4)=(RAD1(IORDER)**2*YO)/(E(IORDER)*XI)
      FLEX(2,1)=-(RAD1(IORDER)**2*(XLENT(IORDER)/2.))/(E(IORDER)*XI)
      FLEX(2,2)=(RAD1(IORDER)**2)/(E(IORDER)*AREA)
     $+((XLENT(IORDER)/2)**2*RAD1(IORDER)**2)/(E(IORDER)*XI)
      FLEX(2,3)=-(YO*RAD1(IORDER)**2)/(E(IORDER)*XI)
      FLEX(2,4)=(RAD1(IORDER)**2)/(E(IORDER)*AREA)
     $-((XLENT(IORDER)/2)*YO*RAD1(IORDER)**2)/(E(IORDER)*XI)
      FLEX(3,1)=RAD1(IORDER)**2/(E(IORDER)*XI)
      FLEX(3,2)=-(RAD1(IORDER)**2*YO)/(E(IORDER)*XI)
      FLEX(3,3)=RAD1(IORDER)**2/(E(IORDER)*XI)
      FLEX(3,4)=(RAD1(IORDER)**2*YO)/(E(IORDER)*XI)
      FLEX(4,1)=(RAD1(IORDER)**2*YO)/(E(IORDER)*XI)
      FLEX(4,2)=(RAD1(IORDER)**2)/(E(IORDER)*AREA)
     $-((XLENT(IORDER)/2)*YO*RAD1(IORDER)**2)/(E(IORDER)*XI)
      FLEX(4,3)=(RAD1(IORDER)**2*YO)/(E(IORDER)*XI)
      FLEX(4,4)=((1./AREA)+YO**2/XI)*(RAD1(IORDER)**2/E(IORDER))
C      ELDELF(1)=(RAD1(IORDER)**2*ET*XNORM)/(E(IORDER)*XI)
C      ELDELF(2)=((RAD1(IORDER)**2*COS(PHIONE)/AREA)-(RAD1(IORDER)**2*
C     $(0.)*ET/XI))*(XNORM/E(IORDER))
```

```
C
C
C
C
C     ENFORCING COMPATIBILITY
C
      FLEX(1,1)=-FLEX(1,1)
      FLEX(2,2)=-FLEX(2,2)
      FLEX(3,3)=-FLEX(3,3)
      FLEX(3,4)=-FLEX(3,4)
      FLEX(4,3)=-FLEX(4,3)
      FLEX(4,4)=-FLEX(4,4)
      ELDELF(1)=ELDELF(1)
      ELDELF(2)=ELDELF(2)
      ELDELF(3)=-ELDELF(3)
      ELDELF(4)=-ELDELF(4)
 20   WRITE(28,*) ITYPE(IORDER),RAD1(IORDER),XLENT(IORDER),RAD1(IORDER)
      WRITE(28,*) RAD2(IORDER),XLENT(IORDER),THIC(IORDER),PRES(IORDER)
      WRITE(28,*) XNU(IORDER),E(IORDER),PI,IORDER,NUMELE,COUNT
      WRITE(28,*) KK,BETA,ELDELF(1),ELDELF(2),ELDELF(3),ELDELF(4)
      WRITE(28,*) F1,F2,F3,F4,F8,F9,F10,FACT1,FACT2,FACT3
      DO 10 I=1,4
      WRITE(28,*) (FLEX(I,J),J=1,4)
 10   CONTINUE
      WRITE(28,*) AREA
      WRITE(28,*) XI
      WRITE(28,*) XI
      WRITE(28,*) EW
      WRITE(28,*) EM
      WRITE(28,*) YO
      WRITE(28,*) XHYP
      WRITE(28,*) PHIONE
      WRITE(28,*) ET
      WRITE(28,*) XNORM,XVERT,XH
      WRITE(28,*) XMOMET
      CALL ASSEMB(STFMAT,DEFLEC,IP,IN,PIE)
      RETURN
      END
C
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C                                                                     C
C     SUBROUTINE RITEIT WRITES THE FINAL RESULTS TO                   C
C     FILE RESULT.DAT WHICH MAY BE USED AS HARD COPY BY  THE          C
C     USER AND TO FILE GRAPH.DAT WHICH IS USED BY THE                 C
C     GRAPHIC PROGRAM.   SUBROUTINE RITEIT IS CALLED BY               C
C     SUBROUTINE INVERT.                                              C
C                                                                     C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
```

```
C
C
      SUBROUTINE RITEIT
      COMMON/ARRAYS/ITYPE(10),RAD1(10),RAD2(10),XLENT(10),LOAD(10),
     $              PRES(10),THIC(50),XNU(10),E(10),ELDELF(4),C(50),
     $              FLEX(4,4)
      COMMON/CONTS/PI,IORDER,NUMELE,COUNT,IDIM

      IORDER=0
      IC=-2
      WRITE(30,85)
      WRITE(30,90)
1     IORDER=IORDER+1
      IC=IC+1
      IF(IORDER.GT.NUMELE) GOTO 2
      LL=ITYPE(IORDER)
      GOTO(3000,2000,3000,4000,5000,6000,7000,8000),LL
C
C
C
C
C
3000  CONTINUE
      XL=(RAD1(IORDER)**2+XLENT(IORDER)**2)**.5
      ALPHA=ATAN(RAD1(IORDER)/XLENT(IORDER))
      XZ=.5*XL*TAN(ALPHA)
      XRAD=((XL/2.)**2+XZ**2)**.5
      IF(XRAD.LE.RAD1(IORDER)) XRAD=RAD1(IORDER)
      PHIONE=ASIN(RAD1(IORDER)/XRAD)
      IF(RAD2(IORDER).NE.0) GOTO 100
      WRITE(30,10)
      WRITE(30,20)
      IF(IORDER.EQ.1) XV=-C(4)
      IF(IORDER.EQ.NUMELE) XV=C(IDIM-2)
      XN=PRES(IORDER)*XRAD/2.
      W=XV-XN*COS(PHIONE)
      IF(IORDER.EQ.1) XM=-C(3)
      IF(IORDER.EQ.NUMELE) XM=C(IDIM-3)
      WRITE(30,30)XM,XV
      WRITE(31,*) XM,XV
      GOTO 1
C
C
100   CONTINUE
      WRITE(30,15)
      WRITE(30,45)
      IF(RAD1(IORDER).EQ.RAD1(IORDER-1)) WRITE(30,46)
      IF(RAD2(IORDER).EQ.RAD1(IORDER-1)) WRITE(30,47)
      WRITE(30,50)
      C3=-C(IORDER+3+IC)
      C4=-C(IORDER+4+IC)
      WRITE(30,55) C(IORDER+1+IC),C(IORDER+2+IC),C3,C4
      WRITE(31,*) C(IORDER+1+IC),C(IORDER+2+IC),C3,C4
```

```
      GOTO 1
C
C
C
C
2000  WRITE(30,40)
      WRITE(30,45)
      WRITE(30,50)
      C3=-C(ICRDER+3+IC)
      C4=-C(ICRDER+4+IC)
      WRITE(30,55) C(ICRDER+1+IC),C(ICRDER+2+IC),C3,C4
      WRITE(31,*) C(ICRDER+1+IC),C(ICRDER+2+IC),C3,C4
      GOTO 1
C
C
C
C
C
C
4000   CONTINUE
      WRITE(30,05)
      WRITE(30,45)
      IF(RAD1(ICRDER).EQ.RAD1(ICRDER-1)) WRITE(30,46)
      IF(RAD2(ICRDER).EQ.RAD1(ICRDER-1)) WRITE(30,47)
      WRITE(30,50)
      C3=-C(ICRDER+3+IC)
      C4=-C(ICRDER+4+IC)
      WRITE(30,55) C(ICRDER+1+IC),C(ICRDER+2+IC),C3,C4
      WRITE(31,*) C(ICRDER+1+IC),C(ICRDER+2+IC),C3,C4
      GOTO 1
C
C
C
C
5000  CONTINUE
      WRITE(30,101)
      WRITE(30,20)
      IF(ICRDER.EQ.1) XV=-C(4)
      IF(ICRDER.EQ.1) XM=-C(3)
      IF(ICRDER.EQ.NUMELE) XV=C(IDIM-2)
      IF(ICRDER.EQ.NUMELE) XM=C(IDIM-3)
      WRITE(30,30) XM,XV
      WRITE(31,*) XM,XV
      GOTO 1
C
6000  CONTINUE
      WRITE(30,60)
      WRITE(30,20)
      IF(ICRDER.EQ.1) XV=-C(4)
      IF(ICRDER.EQ.NUMELE) XV=C(IDIM-2)
      IF(ICRDER.EQ.1) XM=-C(3)
      IF(ICRDER.EQ.NUMELE) XM=C(IDIM-3)
      WRITE(30,30) XM,XV
      WRITE(31,*) XM,XVP
```

```
      GOTO 1
C
C
C
C
7000  CONTINUE
      WRITE(30,105)
      WRITE(30,20)
      IF(IORDER.EQ.1) XV=-C(4)
      IF(IORDER.EQ.1) XM=-C(3)
      IF(IORDER.EQ.NUMELE) XV=C(IDIM-2)
      IF(IORDER.EQ.NUMELE) XM=C(IDIM-3)
      WRITE(30,30) XM,XV
      WRITE(31,*) XM,XV
      GOTO 1
C
C
8000  CONTINUE
      WRITE(30,120)
      WRITE(30,45)
      WRITE(30,50)
      C3=-C(IORDER+3+IC)
      C4=-C(IORDER+4+IC)
      WRITE(30,55) C(IORDER+1+IC),C(IORDER+2+IC),C3,C4
      WRITE(31,*) C(IORDER+1+IC),C(IORDER+2+IC),C3,C4
      GOTO 1
C
C
C
10    FORMAT(////,15X,'HEMISPHERICAL HEAD')
15    FORMAT(////,15X,'OPEN CROWN SPHERICAL SECTION')
20    FORMAT(//,10X,'MOMENT',15X,'SHEAR')
30    FORMAT(/,3X,F8.1,12X,F8.1)
40    FORMAT(////,15X,'CYLINDRICAL SHELL')
45    FORMAT(//,10X,'END 1',20X,'END 2')
46    FORMAT(7X,'(LARGE END)',17X,'(SMALL END)')
47    FORMAT(7X,'(SMALL END)',17X,'(LARGE END)')
50    FORMAT(//,3X,'MOMENT',3X,'SHEAR',10X,'MOMENT',3X,'SHEAR')
55    FORMAT(/,3X,2F10.1,3X,2F10.1)
60    FORMAT(////,15X,'CLOSED CONICAL HEAD')
65    FORMAT(////,15X,'OPEN CONICAL SECTION')
66    FORMAT(////,15X,'CLOSED CONICAL HEAD')
85    FORMAT(/,2X,'ALL MOMENTS AND SHEARS ARE IN GLOBAL X-Y COORDINATES'
     &)
90    FORMAT(/,2X,'MOMENTS ARE IN (IN-LB/IN) AND SHEARS ARE IN (LB/IN)')
100   FORMAT(////,15X,'STIFFENING RING')
101   FORMAT(////,'FLAT PLATE')
105   FORMAT(////,15X,'ELLIPTICAL END')
2     CONTINUE
      RETURN
      END
C
C
C
```

```
C
C
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C                                                                     C
C     SUBROUTINE ELLIPS ESTIMATES THE PRIMARY                         C
C     DEFLECTIONS AND THE INFLUENCE COEFFICIENTS                      C
C     FOR FLAT ELLIPSOIDS.                                            C
C                                                                     C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
      SUBROUTINE ELLIPS(STFMAT,DEFLEC,IP,IN,PIE)
      DIMENSION STFMAT(IDIM,IDIM),DEFLEC(IDIM),IP(IDIM),IN(IDIM,2)
      DIMENSION PIE(IDIM)
      COMMON/ARRAYS/ITYPE(10),RAD1(10),RAD2(10),XLENT(10),LOAD(10),
     $              PRES(10),THIC(50),XNU(10),E(10),ELDELF(4),C(50),
     $              FLEX(4,4)
      COMMON/CONTS/PI,IORDER,NUMELE,COUNT,IDIM
C
C
C     INITIALIZING FLEX
      DO 11 NI=1,4
      DO 11 NJ=1,4
11    FLEX(NI,NJ)=0.0
C
C
C
C
C      CALCULATING GEOMETRIC PARAMETERS
C
C
      RA=RAD1(IORDER)**2/XLENT(IORDER)
      PHI=ATAN(RAD1(IORDER)/(RA-XLENT(IORDER)))
      RP1=RAD1(IORDER)**2/(XLENT(IORDER)*COS(PHI)**3*
     $((1.+(RAD1(IORDER)**2/XLENT(IORDER)**2)*TAN(PHI)**2)**1.5))
      RP2=RAD1(IORDER)**2/(XLENT(IORDER)*COS(PHI)*((1.+(RAD1(IORDER)**2
     $/XLENT(IORDER)**2)*TAN(PHI)**2)**.5))
      XK=(E(IORDER)*THIC(IORDER)**2)/(12.*(1.-XNU(IORDER)**2))**.5
C
C
C
C     CALCULATING NUMERICAL PARAMETERS
C
C
      XN=(1.-(XLENT(IORDER)**2/RAD1(IORDER)**2))**.5
      ALPHA=ATAN((RAD1(IORDER)/XLENT(IORDER))*TAN(PHI))
C
```

```
C
C     EVALUATING ELLIPTICAL INTEGRAL OF THE SECOND KIND
C
C
      CALL GAUSS(XN,ALPHA,ELPINT)
C
      ZETA=(((E(IORDER)*THIC(IORDER)*RAD1(IORDER)**2)
     $/(XK*XLENT(IORDER)))**.5)*(.5*(XLENT(IORDER)/RAD1(IORDER))
     $+(ALPHA+ELPINT))
C
C
C     EVALUATING KELVIN FUNCTIONS
C
C
      CALL KELVIN(XKER,XKEI,BER,BEI,DXKER,DXKEI,DBER,DBEI,ZETA)
      D=BER*DBEI-BEI*DBER-(((1.-XNU(IORDER))/ZETA)*(DBEI**2+
     $+DBER**2))
      XK11=(1/D)*(BEI**2+BER**2+(2/ZETA)*(BEI*DBER-BER*DBEI)
     $+((1.-XNU(IORDER)**2)/ZETA**2)*(DBEI**2+DBER**2))
      XK12=(BEI*DBEI+BER*DBER)/D
      XK22=(DBEI**2+DBER**2)/D
C
C
C     PRIMARY SECTIONS
C
C
      ELDELF(1)=-((PRES(IORDER)*RP2)/(2.*E(IORDER)*THIC(IORDER)
     $*TAN(PHI)))*((RP2/RP1)-1)*((RP2/RP1)+3)
      ELDELF(2)=((PRES(IORDER)*RP2**2*SIN(PHI))/(2.*E(IORDER)*
     $THIC(IORDER)))*(2.-XNU(IORDER)-(RP1/RP2))
      ELDELF(3)=-((PRES(IORDER)*RP2)/(2.*E(IORDER)*THIC(IORDER)
     $*TAN(PHI)))*((RP2/RP1)-1)*((RP2/RP1)+3)
      ELDELF(4)=ELDELF(2)
C
C
C     INFLUENCE COEFFICIENTS
C
C
C
      FLEX(2,2)=(RP2**1.5*SIN(PHI)**2*XK11)/(E(IORDER)*THIC(IORDER)*XK)
     $**.5
      FLEX(1,2)=(RP2*SIN(PHI)*XK12)/XK
      FLEX(2,1)=FLEX(1,2)
      FLEX(1,1)=(((E(IORDER)*THIC(IORDER)*RP2)**.5)*XK22)/XK**1.5
      FLEX(3,1)=0
      FLEX(3,2)=0
      FLEX(3,3)=FLEX(1,1)
      FLEX(3,4)=FLEX(1,2)
      FLEX(4,1)=0
      FLEX(4,2)=0
      FLEX(4,3)=FLEX(2,1)
      FLEX(4,4)=FLEX(2,2)
C
```

```
C
C
C
C       ENFORCING COMPATABILITY
C
C
      FLEX(1,1)=-FLEX(1,1)
      FLEX(1,2)=FLEX(1,2)
      FLEX(2,1)=FLEX(2,1)
      FLEX(2,2)=-FLEX(2,2)
      FLEX(3,3)=-FLEX(3,3)
      FLEX(3,4)=-FLEX(3,4)
      FLEX(4,3)=-FLEX(4,3)
      FLEX(4,4)=-FLEX(4,4)
      ELDELF(1)=ELDELF(1)
      ELDELF(2)=ELDELF(2)
      ELDELF(3)=-ELDELF(3)
      ELDELF(4)=-ELDELF(4)
C
C
      WRITE(32,*) ELDELF(1),ELDELF(2),ELDELF(3),ELDELF(4)
      DO 10 I=1,4
      WRITE(32,*) (FLEX(I,J),J=1,4)
10    CONTINUE
      WRITE(32,*) XK,XK11,XK12,XK22,D
      WRITE(32,*) RP1,RP2,D,XK
      AA=XLENT(IORDER)**2/RAD1(IORDER)**2
      WRITE(32,*) PHI,ALPHA,AA
      WRITE(32,*) RA,THIC(IORDER),E(IORDER),XK
      CALL ASSEMB(STFMAT,DEFLEC,IP,IN,PIE)
      RETURN
      END
C
C
C
C
C
C
C
C
C
C
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                    C
C                                                                    C
C       SUBROUTINE GAUSS EVALUATES THE ELLIPTIC INTEGRAL OF THE      C
C       SECOND KIND USING SIX POINT GAUSS QUADRATURE.  IT IS         C
C       CALLED BY SUBROUTINE ELLIPS.                                 C
C                                                                    C
C                                                                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
```

```
      SUBROUTINE GAUSS(XN,ALPHA,ELPINT)
      FUNC(X)=(((1.-XK**2*SIN((X+1)*PI/A)**2)**.5)*(PI/A)
      PI=3.141592654
      XK=SIN(XN)
      B=ALPHA/PI
      A=1.*(1/B)
      C1=.1713244924
      C6=C1
      C2=.3607615730
      C5=C2
      C3=.4679139346
      C4=C3
      GCON1=.9324695142
      GCON2=.6612093865
      GCON3=.2386191861
      ELPINT=C1*FUNC(-GCON1)+C2*FUNC(-GCON2)+C3*FUNC(-GCON3)
     $+C4*FUNC(GCON3)+C5*FUNC(GCON2)+C6*FUNC(GCON1)
      XN=XN*180/PI
      ALPHA=ALPHA*180/PI
      WRITE(33,*) ELPINT
      WRITE(33,*) XN,ALPHA
      XN=XN*PI/180
      ALPHA=ALPHA*PI/180
      RETURN
      END
C
C
C
C
C
C
C
C
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                                                                      C
C     SUBROUTINE KELVIN EVALUATES THE KELVIN FUNCTIONS                 C
C     USED TO DETERMINE THE INFLUENCE COEFFICIENTS FOR                 C
C     ELLIPTICAL AND OPEN AND CLOSED HEMISPHERICAL HEADS.              C
C     IT IS CALLED BY SUBROUTINES ELLIPS, HEMI AND SPHERE.             C
C                                                                      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C
      SUBROUTINE KELVIN(XKER,XKEI,BER,BEI,DXKER,DXKEI,DBER,DBEI,ZETA)
      XKER=0
      XKEI=0
      DXKER=0
      DXKEI=0
      BER=0
```

```
      BEI=0
      DBER=0
      DBEI=0
      FACT1=0
      FACT=0
      XLAMO=0
      XLAME=0
      DXLAMO=0
      DXLAME=0
      C=.577215665
      PI=3.1415927
      IF(ZETA.GT.15) GOTO 30
      IF(ZETA.GT.10) GOTO 30
      DO 15 I=1,8
      K=4*(I-1)*4
      K1=K-2
      KK=K/2-1
      FACT=1
      FACT1=1
      DO 5 II=1,KK
      J=1+II
      FACT1=1*II*FACT1
5      FACT=I*J*FACT
      SBER=((-1)**I)*(((ZETA/2)**K)/(FACT**2))
      SBEI=((ZETA/2)**K1/(FACT1**2))*(-1)**(I+1)
      SDBER=(-1)**I*(((K/2)*(ZETA/2)**(K-1))/(FACT**2))
      SDBEI=(-1)**(I+1)*((((ZETA/2)**(K1-1))*(K1/2))/(FACT1**2))
      DBER=DBER+SDBER
      DBEI=DBEI+SDBEI
      BEI=BEI+SBEI
15     BER=BER+SBER
      BER=BER+1
      FACT=1
      DO 16 I=1,8
      IF(I.EQ.1) LSIGN=1
      IF(I.EQ.2) LSIGN=-1
      IF(I.EQ.4) LSIGN=1
      IF(I.EQ.6) LSIGN=-1
      IF(I.EQ.8) LSIGN=1
      FACT=1.*I*FACT
      XLAM=(((ZETA/2.)**(2.*I))/(FACT**2))
      DXLAM=(I*(ZETA/2)**(2*I-1))/(FACT**2)
      SUM=0
      DO 17 II=1,I
17    SUM=(1./II)+SUM
      IF(((-1)**I).LT.0) XLAMO=(LSIGN)*XLAM*SUM+XLAMO
      IF(((-1)**I).GT.0) XLAME=(LSIGN)*XLAM*SUM+XLAME
      IF(((-1)**I).LT.0) DXLAMO=(LSIGN)*DXLAM*SUM+DXLAMO
      IF(((-1)**I).GT.0) DXLAME=(LSIGN)*DXLAM*SUM+DXLAME
16    CONTINUE
      AKER=-((ALOG(ZETA/2)+C)*BER)+(PI/4)*BEI+XLAME
      AKEI=-((ALOG(ZETA/2)+C)*BEI)-(PI/4)*BER+XLAMO
      DXKER=-((ALOG(ZETA/2)+C)*DBER)-(1/ZETA)*BER+(PI/4)*DBEI+DXLAME
      DXKEI=-((ALOG(ZETA/2)+C)*DBEI)-(1/ZETA)*BEI-(PI/4)*DBER+DXLAMO
      WRITE(34,*) ZETA
```

```
      WRITE(34,*) XKER,XKEI,BER,BEI
      WRITE(34,*) DXKER,DXKEI,DBER,DBEI
      RETURN
20    CONTINUE
      DBETAS=0
      DBETA=0
      DLAMA=0
      DLAMAS=0
      BETSER=0
      BETA=0
      XLAMA=0
      XLAMSE=0
      PHI=ZETA/(2.**.5)-(PI/8.)
      ETA=ZETA/(2.**.5)+(PI/8.)
      DO 22 I=1,3
      K=4+(I-1)*4
      K1=K-2
      KK=K/2-1
      FACT=1
      DO 26 II=1,KK
      J=1+II
26    FACT=1*J*FACT
      IF(I.EQ.1) NN=-1
      IF(I.GT.1) NN=NN+2
      BETA=COS(I*PI/4)/(FACT*(8.*ZETA)**I)
      XLAMA=SIN(I*PI/4)/(FACT*(8.*ZETA)**I)
      DBETA=COS(I*PI/4)/(FACT*I*(8.*ZETA)**(I-1)*8.)
      DLAMA=SIN(I*PI/4)/(FACT*I*(8.*ZETA)**(I-1)*8.)
      BETSER=(NN+2)**2*BETA+BETSER
      XLAMSE=(NN+2)**2*XLAMA+XLAMSE
      DLAMAS=(NN+2)**2*DLAMA+DLAMAS
      DBETAS=(NN+2)**2*DBETA+DBETAS
22    CONTINUE
      BETSER=1+BETSER
      A=((EXP(ZETA/(2.**.5)))*PI)/(2.*(PI*ZETA)**1.5))*(COS(PHI)*ZETA
     $-SIN(PHI)*ZETA-COS(PHI))
      B=((EXP(ZETA/(2.**.5)))*PI)/(2.*(PI*ZETA)**1.5))*(SIN(PHI)*ZETA
     $+COS(PHI)*ZETA-SIN(PHI))
      BER=((EXP(ZETA/(2.**.5)))/((2.*PI*ZETA)**.5))*(COS(PHI)*BETSER
     $+SIN(PHI)*XLAMSE)
      BEI=((EXP(ZETA/(2.**.5)))/((2.*PI*ZETA)**.5))*(SIN(PHI)*BETSER
     $-COS(PHI)*XLAMSE)
      DBER=A*DBETAS+B*DLAMAS
      DBEI=B*DBETAS-A*DLAMAS
      WRITE(34,*) ZETA
      WRITE(34,*) XKER,XKEI,BER,BEI
      WRITE(34,*) DXKER,DXKEI,DBER,DBEI
      RETURN
30     CONTINUE
      BER=(EXP(ZETA/(2.**.5)))/((2.*PI*ZETA)**.5))*
     $COS((ZETA/(2.**.5))-PI/8)
      BEI=(EXP(ZETA/(2.**.5)))/((2.*PI*ZETA)**.5))*
     $SIN((ZETA/(2.**.5))-PI/8)
      XKER=(PI/(2.*ZETA))**.5*EXP(-ZETA/(2.**.5))*
     $COS((ZETA/(2.**.5))+PI/8)
```

```
      XKEI=-(PI/(2.*ZETA))**.5*EXP(-ZETA/(2.**.5))*
     $SIN((ZETA/(2.**.5))+PI/8)
      DBER=(EXP(ZETA/(2.**.5))/((2.*PI*ZETA)**.5))*
     $COS((ZETA/(2.**.5))+PI/8)
      DBEI=(EXP(ZETA/(2.**.5))/((2.*PI*ZETA)**.5))*
     $SIN((ZETA/(2.**.5))+PI/8)
      DXKER=-(PI/(2.*ZETA))**.5*EXP(-ZETA/(2.**.5))*
     $COS((ZETA/(2.**.5))-PI/8)
      DXKEI=(PI/(2.*ZETA))**.5*EXP(-ZETA/(2.**.5))*
     $SIN((ZETA/(2.**.5))-PI/8)
      RETURN
      END
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                    C
C                                                                    C
C       SUBROUTINE CLSCON CALCULATES THE INFLUENCE COEF-             C
C       FICIENTS FOR  TRUNCATED (LONG AND SHORT) CONICAL             C
C       SECTIONS.                                                    C
C                                                                    C
C                                                                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C
C
C
C
      SUBROUTINE CLSCON(STFMAT,DEFLEC,IP,IN,PIE)
      DIMENSION STFMAT(IDIM,IDIM),DEFLEC(IDIM),IP(IDIM),IN(IDIM)
      DIMENSION PIE(IDIM)
      COMMON/ARRAYS/ITYPE(10),RAD1(10),RAD2(10),XLENT(10),LOAD(10),
     $                PRES(10),THIC(50),XNU(10),E(10),ELDELF(4),C(50),
     $                FLEX(4,4)
      COMMON/CONTS/PI,IORDER,NUMELE,COUNT,IDIM
```

```
C
C
C     INITIALIZING FLEX
      DO 11 NI=1,4
      DO 11 NJ=1,4
11    FLEX(NI,NJ)=0.0
C
C
C
C      CALCULATING SHELL GEOMETRY
C
C
C
      ALPHA=ATAN(XLENT(ICRDER)/(RAD1(ICRDER)))
       X2=((XL+XLENT(ICRDER))**2.+RAD1(ICRDER)**2.)**.5
      ALPHA2=ASIN(RAD1(ICRDER)/X2)
      GAMMA=(PI/2.)-ALPHA
      RADO=RAD1(ICRDER)/COS(GAMMA)
      PHI=PI-ALPHA
C
C      PRIMARY SOLUTIONS
C
C
C
C
C
C      UNIFORM PRESSURE
C
C
C
      XM=X2
      ELDELF(1)=((PRES(ICRDER)*XM*COTAN(ALPHA)**2)/(E(ICRDER)*
     $THIC(ICRDER)*2.))*(3.+(XL/XM)**2)
      ELDELF(2)=((PRES(ICRDER)*XM**2*COS(ALPHA)*COTAN(ALPHA))
     $/(E(ICRDER)*THIC(ICRDER)))*(1.-((XNU(ICRDER/2.)*(1.-(XL/XM)**2))))
      ELDELF(3)=((PRES(ICRDER)*XM*COTAN(ALPHA)**2)/(E(ICRDER)*
     $THIC(ICRDER)*2.))*(3.+(XL/XM)**2)
      ELDELF(4)=((PRES(ICRDER)*XM**2*COS(ALPHA)*COTAN(ALPHA))
     $/(E(ICRDER)*THIC(ICRDER)))*(1.-((XNU(ICRDER/2.)*(1.-(XL/XM)**2))))
C
C
C
C
C
C      SECONDARY SOLUTIONS
C
C
      XM=X2
      XK=(3.*(1.-XNU(ICRDER)**2))**.25/(THIC(ICRDER)*XM*COTAN(ALPHA))
     $**.5
      BETA=XK*XM
```

```
      F1=SINH(BETA)**2-SIN(BETA)**2
      F2=SINH(BETA)**2+SIN(BETA)**2
      F3=SINH(BETA)*COSH(BETA)+SIN(BETA)*COS(BETA)
      F4=SINH(BETA)*COSH(BETA)-SIN(BETA)*COS(BETA)
      F8=SINH(BETA)*SIN(BETA)
      F9=COSH(BETA)*SIN(BETA)-SINH(BETA)*COS(BETA)
      F10=COSH(BETA)*SIN(BETA)+SINH(BETA)*COS(BETA)
      D=(E(IORDER)*THIC(IORDER)**3)/(12.*(1.-XNU(IORDER)**2))
      FACT1=1./(2.*D*XK)
      FACT2=SIN(ALPHA)/(2.*D*XK**2)
      FACT3=SIN(ALPHA)**2/(2.*D*XK**3)
      FLEX(1,1)=FACT1*(2.*F3/F1)
      FLEX(1,2)=FACT2*(F2/F1)
      FLEX(2,1)=FLEX(1,2)
      FLEX(2,2)=FACT3*(F4/F1)
      FLEX(3,1)=(1./(2.*D*XK))*(2.*F10/F1)
      FLEX(3,2)=FACT2*(2.*F8/F1)
      FLEX(4,1)=FACT2*(F9/F1)
      FLEX(4,2)=FACT3*(F9/F1)
      FLEX(1,3)=(1./(2.*D*XK))*(2.*F10/F1)
      FLEX(1,4)=FACT2*(2.*F8/F1)
      FLEX(2,3)=FACT1*(2.*F8/F1)
      FLEX(2,4)=FACT3*(F9/F1)
      FLEX(3,3)=FACT1*(2.*F3/F1)
      FLEX(3,4)=FACT2*(F2/F1)
      FLEX(4,3)=FLEX(3,4)
      FLEX(4,4)=FACT3*F4/F1
C
C
C     ENFORCING COMPATIBILITY
C
C
      FLEX(1,1)=-FLEX(1,1)
C      FLEX(1,2)=-FLEX(1,2)
C      FLEX(2,1)=-FLEX(2,1)
      FLEX(2,2)=-FLEX(2,2)
      FLEX(3,3)=-FLEX(3,3)
      FLEX(3,4)=-FLEX(3,4)
      FLEX(4,3)=-FLEX(4,3)
      FLEX(4,4)=-FLEX(4,4)
      ELDELF(1)=ELDELF(1)
      ELDELF(2)=ELDELF(2)
      ELDELF(3)=-ELDELF(3)
      ELDELF(4)=-ELDELF(4)
      DO 10 I=1,4
      WRITE(27,*) (FLEX(I,J),J=1,4)
10    CONTINUE
      WRITE(27,*) ITYPE(ICRDER),RAD1(ICRDER),XLENT(ICRDER),RAD1(ICRDER)
      WRITE(27,*) RAD2(ICRDER),XLENT(ICRDER),THIC(ICRDER),PRES(ICRDER)
      WRITE(27,*) XNU(ICRDER),E(ICRDER),PI,ICRDER,NUMELE,COUNT
      WRITE(27,*) XK,BETA,ELDELF(1),ELDELF(2),ELDELF(3),ELDELF(4)
      WRITE(27,*) F1,F2,F3,F4,F8,F9,F10,FACT1,FACT2,FACT3
      WRITE(27,*) ALPHA,GAMMA,RADO,XL,X1,X2,XM,D
      WRITE(27,*) ALPHA2,A1,A2,A
      WRITE(27,*) ALPHA,PHI
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C     PROGRAM GRAPH                          BY: BRIAN G. SMITH       C
C                                                                     C
C     GRAPH TAKES THE RESULTS FROM ALSHEL AND PLOTS THE               C
C     MERIDIONAL MEMBRANE STRESS RESULTANT, TANGENTIAL                C
C     STRESS RESULTANT, SHEAR STRESS RESULTANT, MERIDIONAL            C
C     MOMENT, TANGENTIAL MOMENT, MERIDIONAL DEFLECTION,               C
C     AND MERIDIONAL ROTATION AS A FUNCTION OF THE                    C
C     MERIDIONAL POSITION OF A GIVEN SHELL.                           C
C                                                                     C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      COMMON/ARRAYS/ITYPE(10),RAD1(10),RAD2(10),XLENT(10),
     $              PRES(10),THIC(10),XNU(10),E(10),LOAD(10)
      COMMON/CONST/XINCRM,L,XK,PI,K,D,NUMELE,RAD,PHICNE,PHITWO
      COMMON/FACT/F1,F2,F3,F4,F5,F6,F7,F8,F9,F10,F11,F12,
     $              F13,F14,F15,F16,F17,F18,F19,F20,F1C,F2C,
     $              F3C,F4C,F5C,F6C,F7C,F8C,F9C,F10C,F11C,
     $              F12C,F13C,F14C,F15C,F16C,F17C,F18C,F19C,F20C,
     $              G2,G3,G6
      DIMENSION NAMES(10),TITLE(20)
      DIMENSION X(1000),Y(1000)
      DATA NAMES/4HTITL,4HNUMB,4HTYPE,4HRAD1,4HRAD2,4HLENT,
     $           4HTHIC,4HLOAD,4HPRES,4HMATR/
      OPEN(UNIT=20,FILE='ALSHEL.DAT',DEVICE='DSK')
      OPEN(UNIT=1,FILE='GRAPH.DAT',DEVICE='DSK')
      OPEN(UNIT=2,FILE='CHECK.DAT',DEVICE='DSK')
      PI=3.14592654
      DO 2000 I=1,2
      READ(20,5000) NAME
      IF(NAME.NE.NAMES(I)) GOTO 2010
      GOTO(2020,2030),I
C
C
C     READING JOB TITLE
C
C
2020  READ(20,5010) TITLE
      WRITE(24,5020) NAME
      WRITE(24,5030) TITLE
C
C
C     READING NUMBER OF SHELLS
C
C
2000  CONTINUE
2030  READ(20,*) NUMELE
      IDIM=2.*NUMELE+2.
      NN=7
```

```
      KK=0
1     KK=KK+1
      IF(KK.GT.NN) GOTO 2140
      READ(20,5000) NAME
      DO 2040 II=3,10
      IF(NAME.EQ.NAMES(II)) GOTO 2050
2040  CONTINUE
2050  M=II-2
      GOTO(2060,2070,2080,2090,2100,2110,2120,2130),M
C
C
C     (1.)  READING THE SHELL TYPES
C
2060  CONTINUE
      READ(20,*),(ITYPE(J),J=1,NUMELE)
      GOTO 1
C
C
C     (2.)  READING THE PRIMARY RADII (RAD1)
C
2070  CONTINUE
      READ(20,*),(RAD1(J),J=1,NUMELE)
      GOTO 1
C
C
C     (3.)  READING ANY SECONDARY RADII (RAD2)
C
2080  CONTINUE
      NN=8
3     DO 2 J=1,NUMELE
      IF(ITYPE(J).NE.1.AND.ITYPE(J).NE.4) GOTO 2
      READ(20,*) RAD2(J)
2     CONTINUE
C     READ(20,*),(RAD2(J),J=1,NUMELE)
      GOTO 1
C
C
C     (4.)  READING AXIAL LENGTH OF THE SHELLS
C
2090  CONTINUE
      READ(20,*),(XLENT(J),J=1,NUMELE)
      GOTO 1
C
C
C     (5.)  READING THE SHELL THICKNESSES
C
2100  CONTINUE
      READ(20,*),(THIC(J),J=1,NUMELE)
      GOTO 1
C
C
C     (6.)  READING THE TYPE OF EXTERNAL LOADING
C
2110  CONTINUE
      READ(20,*),(LOAD(J),J=1,NUMELE)
```

```
      GOTO 1
C
C
C     (7.)  READING THE MAXIMUM PRESSURE
C
2120  CONTINUE
      READ(20,*),(PRES(J),J=1,NUMELE)
      GOTO 1
C
C
C     (8.)  READING MATERIAL PROPERTIES I.E. E AND NU
C
2130  CONTINUE
      READ(20,*),(E(J),J=1,NUMELE)
      READ(20,*),(XNU(J),J=1,NUMELE)
      GOTO 1
2140  CONTINUE
      CALL ARRANG(X,Y)
      GOTO 1000
5000  FORMAT(A4)
5010  FORMAT(20A4)
5020  FORMAT(/1X,A4)
5030  FORMAT(/1X,20A4)
2010  WRITE(5,5040)
5040  FORMAT(//2X,'DATA MODULES ARE OUT OF ORDER')
1000  CONTINUE
      STOP
      END
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C                                                                     C
C     SUBROUTINE FACTOR CALCULATES THE TRIGONEMETRIC                  C
C     FUNCTIONS USED IN CALCULATING DIFFERENT EDGE                    C
C     CONDITIONS AS FUNCTIONS OF THE AXIAL LENGTH                     C
C     FOR THE VARIOUS SHELL GEOMETRIES.                               C
C                                                                     C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
      SUBROUTINE FACTOR(ZETA)
      COMMON/ARRAYS/ITYPE(10),RAD1(10),RAD2(10),XLENT(10),
     $              PRES(10),THIC(10),XNU(10),E(10),LOAD(10)
      COMMON/CONST/XINCRM,L,XK,PI,K,D,NUMELE,RAD,PHICNE,PHITWO
      COMMON/FACT/F1,F2,F3,F4,F5,F6,F7,F8,F9,F10,F11,F12,
     $            F13,F14,F15,F16,F17,F18,F19,F20,F1C,F2C,
     $            F3C,F4C,F5C,F6C,F7C,F8C,F9C,F10C,F11C,
     $            F12C,F13C,F14C,F15C,F16C,F17C,F18C,F19C,F20C,
```

```
$                62,65,66
      XLEN=XLENT(K)
      IF(ITYPE(K).EQ.1) XLEN=1.
      IF(ITYPE(K).EQ.3) XLEN=1.
      XSINH=SINH(XK*XLEN*ZETA)
      XSINE=SIN(XK*XLEN*ZETA)
      XCOSH=COSH(XK*XLEN*ZETA)
      XCOS=COS(XK*XLEN*ZETA)
      F1=XSINH**2-XSINE**2
      F2=XSINH**2+XSINE**2
      F3=XSINH*XCOSH+XSINE*XCOS
      F4=XSINH*XCOSH-XSINE*XCOS
      F5=XSINE**2
      F6=XSINH**2
      F7=XCOSH*XCOS
      F8=XSINH*XSINE
      F9=XCOSH*XSINE-XSINH*XCOS
      F10=XCOSH*XSINE+XSINH*XCOS
      F11=XSINE*XCOS
      F12=XSINH*XCOSH
      F13=XCOSH*XCOS-XSINH*XSINE
      F14=XCOSH*XCOS+XSINH*XSINE
      F15=XCOSH*XSINE
      F16=XSINH*XCOS
      F17=EXP(-XK*XLENT(K)*ZETA*XCOS)
      F18=EXP(-XK*XLENT(K)*ZETA*XSINE)
      F19=EXP(-XK*XLENT(K)*ZETA*(XCOS+XSINE))
      F20=EXP(-XK*XLENT(K)*ZETA*(XCOS-XSINE))
C
C
C     CONSTANT FACTORS
      XLEN=XLENT(K)
      IF(ITYPE(K).EQ.1) XLEN=PHIONE-PHITWO
      IF(ITYPE(K).EQ.3) XLEN=PHIONE
      XSINH=SINH(XK*XLEN)
      XSINE=SIN(XK*XLEN)
      XCOSH=COSH(XK*XLEN)
      XCOS=COS(XK*XLEN)
      F1C=XSINH**2-XSINE**2
      F2C=XSINH**2+XSINE**2
      F3C=XSINH*XCOSH+XSINE*XCOS
      F4C=XSINH*XCOSH-XSINE*XCOS
      F5C=XSINE**2
      F6C=XSINH**2
      F7C=XCOSH*XCOS
      F8C=XSINH*XSINE
      F9C=XCOSH*XSINE-XSINH*XCOS
      F10C=XCOSH*XSINE+XSINH*XCOS
      F11C=XSINE*XCOS
      F12C=XSINH*XCOSH
      F13C=XCOSH*XCOS-XSINH*XSINE
      F14C=XCOSH*XCOS+XSINH*XSINE
      F15C=XCOSH*XSINE
      F16C=XSINH*XCOS
```

```
      F17C=EXP(-XK*XLENT(K)*XCOS)
      F18C=EXP(-XK*XLENT(K)*XSINE)
      F19C=EXP(-XK*XLENT(K)*(XCOS+XSINE))
      F20C=EXP(-XK*XLENT(K)*(XCOS-XSINE))
      G2=.25*(1.-(((RAD1(K)/(RAD1(K)-ZETA))**2)*(1.+2*ALOG(
     5(RAD1(K)-ZETA)/RAD1(K)))))
      G5=.5*(1.-(RAD1(K)/(RAD1(K)-ZETA))**2)
      G8=.5*(1.+XNU(K)+((1.-XNU(K))*(RAD1(K)/(RAD1(K)-ZETA))))
C
C
C
C
C        ROARK CONSTANTS
C
C
C        HEMISPHERICAL HEADS
C
C       IF(ZETA.EQ.0) GOTO 5
C       RK1=1.-((1.-2.*XNU(K))/(2.*XK))*COTAN(PHIONE-ZETA)
C       RK2=1.-((1.+2.*XNU(K))/(2.*XK))*COTAN(PHIONE-ZETA)
C       GOTO 6
C5      RK1=1.
C       RK2=1.
C6      RK3=EXP(-XK*ZETA)/(SIN(PHIONE-ZETA)**.5)
C       RCH=(SIN(PHIONE)**1.5*(1.+RK1**2))/RK1
C       RCM=(2.*XK*SIN(PHIONE)**.5)/(RAD*RK1)
C
C
      RETURN
      END
C
C
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                    C
C                                                                    C
C     SUBROUTINE CYLIND CALCULATES THE RADIAL DEFLECTION,            C
C     ROTATION, TRANSVERSE SHEAR RESULTANT, TANGENTIAL SHEAR         C
C     RESULTANT, AXIAL MOMENT RESULTANT, AND TANGENTIAL              C
C     MOMENT RESULTANT AS A FUNCTION OF AXIAL LENGTH FOR             C
C     BOTH ENDS.                                                     C
C                                                                    C
C                                                                    C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C
      SUBROUTINE CYLIND(X,Y)
      COMMON/ARRAYS/ITYPE(10),RAD1(10),RAD2(10),XLENT(10),
     5              PRES(10),THIC(10),XNU(10),E(10),LOAD(10)
      COMMON/CONST/XINCRM,L,XK,PI,K,J,NUMELE,RAD,PHIONE,PHITWO
      COMMON/FACT/F1,F2,F3,F4,F5,F6,F7,F8,F9,F10,F11,F12,
```

```
      $               F13,F14,F15,F16,F17,F18,F19,F20,F1C,F2C,
      $               F3C,F4C,F5C,F6C,F7C,F8C,F9C,F10C,F11C,
      $               F12C,F13C,F14C,F15C,F16C,F17C,F18C,F19C,F20C,
      $               G2,G5,G8
       DIMENSION X(L),Y(L)
       XK=(3.*(1.-XNU(K)**2))**.25/(RAD1(K)*THIC(K))**.5
       D=(E(K)*THIC(K)**3)/(12.*(1.-XNU(K)**2))
       IEND=0
       READ(1,*) XM1,XV1,XM2,XV2
2      II=0
       IEND=IEND+1
       XM=XM1
       XV=XV1
       IF(IEND.EQ.2) XM=XM2
       IF(IEND.EQ.2) XV=XV2
1      CONTINUE
       II=II+1
       WRITE(5,3)
3      FORMAT(/,3X,'CYLINDRICAL SHELL')
       IF(IEND.EQ.1) WRITE(5,4)
       IF(IEND.EQ.2) WRITE(5,5)
4      FORMAT(/,3X,'FIRST END DATA')
5      FORMAT(/,3X,'SECOND END DATA')
       GOTO(1000,2000,3000,4000,5000,6000),II
C
C
C      CALCULATES DELTA RADIUS AS A FUNCTION OF AXIAL LENGTH
C
C
1000   CONTINUE
       WRITE(5,100)
       READ(5,*) LL
       IF(LL.EQ.2) GOTO 101
       I=0
       ZETA=0
       DO 10 I=2,L
C       WRITE(5,*) ZETA
       X(I)=ZETA*XLENT(K)
       WRITE(2,7)
7       FORMAT(/,3X,'CYLINDER BEFORE FACTOR IS CALLED')
       CALL FACTOR(ZETA)
       WRITE(2,8)
8      FORMAT(/,3X,'SUBROUTINE CYLIND')
       DELRM=-(XM/(2.*D*XK**2))*((F2C/F1C)*F7-(F3C/F1C)*F10+F8)
       IF(IEND.EQ.2) DELRM=XM/(2.*D*XK**2)*(-(2.*F6C/F1C)*F7
     $+(F10C/F1C)*F10)
       DELRM=(XV/(2.*D*XK**3))*((F4C/F1C)*F7-(F5C/F1C)*F15-(F6C/
     $F1C)*F10)
       IF(IEND.EQ.2) DELRM=(XV/(2.*D*XK**3))*(-(F9C/F1C)*F7+(F8C/F1C)
     $*F10)
       Y(I)=DELRM+DELRM
       ZETA=ZETA+(XINCRM/XLENT(K))
10     CONTINUE
       CALL GRAPH(X,Y)
       IF(II.EQ.6) GOTO 15
```

```
101   CONTINUE
100   FORMAT(/,3X,'DEFLECTION AS A FUNCTION OF AXIAL LENGTH (1=YES
     $,2=NO)')
      GOTO 1
C
C
C
C  (2.)  CALCULATES ROTATION AS A FUNCTION OF AXIAL LENGTH
C
C
200   CONTINUE
      WRITE(5,200)
      READ(5,*) LL
      IF(LL.EQ.2) GOTO 201
      I=0
      ZETA=0
      DO 20 I=2,L
      X(I)=ZETA*XLENT(K)
      CALL FACTOR(ZETA)
      BETAM=(XM/(2.*D*XK))*((F2C/F1C)*F9+(2.*F3C/F1C)*F7-F10)
      BETAH=(XV/(2.*D*XK**2))*((F4C/F1C)*F9+(F5C/F1C)*F14
     $+(F6C/F1C)*F13)
      IF(IEND.EQ.2) BETAM=XM/(2.*D*XK)*((2.*F8C/F1C)*F9
     $+(2.*F10C/F1C)*F7)
      IF(IEND.EQ.2) BETAH=XV/(2.*D*XK**2)*((F9C/F1C)*F9
     $+(2.*F6C/F1C)*F7)
      IF(IEND.EQ.1) BETAH=-BETAH
      Y(I)=BETAM+BETAH
      ZETA=ZETA+(XINCRM/XLENT(K))
20    CONTINUE
      CALL GRAPH(X,Y)
      IF(II.EQ.0) GOTO 15
201   CONTINUE
200   FORMAT(/,3X,'ROTATION AS A FUNCTION OF AXIAL LENGTH (1=YES,
     $2=NO)')
      GOTO 1
C
C
C
C  (3.)  CALCULATING TRANSVERSE SHEAR AS A FUNCTION OF
C        AXIAL LENGTH.
C
C
C
3000  CONTINUE
      WRITE(5,300)
      READ(5,*) LL
      IF(LL.EQ.2) GOTO 301
      I=0
      ZETA=0
      DO 30 I=2,L
      X(I)=ZETA*XLENT(K)
      CALL FACTOR(ZETA)
      XQM=(-XK*XM)*((F2C/F1C)*F10-(2.*F3C/F1C)*F8+F9)
      XQH=-XV*((F4C/F1C)*F10+(F5C/F1C)*F13-(F6C/F1C)*F14)
```

```
      IF(IEND.EQ.2) XQH=-XV*((F9C/F1C)*F10-(2.*F8C/F1C)*F8)
      IF(IEND.EQ.2) XQM=XM*XK*((2.*F8C/F1C)*F10-(2.*F10C/F1C
     $)*F8)
      Y(I)=XQM+XQH
      ZETA=ZETA+(XINCRM/XLENT(K))
30    CONTINUE
      CALL GRAPH(X,Y)
      IF(II.EQ.6) GOTO 15
301   CONTINUE
300   FORMAT(/,3X,'SHEAR STRESS RESULTANT AS A FUNCTION OF LENGTH
     $ (1=YES,2=NO)')
      GOTO 1
C
C
C  (4.)  CALCULATING TANGENTIAL SHEAR AS A FUNCTION OF AXIAL LENGTH
C
C
4000  CONTINUE
      WRITE(5,400)
      READ(5,*) LL
      IF(LL.EQ.2) GOTO 401
      I=0
      ZETA=0
      DO 40 I=2,L
      X(I)=ZETA*XLENT(K)
      CALL FACTOR(ZETA)
      XNTHM=XM*2.*XK**2*RAD1(K)*(-(F2C/F1C)*F7+(F3C/F1C)*F10-F8)
      XNTHH=XV*2.*XK*RAD1(K)*((F4C/F1C)*F7-(F5C/F1C)*F15
     $-(F3C/F1C)*F16)
      IF(IEND.EQ.2) XNTHM=XM*2.*XK**2*RAD1(K)*(-(2.*F8C/F1C)*
     $F7+(F10C/F1C)*F10)
      IF(IEND.EQ.2) XNTHH=XV*2.*XK*RAD1(K)*(-(F9C/F1C)*F7
     $+(F8C/F1C)*F10)
      Y(I)=XNTHM+XNTHH
      ZETA=ZETA+(XINCRM/XLENT(K))
40    CONTINUE
      CALL GRAPH(X,Y)
      IF(II.EQ.6) GOTO 15
401   CONTINUE
400   FORMAT(/,3X,'TANGENTIAL STRESS RESULTANT AS A FUNCTION OF AXIAL
     $ LENGTH (1=YES,2=NO)')
      GOTO 1
C
C
C
C  (5.)  CALCULATING AXIAL MOMENT AS A FUNCTION OF AXIAL LENGTH
C
C
5000  CONTINUE
      WRITE(5,500)
      READ(5,*) LL
      IF(LL.EQ.2) GOTO 501
      I=0
      ZETA=0
      DO 50 I=2,L
```

```
      X(I)=ZETA*XLENT(K)
      CALL FACTOR(ZETA)
      XMXM=-XM*((F2C/F1C)*F3-(F3C/F1C)*F9-F7)
      XMXH=-(XV/XK)*(-(F4C/F1C)*F8-(F5C/F1C)*F16+(F6C/F1C)*F15)
      IF(IEND.EQ.2) XMXM=XM*((2.*F6C/F1C)*F8-(F10C/F1C)*F9)
      IF(IEND.EQ.2) XMXH=(XV/XK)*((F9C/F1C)*F8-(F8C/F1C)*F9)
      Y(I)=XMXM+XMXH
      ZETA=ZETA+(XINCRM/XLENT(K))
50    CONTINUE
      CALL GRAPH(X,Y)
      IF(II.EQ.6) GOTO 15
501   CONTINUE
500   FORMAT(/,3X,'AXIAL MOMENT AS AFUNCTION OF AXIAL LENGTH
     $ (1=YES,2=NO)')
      GOTO 1
C
C
C
C  (6.)  CALCULATING TANGENTIAL MOMENT AS A FUNCTION OF AXIAL LENGTH
C
C
6000  CONTINUE
      WRITE(5,600)
      READ(5,*) LL
      IF(LL.EQ.2) GOTO 501
      I=0
      ZETA=0
      DO 60 I=2,L
      X(I)=ZETA*XLENT(K)
      CALL FACTOR(ZETA)
      XMXM=-XM*((F2C/F1C)*F3-(F3C/F1C)*F9-F7)
      XMXH=-(XV/XK)*(-(F4C/F1C)*F8-(F5C/F1C)*F16+(F6C/F1C)*F15)
      IF(IEND.EQ.2) XMXM=XM*((2.*F6C/F1C)*F8-(F10C/F1C)*F9)
      IF(IEND.EQ.2) XMXH=(XV/XK)*((F9C/F1C)*F8-(F8C/F1C)*F9)
      Y(I)=XMXM*XNO(K)+XMXH*XNO(K)
      ZETA=ZETA+(XINCRM/XLENT(K))
60    CONTINUE
      CALL GRAPH(X,Y)
501   CONTINUE
      IF(II.EQ.6) GOTO 15
600   FORMAT(/,3X,'TANGENTIAL MOMENT AS A FUNCTION OF AXIAL LENGTH
     $ (1=YES,2=NO)')
      GOTO 1
C
C     CHECKING FOR END ONE OR TWO
15    IF(IEND.EQ.1) GOTO 2
C
C
      RETURN
      END
C
C
C
C
```

```
C
C
C     SUBROUTINE ARRANG TAKES THE DATA THAT HAS BEEN READ
C     IN FROM ALSHEL.DAT AND PUTS IT INTO THE PROPER ORDER
C     FOR THE APPROPRIATE GEOMETRICAL SUBROUTINES AND GRAPHIC
C     SUBROUTINES.
C
C
C
C
      SUBROUTINE ARRANG(X,Y)
      COMMON/ARRAYS/ITYPE(10),RAD1(10),RAD2(10),XLENT(10),
     $              PRES(10),THIC(10),XNC(10),E(10),LOAD(10)
      COMMON/CONST/XINCRM,L,XK,PI,K,J,NUMELE,RAD,PHICNE,PHITWO
      COMMON/FACT/F1,F2,F3,F4,F5,F6,F7,F8,F9,F10,F11,F12,
     $              F13,F14,F15,F16,F17,F18,F19,F20,F1C,F2C,
     $              F3C,F4C,F5C,F6C,F7C,F8C,F9C,F10C,F11C,
     $              F12C,F13C,F14C,F15C,F16C,F17C,F18C,F19C,F20C,
     $              G2,G5,G8
      DIMENSION X(1000),Y(1000)
C
C
C
C
      K=0
   10 K=K+1
      IF(K.GT.NUMELE) GOTO 10000
      II=ITYPE(K)
      IF(II.EQ.7.OR.II.EQ.8) WRITE(5,1001)
      IF(II.EQ.7) READ(1,*) X1,X2
      IF(II.EQ.8) READ(1,*) X1,X2,X3,X4
      IF(II.EQ.7.OR.II.EQ.8) GOTO 10
      GOTO(1000,2000,3000,4000,5000,6000),II
C
C
C
C
C     (1.)  THIS PORTION ARRANGES DATA FOR OPEN
C           CROWN HEMISPHERICAL SHELLS.
C
C
C
C
1000  CONTINUE
      XL=((RAD1(K)-RAD2(K))**2+XLENT(K)**2)**.5
      ALPHA=ATAN((RAD1(K)-RAD2(K))/XLENT(K))
      THETA=3.1416/2.-ALPHA
      XZ=.5*XL*TAN(ALPHA)
      XM=RAD2(K)/(SIN(THETA))
      T=XM+XZ
      RAL=((XL/2.)**2+T**2)**.5
```

```
      IF(RAD.LE.RAD1(K)) RAD=RAD1(K)
      PHIONE=ASIN(RAD1(K)/RAD)
      PHITWO=ASIN(RAD2(K)/RAD)
      PHI=(PHIONE-PHITWO)*180./3.1416
      LPHI=PHI
      XINCRM=PHI-LPHI
      IF(XINCRM.EQ.0) L=(LPHI/XINCRM)+2
      IF(XINCRM.GT.0) L=(LPHI/XINCRM)+3
      XINCRM=PHI-LPHI
C      WRITE(5,*) RAD1(K),RAD2(K),XLENT(K)
C      WRITE(5,*) XL,ALPHA,THETA
C      WRITE(5,*) XZ,XM,T,RAD
C      WRITE(5,*) PHIONE,PHITWO,PHI,LPHI,XINCRM,L
      CALL SPHERE(X,Y,PHI)
      IF(K.EQ.NUMELE) GOTO 10000
      GOTO 10
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                           C
C    (2.)  THIS PORTION ARRANGES DATA FOR                   C
C          CIRCULAR CYLINDERS.                              C
C                                                           C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
2000  CONTINUE
      LENT=XLENT(K)
      XINCRM=XLENT(K)-LENT
      IF(XINCRM.EQ.0) XINCRM=1
      IF(XINCRM.EQ.0) L=(LENT/XINCRM)+2
      IF(XINCRM.GT.0) L=(LENT/XINCRM)+3
      CALL CYLINC (X,Y)
      IF(K.EQ.NUMELE) GOTO 10000
      GOTO 10
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                           C
C    (3.)  THIS PORTION ARRANGES THE DATA FOR CLOSED HEMI-  C
C          SPHERICAL HEADS.                                 C
C                                                           C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
3000  CONTINUE
      XL=(XLENT(K)**2+RAD1(K)**2)**.5
      ALPHA=ATAN(XLENT(K)/RAD1(K))
      XM=(XL/2.)/COS(ALPHA)
      XZ=RAD1(K)-XM
      T=COTAN(ALPHA)*XZ
      RAD=(T**2+RAD1(K)**2)**.5
      PHIONE=ASIN(RAD1(K)/RAD)
```

```
      IF(RAD.EQ.RAD1(K)) PHIONE=PI/4.
      XINCRM=.01
      L=PHIONE/XINCRM+3
      WRITE(2,*) PHIONE,RAD,L,XINCRM,LPHI
      WRITE(5,*) PHIONE,RAD,L,XINCRM,LPHI
      CALL HEMI(X,Y)
      IF(K.EQ.NUMELE) GOTO 10000
      GOTO 10
C
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                  C
C  (4.)  THIS PORTION ARRANGES DATA FOR OPEN                       C
C          OR CLOSED CONES.                                        C
C                                                                  C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
4000  CONTINUE
      ALPHA=ATAN(XLENT(K)/(RAD1(K)-RAD2(K)))
      XL=RAD2(K)*TAN(ALPHA)
      X1=(XL**2+RAD2(K)**2)**.5
      X2=((XL+XLENT(K))**2+RAD1(K)**2)**.5
      ALPHA2=ASIN(RAD1(K)/X2)
      GAMMA=(PI/2.)-ALPHA
      PHI=(PI/2.-GAMMA)
      RADO=RAD1(K)/COS(GAMMA)
      LBAR=X2-X1
      XINCRM=(X2-X1)-LBAR
      IF(XINCRM.EQ.0) XINCRM=1
      IF(XINCRM.EQ.0) L=(LBAR/XINCRM)+2
      IF(XINCRM.GT.0) L=(LBAR/XINCRM)+3
      WRITE(5,*) L,LBAR,X1,X2,XINCRM
       CALL OPNCON(X,Y,X1,X2,ALPHA,ALPHA2,PHI,XL)
C      IF(RAD2(K).EQ.0) CALL CLSCON(X,Y,X2,ALPHA,ALPHA2,PHI,XL,RADO)
      GOTO 10
C
C
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                  C
C       (5.)  THIS PORTION ARRANGES DATA FOR FLAT                  C
C             CIRCULAR PLATES.                                     C
C                                                                  C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
5000  CONTINUE
      RAD=RAD1(K)
```

```
      XINCRM=RAD1(K)-LRAD
      IF(XINCRM.EQ.0) XINCRM=1
      IF(XINCRM.EQ.0) L=(LRAD/XINCRM)+2
      IF(XINCRM.GT.0) L=(LRAD/XINCRM)+3
      CALL PLATE(X,Y)
      IF(K.EQ.NUMELE) GOTO 10000
      GOTO 10
C
C
C
C
C
C
C
C
10000 CONTINUE
9001  FORMAT(//,2X,'NO GRAPHICAL DISPLAY IS AVAILABLE FOR THIS ELEMENT')
      RETURN
      END
C
C
C
C
C
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C                                                                      C
C     SUBROUTINE SPHERE CALCULATES THE RADIAL DEFLECTION,              C
C     MERIDIONAL ROTATION, TANGENTIAL MOMENT, MERIDIONAL               C
C     MOMENT, SHEAR STRESS RESULTANT, MERIDIONAL STRESS                C
C     RESULTANT, AND TANGENTIAL STRESS RESULTANT AS A                  C
C     FUNCTION OF THE MERIDIONAL LENGTH FOR OPEN CROWN                 C
C     SPHERICAL SHELLS.  IT IS CALLED BY SUBROUTINE ARRANG.            C
C                                                                      C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE SPHERE(X,Y,PHI)
      COMMON/ARRAYS/ITYPE(10),RAD1(10),RAD2(10),XLENT(10),
     $              PRES(10),THIC(10),XNU(10),E(10),LOAD(10)
      COMMON/CONST/XINCRM,L,XK,PI,K,D,NUMELE,RAD,PHICNE,PHITWO
      COMMON/FACT/F1,F2,F3,F4,F5,F6,F7,F8,F9,F10,F11,F12,
     $              F13,F14,F15,F16,F17,F18,F19,F20,F1C,F2C,
     $              F3C,F4C,F5C,F6C,F7C,F8C,F9C,F10C,F11C,
     $              F12C,F13C,F14C,F15C,F16C,F17C,F18C,F19C,F20C,
     $              G2,G5,G8
      DIMENSION X(L),Y(L)
C
C
      X(1)=0
      Y(1)=0
      READ(1,*) XM1,XH1,XM2,XH2
```

```
      AK=(3.*(1.-XNU(K)**2)*(RAD/THIC(K))**2)**.25
      IEND=0
      IF(RAD2(K).EQ.RAD1(K-1)) GOTO 11
      XH11=XH1
      XM11=XM1
      XH22=XH2
      XM22=XM2
      GOTO 2
11    XH22=XH1
      XM22=XM1
      XH11=XH2
      XM11=XM2
2     II=0
      IEND=IEND+1
 1    CONTINUE
      II=II+1
      IF(II.GT.7) GOTO 15
      WRITE(5,3)
3     FORMAT(/,3X,'OPEN CROWN SPHERICAL SECTION')
      IF(IEND.EQ.1.AND.RAD2(K).EQ.RAD1(K-1)) WRITE(5,4)
      IF(IEND.EQ.2.AND.RAD2(K).EQ.RAD1(K+1)) WRITE(5,4)
      IF(IEND.EQ.1.AND.RAD1(K).EQ.RAD1(K-1)) WRITE(5,5)
      IF(IEND.EQ.2.AND.RAD1(K).EQ.RAD1(K+1)) WRITE(5,5)
4     FORMAT(/,3X,'SMALL END')
5     FORMAT(/,3X,'LARGE END')
      GOTO(1000,2000,3000,4000,5000,6000,7000),II
C
C
C
C     (1.)  CALCULATING THE RADIAL DEFLECTION AS A FUNCTION OF THE
C           MERIDIONAL LENGTH (PHI).
C
C
C
1000  CONTINUE
      WRITE(5,100)
      READ(5,*) LL
      IF(LL.EQ.2) GOTO 101
      I=0
      ZETA=0
      DO 10 I=2,L
      X(I)=ZETA*180./PI
      CALL FACTOR(ZETA)
      DELR1H=-((XH11*RAD*XK)/(E(K)*THIC(K)))*SIN(PHIONE-ZETA)*SIN
     $(PHIONE)*(-F9-(2.*F8C/F1C)*F7+(F2C/F1C)*F10)
      DELR1M=-((XM11*2.*XK**2*SIN(PHIONE-ZETA))/(E(K)*THIC(K)))*
     $((F8C/F1C)*F14+(F5C/F1C)*F13-(F3C/F1C)*F10)
      DELR2H=-((XH22*SIN(PHITWO)*2.*RAD*XK*SIN(PHIONE-ZETA))
     $/(E(K)*THIC(K)))*((F9C/F1C)*F7-(F8C/F1C)*F10)
      DELR2M=((XM22*2.*XK**2*SIN(PHIONE-ZETA))/(E(K)*THIC(K)))
     $*(-2.*(F8C/F1C)*F7+(F10C/F1C)*F10)
      IF(RAD2(K).NE.0.AND.RAD1(K).EQ.RAD1(K-1)) DELR1M=-DELR1M
      IF(RAD2(K).NE.0.AND.RAD1(K).EQ.RAD1(K-1)) DELR2M=-DELR2M
      IF(IEND.EQ.1.AND.RAD2(K).EQ.RAD1(K-1)) Y(I)=DELR2H+DELR2M
```

```
      IF(IEND.EQ.1.AND.RAD1(K).EQ.RAD1(K-1)) Y(I)=DELR1H+DELR1M
      IF(IEND.EQ.2.AND.RAD2(K).EQ.RAD1(K+1)) Y(I)=DELR2H+DELR2M
      IF(IEND.EQ.2.AND.RAD1(K).EQ.RAD1(K+1)) Y(I)=DELR1H+DELR1M
      ZETA=ZETA+XINCRM*PI/180.
10    CONTINUE
      CALL GRAPH(X,Y)
      IF(II.EQ.7) GOTO 15
101   CONTINUE
100   FORMAT(//,3X,'DELTA AS A FUNCTION OF ANGLE (1=YES,2=NO)')
      GOTO 1
C
C
C
C
C   (2.)  CALCULATING MERIDIONAL ROTATION AS A FUNCTION OF THE
C         MERIDIONAL LENGTH (PHI).
C
C
2000  CONTINUE
      WRITE(5,200)
      READ(5,*) LL
      IF(LL.EQ.2) GOTO 201
      I=0
      ZETA=0
      DO 20 I=2,L
      X(I)=ZETA*180./PI
      CALL FACTOR(ZETA)
      BETA1H=-((XH11*2.*XK**2*SIN(PHIONE))/(E(K)*THIC(K)))
     $*(-F8+(F4C/F1C)*F9+(F2C/F1C)*F7)
      BETA1M=-((XM11*4.*XK**3)/(E(K)*THIC(K)*RAD))*((F6C/F1C)*F10-
     $(F5C/F1C)*F15-(F3C/F1C)*F7)
      BETA2M=((XM22*4.*XK**3)/(E(K)*THIC(K)*RAD))*((F8C/F1C)*F9
     $+(F10C/F1C)*F7)
      BETA2H=((XH22*SIN(PHITWO)*2.*XK**2)/(E(K)*THIC(K)))
     $*((F9C/F1C)*F9+(2.*F8C/F1C)*F7)
      IF(RAD2(K).NE.0.AND.RAD1(K).EQ.RAD1(K-1)) BETA1H=-BETA1H
      IF(RAD2(K).NE.0.AND.RAD1(K).EQ.RAD1(K-1)) BETA2H=-BETA2H
      IF(IEND.EQ.1.AND.RAD2(K).EQ.RAD1(K-1)) Y(I)=BETA2H+BETA2M
      IF(IEND.EQ.1.AND.RAD1(K).EQ.RAD1(K-1)) Y(I)=BETA1H+BETA1M
      IF(IEND.EQ.2.AND.RAD2(K).EQ.RAD1(K+1)) Y(I)=BETA2H+BETA2M
      IF(IEND.EQ.2.AND.RAD1(K).EQ.RAD1(K+1)) Y(I)=BETA1H+BETA1M
      ZETA=ZETA+XINCRM*PI/180.
20    CONTINUE

      CALL GRAPH(X,Y)
      IF(II.EQ.7) GOTO 15
200   FORMAT(//,3X,'ROTATION AS A FUNCTION OF ANGLE (1=YES,2=NO)')
201   CONTINUE
      GOTO 1
C
C
C
C
C   (3.)  CALCULATING MERIDIONAL RESULTANT MOMENT AS A FUNCTION OF
C         MERIDIONAL LENGTH (PHI).
```

```
C
C
3000  CONTINUE
      WRITE(5,300)
      READ(5,*) LL
      IF(LL.EQ.2) GOTO 301
      I=0
      ZETA=0
      DO 30 I=2,L
      X(I)=ZETA*180./PI
      CALL FACTOR(ZETA)
      XMPH1M=((XH11*RAD*SIN(PHIONE))/(2.*XK))*(-F10+(2.*F4C/F1C)*F8
     S-(F2C/F1C)*F9)
      XMPH1M=XM11*((F6C/F1C)*F13-(F5C/F1C)*F14+(F3C/F1C)*F9)
      XMPH2M=XM22*((2.*F8C/F1C)*F8-(F10C/F1C)*F9)
      XMPH2H=(XH22*RAD*SIN(PHITWO)/XK)*((F9C/F1C)*F8-(F8C/F1C)*F9)
      IF(RAD2(K).NE.0.AND.RAD1(K).EQ.RAD1(K-1)) XMPH1H=-XMPH1H
      IF(RAD2(K).NE.0.AND.RAD1(K).EQ.RAD1(K-1)) XMPH2H=-XMPH2H
      IF(IEND.EQ.1.AND.RAD2(K).EQ.RAD1(K-1)) Y(I)=XMPH2H+XMPH2M
      IF(IEND.EQ.1.AND.RAD1(K).EQ.RAD1(K-1)) Y(I)=XMPH1H+XMPH1M
      IF(IEND.EQ.2.AND.RAD2(K).EQ.RAD1(K+1)) Y(I)=XMPH2H+XMPH2M
      IF(IEND.EQ.2.AND.RAD1(K).EQ.RAD1(K+1)) Y(I)=XMPH1H+XMPH1M
      ZETA=ZETA+XINCRM*PI/180.
30    CONTINUE
      CALL GRAPH(X,Y)
      IF(II.EQ.7) GOTO 15
301   CONTINUE
300   FORMAT(//,3X,'MERIDIONAL MOMENT AS A FUNCTION
     S OF ANGLE (1=YES,2=NO)')
      GOTO 1
C
C
C
C
C  (4.)  CALCULATING TANGENTIAL MOMENT RESULTANT AS A FUNCTION
C        OF THE MERIDIONAL LENGTH (PHI).
C
C
4000  CONTINUE
      WRITE(5,400)
      READ(5,*) LL
      IF(LL.EQ.2) GOTO 401
      I=0
      ZETA=0
      DO 40 I=2,L
      X(I)=ZETA*180./PI
      CALL FACTOR(ZETA)
      XMPH1H=((XH11*RAD*SIN(PHIONE))/(2.*XK))*(-F10+(2.*F4C/F1C)*F8
     S-(F2C/F1C)*F9)
      XMPH1M=XM11*((F6C/F1C)*F13-(F5C/F1C)*F14+(F3C/F1C)*F9)
      XMPH2M=XM22*((2.*F8C/F1C)*F8-(F10C/F1C)*F9)
      XMPH2H=(XH22*RAD*SIN(PHITWO)/XK)*((F9C/F1C)*F8-(F8C/F1C)*F9)
      IF(IEND.EQ.1.AND.RAD2(K).EQ.RAD1(K-1)) Y(I)=XNU(K)*(XMPH2H+XMPH2M)
      IF(IEND.EQ.1.AND.RAD1(K).EQ.RAD1(K-1)) Y(I)=XNU(K)*(XMPH1H+XMPH1M)
      IF(IEND.EQ.2.AND.RAD2(K).EQ.RAD1(K+1)) Y(I)=XNU(K)*(XMPH2H+XMPH2M)
```

```
      IF(IEND.EQ.2.AND.RAD1(K).EQ.RAD1(K+1)) Y(I)=XNU(K)*(XMPH1H+XMPH1M)
      ZETA=ZETA+XINCRM*PI/180.
40    CONTINUE
      CALL GRAPH(X,Y)
      IF(II.EQ.7) GOTO 15
401   CONTINUE
400   FORMAT(//,3X,'TANGENTIAL MOMENT AS A FUNCTION
     $ OF ANGLE (1=YES,2=NO)')
      GOTO 1
C
C
C
C
C     (b.)  CALCULATING SHEAR STRESS RESULTANT AS A FUNCTION OF THE
C           MERIDIONAL LENGTH (PHI).
C
C
500   CONTINUE
      WRITE(5,500)
      READ(5,*) LL
      IF(LL.EQ.2) GOTO 501
      I=0
      ZETA=0
      DO 50 I=2,L
      X(I)=ZETA*180./PI
      CALL FACTOR(ZETA)
      XQPH1M=XH11*SIN(PH1ONE)*(F7-(F4C/F1C)*F10+(F2C/F1C)*F8)
      XQPH1M=-(XM11*2.*XK/RAD)*((F6C/F1C)*F15+(F5C/F1C)*F10-(F3C/F1C)
     $*F8)
      XQPH2M=-XH22*SIN(PH1TWO)*((F9C/F1C)*F10-(2.*F8C/F1C)*F8)
      XQPH2M=-(XM22*2.*XK/RAD)*(-(F8C/F1C)*F10+(F10C/F1C)*F8)
      IF(RAD2(K).NE.0.AND.RAD1(K).EQ.RAD1(K-1)) XQPH1M=-XQPH1M
      IF(RAD2(K).NE.0.AND.RAD1(K).EQ.RAD1(K-1)) XQPH2M=-XQPH2M
      IF(IEND.EQ.1.AND.RAD2(K).EQ.RAD1(K-1)) Y(I)=XQPH2H+XQPH2M
      IF(IEND.EQ.1.AND.RAD1(K).EQ.RAD1(K-1)) Y(I)=XQPH1H+XQPH1M
      IF(IEND.EQ.2.AND.RAD2(K).EQ.RAD1(K+1)) Y(I)=XQPH2H+XQPH2M
      IF(IEND.EQ.2.AND.RAD1(K).EQ.RAD1(K+1)) Y(I)=XQPH1H+XQPH1M
      ZETA=ZETA+XINCRM=PI/180.
50    CONTINUE
      CALL GRAPH(X,Y)
      IF(II.EQ.7) GOTO 15
501   CONTINUE
500   FORMAT(//,3X,'SHEAR STRESS RESULTANT AS A FUNCTION
     $ OF ANGLE (1=YES,2=NO)')
      GOTO 1
C
C
C
C
C     (c.)  CALCULATING MERIDIONAL STRESS RESULTANT AS A FUNCTION
C           OF THE MERIDIONAL LENGTH (PHI).
C
C
600   CONTINUE
      WRITE(5,600)
```

```
      READ(5,*) LL
      IF(LL.EQ.2) GOTO 601
      I=0
      ZETA=0
      DO 60 I=2,L
      X(I)=ZETA*180./PI
      CALL FACTOR(ZETA)
      IF(ZETA.EQ.0.OR.I.EQ.92) GOTO 65
      XNPH1H=-XH11*SIN(PHIONE)*COTAN(PHIONE-ZETA)*(F7-(F4C/F1C)*F10
     $+(F2C/F1C)*F8)
      XNPH1M=(XM11*COTAN(PHIONE-ZETA)*2.*XK/RAD)*((F6C/F1C)*F15+
     $(F5C/F1C)*F10-(F3C/F1C)*F8)
      XNPH2M=-(XM22*2.*XK*COTAN(PHIONE-ZETA)/RAD)*((F8C/F1C)*F10-
     $(F10C/F1C)*F6)
      XNPH2H=(XH22*COTAN(PHIONE-ZETA)*SIN(PHITHO))*((F9C/F1C)*F10-
     $(2.*F8C/F1C)*F6)
      IF(RAD2(K).NE.0.AND.RAD1(K).EQ.RAD1(K-1)) XNPH1M=-XNPH1M
      IF(RAD2(K).NE.0.AND.RAD1(K).EQ.RAD1(K-1)) XNPH2M=-XNPH2M
65      IF(IEND.EQ.1.AND.RAD2(K).EQ.RAD1(K-1)) Y(I)=XNPH2H+XNPH2M
      IF(IEND.EQ.1.AND.RAD1(K).EQ.RAD1(K-1)) Y(I)=XNPH1H+XNPH1M
      IF(IEND.EQ.2.AND.RAD2(K).EQ.RAD1(K+1)) Y(I)=XNPH2H+XNPH2M
      IF(IEND.EQ.2.AND.RAD1(K).EQ.RAD1(K+1)) Y(I)=XNPH1H+XNPH1M
       ZETA=ZETA+XINCR*PI/180.
60    CONTINUE
      CALL GRAPH(X,Y)
      IF(II.EQ.7) GOTO 15
601   CONTINUE
600   FORMAT(//,3X,'MERIDIONAL STRESS RESULTANT AS A FUNCTION
     $ OF ANGLE (1=YES,2=NO)')
      GOTO 1
C
C
C
C  (7.)  CALCULATING THE TANGENTIAL STRESS RESULTANT AS A FUNCTION
C        OF THE MERIDIONAL LENGTH (PHI).
C
C
700   CONTINUE
      WRITE(5,700)
      READ(5,*) LL
      IF(LL.EQ.2) GOTO 701
      I=0
      ZETA=0
      DO 70 I=2,L
      X(I)=ZETA*180./PI
      CALL FACTOR(ZETA)
      XNTH1H=-(XH11*XK*SIN(PHIONE))*(-F9-(2.*F4C/F1C)*F7+(F2C/F1C)*
     $F10)
      XNTH1M=-(XM11*2.*XK**2/RAD)*((F6C/F1C)*F14+(F5C/F1C)*F13-
     $(F3C/F1C)*F10)
      XNTH2M=(XM22*2.*XK**2/RAD)*(-(2.*F8C/F1C)*F7+(F10C/F1C)*F10)
      XNTH2H=(XH22*2.*XK*SIN(PHITHO))*(-(F9C/F1C)*F7+(F6C/F1C)*F10)
      IF(RAD2(K).NE.0.AND.RAD1(K).EQ.RAD1(K-1)) XNTH1M=-XNTH1M
      IF(RAD2(K).NE.0.AND.RAD1(K).EQ.RAD1(K-1)) XNTH2M=-XNTH2M
      IF(IEND.EQ.1.AND.RAD2(K).EQ.RAD1(K-1)) Y(I)=XNTH2H+XNTH2M
```

```
      IF(IEND.EQ.1.AND.RAD1(K).EQ.RAD1(K-1)) Y(I)=XNTH1H+XNTH1M
      IF(IEND.EQ.2.AND.RAD2(K).EQ.RAD1(K+1)) Y(I)=XNTH2H+XNTH2M
      IF(IEND.EQ.2.AND.RAD1(K).EQ.RAD1(K+1)) Y(I)=XNTH1H+XNTH1M
      ZETA=ZETA+XINCRM*PI/180.
70    CONTINUE
      CALL GRAPH(X,Y)
      IF(II.EQ.7) GOTO 15
701   CONTINUE
700   FORMAT(//,3X,'TANGENTIAL STRESS RESULTANT AS A FUNCTION
     $ OF ANGLE (1=YES,2=NO)')
      GOTO 1
C
C
C
C   CHECKING IF BOTH ENDS HAVE BEEN CALCULATED
C
15    IF(IEND.EQ.1) GOTO 2
      RETURN
      END
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                          C
C                                                                          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                          C
C    SUBROUTINE HEMI DETERMINES THE RADIAL DEFLECTION,                     C
C    MERIDIONAL ROTATION, TANGENTIAL MOMENT, MERIDIONAL                    C
C    MOMENT, SHEAR STRESS RESULTANT, MERIDIONAL STRESS                     C
C    RESULTANT, AND THE TANGENTIAL STRESS RESULTANT AS A                   C
C    FUNCTION OF THE MERIDIONAL LENGTH (PHI).                              C
C                                                                          C
C                                                                          C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      SUBROUTINE HEMI(X,Y)
      COMMON/ARRAYS/ITYPE(10),RAD1(10),RAD2(10),XLENT(10),
     $              PRES(10),THIC(10),XNU(10),E(10),LOAD(10)
      COMMON/CONST/XINCRM,L,XK,PI,K,J,NUMELE,RAD,PHICNE,PHITWO
      COMMON/FACT/F1,F2,F3,F4,F5,F6,F7,F8,F9,F10,F11,F12,
     $              F13,F14,F15,F16,F17,F18,F19,F20,F1C,F2C,
     $              F3C,F4C,F5C,F6C,F7C,F8C,F9C,F10C,F11C,
     $              F12C,F13C,F14C,F15C,F16C,F17C,F18C,F19C,F20C,
     $              G2,G5,G8
      DIMENSION X(L),Y(L)
C
C
      X(1)=0
      Y(1)=0
      READ(1,*) XM,XH
      XK=(3.*(1.-XNU(K)**2)*(RAD/THIC(K))**2)**.25
      II=0
```

```
  1     CONTINUE
        WRITE(5,2)
2       FORMAT(/,5X,'HEMISPERICAL HEAD')
        II=II+1
        IF(II.GT.7) GOTO 15
        GOTO(1000,2000,3000,4000,5000,6000,7000),II
C
C
C  (1.)  RADIAL DEFLECTION AS A FUNCTION OF PHI.
C
C
1000    CONTINUE
        WRITE(5,100)
        READ(5,*) LL
        IF(LL.EQ.2) GOTO 101
        I=0
        ZETA=0
        DO 10 I=2,L
        X(I)=ZETA*180./PI
        CALL FACTOR(ZETA)
        DELRM=-((XM*RAD*XK*SIN(PHIONE-ZETA)*SIN(PHIONE))/(E(K)*THIC(K)))
     $*(-F9-(2.*F4C/F1C)*F7+(F2C/F1C)*F10)
        DELRM=-XM*2.*(XK**2)*SIN(PHIONE-ZETA)*((F6C/F1C)*F14+(F5C/F1C)
     $*F13-(F3C/F1C)*F10)*(1./(E(K)*THIC(K)))
        Y(I)=DELRH+DELRM
        WRITE(2,*) DELRH,Y(I),X(I)
        ZETA=ZETA+XINCRM
10      CONTINUE
        CALL GRAPH(X,Y)
        IF(II.EQ.7) GOTO 15
101     CONTINUE
100     FORMAT(/,5X,'DELTA AS A FUNCTION OF ANGLE (1=YES,2=NO)')
        GOTO 1
C
C
C
C  (2.)  MERIDIONAL ROTATION AS A FUNCTION OF PHI.
C
C
C
C
2000    CONTINUE
        WRITE(5,200)
        READ(5,*) LL
        IF(LL.EQ.2) GOTO 201
        I=0
        ZETA=0
        DO 20 I=2,L
        CALL FACTOR(ZETA)
        X(I)=ZETA*(180./PI)
        DETAM=-((XM*2.*(XK**2)*SIN(PHIONE))/(E(K)*THIC(K)))*
     $(-F8+(F4C/F1C)*F5+(F2C/F1C)*F7)
        DETAM=-((XM*4.*(XK**3))/(E(K)*THIC(K)*RAD))*((F8C/F1C)*F15
```

```
     $-(F5C/F1C)*F15-(F3C/F1C)*F7)
      Y(I)=BETAH+BETAM
      WRITE(2,*) BETAH,Y(I),X(I)
      ZETA=ZETA+XINCRM
20    CONTINUE
      CALL GRAPH(X,Y)
      IF(II.EQ.7) GOTO 15
200   FORMAT(///,3X,'ROTATION AS A FUNCTION OF ANGLE (1=YES,2=NO)')
201   CONTINUE
      GOTO 1
C
C
C
C
C  (3.)  SHEAR STRESS RESULTANT AS A FUNCTION OF PHI.
C
C
3000  CONTINUE
      WRITE(5,300)
      READ(5,*) LL
      IF(LL.EQ.2) GOTO 301
      I=0
      ZETA=0
      DO 30 I=2,L
      CALL FACTOR(ZETA)
      X(I)=ZETA*(180./PI)
      XQPHM=XM*SIN(PHIONE)*(F7-(F4C/F1C)*F10+(F2C/F1C)*F8)
      XQPHM=(-XM*2.*XK/RAD)*((F6C/F1C)*F15+(F5C/F1C)*F16
     $-(F3C/F1C)*F8)
      Y(I)=XQPHM+XQPHH
      ZETA=ZETA+XINCRM
30    CONTINUE
      CALL GRAPH(X,Y)
      IF(II.EQ.7) GOTO 15
300   FORMAT(///,3X,'SHEAR STRESS RESULTANT AS A FUNCTION OF ANGLE
     $ (1=YES,2=NO)')
301   CONTINUE
      GOTO 1
C
C
C
C  (4.)  MERIDIONAL STRESS RESULTANT AS A FUNCTION OF PHI.
C
4000  CONTINUE
      WRITE(5,400)
      READ(5,*) LL
      IF(LL.EQ.2) GOTO 401
      I=0
      ZETA=0
      DO 40 I=2,L
      CALL FACTOR(ZETA)
      X(I)=ZETA*(180./PI)
      XNPHM=-XM*SIN(PHIONE)*COTAN(PHIONE-ZETA)*(F7-(F4C/F1C)*
```

```
     $F10+(F2C/F1C)*F8)
      XNPHM=((XM*COTAN(PHIONE-ZETA)*2.*XK)/RAD)*((F6C/F1C)*F15
     $+(F5C/F1C)*F10-(F3C/F1C)*F8)
40       Y(I)=XNPHM+XNPHM
      ZETA=ZETA+XINCRM
40    CONTINUE
      CALL GRAPH(X,Y)
      IF(II.EQ.7) GOTO 15
400   FORMAT(///,5X,'MERIDIONAL STRESS RESULTANT AS A FUNCTION OF THE ANGL
     $ (1=YES,2=NO)')
401   CONTINUE
      GOTO 1
C
C
C
C  (5.)  TANGENTIAL STRESS RESULTANT AS A FUNCTION OF PHI.
C
C
5000  CONTINUE
      WRITE(5,500)
      READ(5,*) LL
      IF(LL.EQ.2) GOTO 501
      I=0
      ZETA=0
      DO 50 I=2,L
      X(I)=ZETA*(180./PI)
      CALL FACTOR(ZETA)
      XNTHM=-XM*XK*SIN(PHIONE)*(-F9-(2.*F4C/F1C)*F7+(F2C/F1C)*F10)
      XNTHM=-((XM**2.*XK**2)/RAD)*((F6C/F1C)*F14+(F5C/F1C)*F13
     $-(F3C/F1C)*F10)
      Y(I)=XNTHM+XNTHM
      ZETA=ZETA+XINCRM
50    CONTINUE
      CALL GRAPH(X,Y)
      IF(II.EQ.7) GOTO 15
500   FORMAT(///,5X,'TANG. STRESS RESULTANT AS A FUNCTION OF ANGLE
     $ (1=YES,2=NO)')
501   CONTINUE
      GOTO 1
C
C
C
C  (6.)  MERIDIONAL MOMENT AS A FUNCTION OF PHI.
C
C
6000  CONTINUE
      WRITE(5,600)
      READ(5,*) LL
      IF(LL.EQ.2) GOTO 601
      I=0
      ZETA=0
      DO 60 I=2,L
      X(I)=ZETA*(180./PI)
```

```
      CALL FACTOR(ZETA)
      XMPHH=(XH*RAD*SIN(PHIONE)/(2.*XK))*(-F10+(2.*F4C/F1C)*F8
     $-(F2C/F1C)*F8-(F2C/F1C)*F9)
      XMPHM=XM*((F6C/F1C)*F13-(F5C/F1C)*F14+(F3C/F1C)*F9)
      Y(I)=XMPHH+XMPHM
      ZETA=ZETA+XINCRM
50    CONTINUE
      CALL GRAPH(X,Y)
      IF(II.EQ.7) GOTO 15
600   FORMAT(///,3X,'MERI. MOMENT AS A FUNCTION OF ANGLE (1=YES,2=NO)')
601   CONTINUE
      GOTO 1
C
C
C
C
C  (7.)  TANGENTIAL MOMENT AS A A F.UNCTION OF PHI.
C
C
C
7000  CONTINUE
      WRITE(5,700)
      READ(5,*) LL
      IF(LL.EQ.2) GOTO 701
      I=0
      ZETA=0
      DO 70 I=2,L
      CALL FACTOR(ZETA)
      XMPHH=((XH*RAD*SIN(PHIONE))/(2.*XK))*(-F10+(2.*F4C/F1C
     $)*F9)
      XMPHH=(XH*RAD*SIN(PHIONE)/(2.*XK))*(-F10+(2.*F4C/F1C)*F8
     $-(F2C/F1C)*F8-(F2C/F1C)*F9)
      XMPHM=XM*((F6C/F1C)*F13-(F5C/F1C)*F14+(F3C/F1C)*F9)
      Y(I)=XMPHH*XNU(K)+XMPHM*XNU(K)
      ZETA=ZETA+XINCRM
70    CONTINUE
      CALL GRAPH(X,Y)
701   CONTINUE
      IF(II.EQ.7) GOTO 15
700   FORMAT(///,3X,'TANG. MOMENT AS A FUNCTION OF ANGLE (1=YES,2=NO)')
      GOTO 1
C
C
15    CONTINUE
      RETURN
      END
C
C
C
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                    C
C                                                                    C
C       SUBROUTINE UPNCON CALCULATES THE RADIAL DEFLECTION           C
C       MERIDIONAL ROTATION, SHEAR STRESS RESULTANT,                 C
C       MERIDIONAL STRESS RESULTANT, TANGENTIAL STRESS               C
```

```
C     RESULTANT,MERIDIONAL MOMENT AS A FUNCTION OF THE          C
C     ACTUAL LENGTH FOR TRUNCATED CONICAL FRUSTRUMS.            C
C                                                               C
C                                                               C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C
      SUBROUTINE CPMCON(X,Y,X1,X2,ALPHA,ALPHA2,PHI,XL)
      COMMON/ARRAYS/ITYPE(10),RAD1(10),RAD2(10),XLENT(10),
     $              PRES(10),THIC(10),XNU(10),E(10),LOAD(10)
      COMMON/CONST/XINCRM,L,XK,PI,K,D,NUMELE,RAD,PHICNE,PHITWO
      COMMON/FACT/F1,F2,F3,F4,F5,F6,F7,F8,F9,F10,F11,F12,
     $              F13,F14,F15,F16,F17,F18,F19,F20,F1C,F2C,
     $              F3C,F4C,F5C,F6C,F7C,F8C,F9C,F10C,F11C,
     $              F12C,F13C,F14C,F15C,F16C,F17C,F18C,F19C,F20C,
     $              G2,G5,G8
      DIMENSION X(L),Y(L)
C
C
      X(1)=0
      Y(1)=0
      IF(RAD2(K).NE.0) READ(1,*) XM1,XH1,XM2,XH2
      IF(RAD2(K).EQ.0) READ(1,*) XM1,XH1
      XLENT(K)=X2-X1
      IF(RAD2(K).NE.0) GOTO 8
      XM11=XM1
      XH11=XH1
      GOTO 2
5     IF(RAD2(K).EQ.RAD1(K-1)) GOTO 11
      XH11=XH1
      XM11=-XM1
      XH22=XH2
      XM22=XM2
      GOTO 2
11    XH22=XH1
      XM22=-XM1
      XH11=XH2
      XM11=XM2
      IEND=0
2     II=0
      IEND=IEND+1
1     CONTINUE
      IF(IEND.EQ.1.AND.RAD2(K).EQ.RAD1(K-1)) XM=X1
      IF(IEND.EQ.2.AND.RAD1(K).EQ.RAD1(K+1)) XM=X2
      IF(IEND.EQ.1.AND.RAD1(K).EQ.RAD1(K-1))  XM=X2
      IF(IEND.EQ.2.AND.RAD2(K).EQ.RAD1(K+1)) XM=X1
      IF(RAD2(K).EQ.0) XM=X2
      XK=(3.*(1.-XNU(K)**2))**.25/(THIC(K)*XM*COTAN(ALPHA))**.5
      D=(E(K)*THIC(K)**3)/(12.*(1.-XNU(K)**2))
      II=II+1
      IF(II.GT.7) GOTO 15
      IF(RAD2(K).NE.0) WRITE(5,3)
      IF(RAD2(K).EQ.0) WRITE(5,6)
```

```
3     FORMAT(/,3X,'TRUNCATED CONICAL SECTION')
      IF(RAD2(K).EQ.0) GOTO 7
5     FORMAT(/,3X,'CONICAL END')
      IF(IEND.EQ.1.AND.RAD2(K).EQ.RAD1(K-1)) WRITE(5,4)
      IF(IEND.EQ.2.AND.RAD2(K).EQ.RAD1(K+1)) WRITE(5,4)
      IF(IEND.EQ.1.AND.RAD1(K).EQ.RAD1(K-1)) WRITE(5,5)
      IF(IEND.EQ.2.AND.RAD1(K).EQ.RAD1(K+1)) WRITE(5,5)
4     FORMAT(/,3X,'SMALL END')
5     FORMAT(/,3X,'LARGE END')
7     CONTINUE
      GOTO(1000,2000,3000,4000,5000,6000,7000),II
C
C
C
C  (1.)  RADIAL DEFLECTION AS A FUNCTION OF THE ACTUAL
C        LENGTH OF THE SHELL.
C
C
1000  CONTINUE
      WRITE(5,100)
      READ(5,*) LL
      IF(LL.EQ.2) GOTO 101
      I=0
      ZETA=0
      DO 10 I=2,L
      X(I)=ZETA*(X2-X1)
      CALL FACTOR(ZETA)
      DELR1M=((XM11*SIN(ALPHA)**2)/(2.*D*XK**3))*(-(F9C/F1C)*F7
     $+(F8C/F1C)*F10)
      DELR2M=((XM22*SIN(ALPHA)**2)/(4.*D*XK**3))*(F9+(2.*F4C/F1C)
     $*F7-(F2C/F1C)*F10)
      DELR1M=-((XM11*SIN(ALPHA))/(2.*D*XK**2))*((2.*F8C/F1C)*F7
     $-(F10C/F1C)*F10)
      DELR2M=((XM22*SIN(ALPHA))/(2.*D*XK**2))*((F6C/F1C)*F14+
     $(F5C/F1C)*F13-(F3C/F1C)*F10)
      IF(RAD2(K).NE.0.AND.RAD1(K).EQ.RAD1(K-1)) DELR1M=-DELR1M
      IF(RAD2(K).NE.0.AND.RAD1(K).EQ.RAD1(K-1)) DELR2M=-DELR2M
      IF(RAD2(K).EQ.0.AND.K.EQ.NUMELE) DELR1M=-DELR1M
      IF(IEND.EQ.1.AND.RAD1(K).EQ.RAD1(K-1)) Y(I)=DELR1H+DELR1M
      IF(IEND.EQ.2.AND.RAD1(K).EQ.RAD1(K+1)) Y(I)=DELR1H+DELR1M
      IF(IEND.EQ.1.AND.RAD2(K).EQ.RAD1(K-1)) Y(I)=DELR2H+DELR2M
      IF(IEND.EQ.2.AND.RAD2(K).EQ.RAD1(K+1)) Y(I)=DELR2H+DELR2M
      IF(RAD2(K).EQ.0) Y(I)=DELR1M+DELR1H
      ZETA=ZETA+(XINCRM/(X2-X1))
10    CONTINUE
      CALL GRAPH(X,Y)
      WRITE(5,*) ZETA
      WRITE(5,*) L
      IF(II.EQ.7) GOTO 15
101   CONTINUE
100   FORMAT(/,3X,'DELTA AS A FUNCTION OF SHELL LENGTH (1=YES,2=NO)')
      GOTO 1
C
C
```

```
C
C
C  (2.)  MERIDIONAL ROTATION AS A FUNCTION OF THE ACTUAL SHELL LENGTH.
C
C
2000  CONTINUE
      WRITE(5,200)
      READ(5,*) LL
      IF(LL.EQ.2) GOTO 201
      I=0
      ZETA=0
      DO 20 I=2,L
      X(I)=ZETA*(X2-X1)
      CALL FACTOR(ZETA)
      BETA1H=((XH11*SIN(ALPHA))/(2.*D*XK**2))*((F9C/F1C)*F9
     $+(2.*F8C/F1C)*F7)
      BETA2H=-((XH22*SIN(ALPHA))/(2.*D*XK**2))*(-F8+(F4C/F1C)*F9
     $+(F2C/F1C)*F7)
      BETA1M=((XM11/(D*XK))*((F3C/F1C)*F9+(F1DC/F1C)*F7)
      BETA2M=((XM22/(D*XK))*((F6C/F1C)*F10-(F5C/F1C)*F10-(F3C/F1C)*F7)
      IF(RAD2(K).NE.0.AND.RAD1(K).EQ.RAD1(K-1)) BETA1H=-BETA1H
      IF(RAD2(K).NE.0.AND.RAD1(K).EQ.RAD1(K-1)) BETA2H=-BETA2H
      IF(RAD2(K).EQ.0.AND.K.EQ.NUMELE) BETA1H=-BETA1H
      IF(IEND.EQ.1.AND.RAD1(K).EQ.RAD1(K-1)) Y(I)=BETA1H+BETA1M
      IF(IEND.EQ.2.AND.RAD1(K).EQ.RAD1(K+1)) Y(I)=BETA1H+BETA1M
      IF(IEND.EQ.1.AND.RAD2(K).EQ.RAD1(K-1)) Y(I)=BETA2H+BETA2M
      IF(IEND.EQ.2.AND.RAD2(K).EQ.RAD1(K+1)) Y(I)=BETA2H+BETA2M
      IF(RAD2(K).EQ.0) Y(I)=BETA1H+BETA1M
      ZETA=ZETA+(XINCRM/(X2-X1))
20    CONTINUE
      CALL GRAPH(X,Y)
      IF(II.EQ.7) GOTO 15
201   CONTINUE
200   FORMAT(/,5X,'MERIDIONAL ROTATION AS A FUNCTION OF SHELL LENGTH
     $ (1=YES,2=NO)')
      GOTO 1
C
C
C
C
C  (3.)  SHEAR STRESS RESULTANT AS A FUNCTION OF THE ACTUAL LENGTH
C        OF THE SHELL.
C
C
3000  CONTINUE
      WRITE(5,300)
      READ(5,*) LL
      IF(LL.EQ.2) GOTO 301
      WRITE(5,*) K,NUMELE
      I=0
      ZETA=0
      DO 30 I=2,L
      X(I)=ZETA*(X2-X1)
      CALL FACTOR(ZETA)
      XQ1H=(XH11*SIN(ALPHA))*(-(F9C/F1C)*F10+(2.*F8C/F1C)*F8)
```

```
      XQ2M=XM22*SIN(ALPHA)*(F7-(F4C/F1C)*F10+(F2C/F1C)*F8)
      XQ1M=(XM11*2.*XK)*((F8C/F1C)*F10-(F10C/F1C)*F8)
      XQ2M=(XM22*2.*XK)*((F6C/F1C)*F15+(F5C/F1C)*F16-(F3C/F1C)*F8)
      IF(RAD2(K).NE.0.AND.RAD1(K).EQ.RAD1(K-1)) XQ1M=-XQ1M
      IF(RAD2(K).NE.0.AND.RAD1(K).EQ.RAD1(K-1)) XQ2M=-XQ2M
      IF(RAD2(K).EQ.0.AND.K.EQ.NUMELE) XQ1M=-XQ1M
      IF(IEND.EQ.1.AND.RAD1(K).EQ.RAD1(K-1)) Y(I)=XQ1H+XQ1M
      IF(IEND.EQ.2.AND.RAD1(K).EQ.RAD1(K+1)) Y(I)=XQ1H+XQ1M
      IF(IEND.EQ.1.AND.RAD2(K).EQ.RAD1(K-1)) Y(I)=XQ2H+XQ2M
      IF(IEND.EQ.2.AND.RAD2(K).EQ.RAD1(K+1)) Y(I)=XQ2H+XQ2M
      IF(RAD2(K).EQ.0) Y(I)=XQ1H+XQ1M
      ZETA=ZETA+(XINCRM/(X2-X1))
30    CONTINUE
      CALL GRAPH(X,Y)
      IF(II.EQ.7) GOTO 15
301   CONTINUE
300   FORMAT(/,5X,'SHEAR STRESS RESULTANT AS A FUNCTION OF SHELL
     $ LENGTH (1=YES,2=NO)')
      GOTO 1
C
C
C
C  (4.)  MERIDIONAL STRESS RESULTANT AS A FUNCTION OF THE ACTUAL
C        SHELL LENGTH.
C
C
400   CONTINUE
      WRITE(5,400)
      READ(5,*) LL
      IF(LL.EQ.2) GOTO 401
      I=0
      ZETA=0
      DO 40 I=2,L
      X(I)=ZETA*(X2-X1)
      CALL FACTOR(ZETA)
      XNX1H=-XM11*COS(ALPHA)*(-(F9C/F1C)*F10+(2.*F8C/F1C)*F8)
      XNX2H=-XM22*COS(ALPHA)*(F7-(F4C/F1C)*F10+(F2C/F1C)*F8)
      XNX1M=-XM11*2.*XK*COTAN(ALPHA)*((F8C/F1C)*F10-(F10C/F1C)*F8)
      XNX2M=-XM22*2.*XK*COTAN(ALPHA)*((F6C/F1C)*F15+(F5C/F1C)*F16
     $-(F3C/F1C)*F8)
      IF(RAD2(K).NE.0.AND.RAD1(K).EQ.RAD1(K-1)) XNX1M=-XNX1M
      IF(RAD2(K).NE.0.AND.RAD1(K).EQ.RAD1(K-1)) XNX2M=-XNX2M
      IF(RAD2(K).EQ.0.AND.K.EQ.NUMELE) XNX1M=-XNX1M
      IF(IEND.EQ.1.AND.RAD1(K).EQ.RAD1(K-1)) Y(I)=XNX1H+XNX1M
      IF(IEND.EQ.2.AND.RAD1(K).EQ.RAD1(K+1)) Y(I)=XNX1H+XNX1M
      IF(IEND.EQ.1.AND.RAD2(K).EQ.RAD1(K-1)) Y(I)=XNX2H+XNX2M
      IF(IEND.EQ.2.AND.RAD2(K).EQ.RAD1(K+1)) Y(I)=XNX2H+XNX2M
      IF(RAD2(K).EQ.0) Y(I)=XNX1H+XNX1M
      ZETA=ZETA+(XINCRM/(X2-X1))
40    CONTINUE
      CALL GRAPH(X,Y)
      IF(II.EQ.7) GOTO 15
401   CONTINUE
400   FORMAT(/,5X,'MERIDIONAL STRESS RESULTANT AS A FUNCTION OF SHELL
```

```
     $ LENGTH (1=YES,2=NO)')
      GOTO 1
C
C
C
C
C  (5.)  TANGENTIAL STRESS RESULTANT AS A FUNCTION OF THE ACTUAL
C        SHELL LENGTH.
C
C
5000  CONTINUE
      WRITE(5,5001)
      READ(5,*) LL
      IF(LL.EQ.2) GOTO 501
      I=0
      ZETA=0
      DO 50 I=2,L
      X(I)=ZETA*(X2-X1)
      CALL FACTOR(ZETA)
      XNTH1M=2.*XM11*XM*XK*COS(ALPHA)*(-(F9C/F1C)*F7+(F8C/F1C)*F10)
      XNTH2M=XM22*XM*XK*COS(ALPHA)*(F9*(2.*F4C/F1C)*F7-(F2C/F1C)*F10)
      XNTH1M=-XM11*2.*XK**2*XM*COTAN(ALPHA)*((2.*F8C/F1C)
     $*F7-(F10C/F1C)*F10)
      XNTH2M=XM22*2.*XK**2*XM*COTAN(ALPHA)*((F6C/F1C)*F14+
     $(F5C/F1C)*F13-(F3C/F1C)*F10)
      IF(RAD2(K).NE.0.AND.RAD1(K).EQ.RAD1(K-1)) XNTH1M=-XNTH1M
      IF(RAD2(K).NE.0.AND.RAD1(K).EQ.RAD1(K-1)) XNTH2M=-XNTH2M
      IF(RAD2(K).EQ.0.AND.K.EQ.NUMELE) XNTH1M=-XNTH1M
      IF(IEND.EQ.1.AND.RAD1(K).EQ.RAD1(K-1)) Y(I)=XNTH1H+XNTH1M
      IF(IEND.EQ.2.AND.RAD1(K).EQ.RAD1(K+1)) Y(I)=XNTH1H+XNTH1M
      IF(IEND.EQ.1.AND.RAD2(K).EQ.RAD1(K-1)) Y(I)=XNTH2H+XNTH2M
      IF(IEND.EQ.2.AND.RAD2(K).EQ.RAD1(K+1)) Y(I)=XNTH2H+XNTH2M
      IF(RAD2(K).EQ.0) Y(I)=XNTH1H+XNTH1M
      ZETA=ZETA+(XINCRM/(X2-X1))
50    CONTINUE
      CALL GRAPH(X,Y)
      IF(II.EQ.7) GOTO 15
501   CONTINUE
5001  FORMAT(/,3X,'TANGENTIAL STRESS RESULTANT AS A FUNCTION OF SHELL
     $ LENGTH (1=YES,2=NO)')
      GOTO 1
C
C
C
C
C  (6.)  MERIDIONAL MOMENT AS A FUNCTION OF THE ACTUAL SHELL LENGTH.
C
C
6000  CONTINUE
      WRITE(5,6001)
      READ(5,*) LL
      IF(LL.EQ.2) GOTO 601
      I=0
      ZETA=0
      DO 60 I=2,L
```

```
      X(I)=ZETA*(X2-X1)
      CALL FACTOR(ZETA)
      XMX1H=((XH11*SIN(ALPHA))/XK)*((F9C/F1C)*F8-(F8C/F1C)*F9)
      XMX2H=((XH22*SIN(ALPHA))/(2.*XK))*(F10-(2.*F4C/F1C)*F8
     $+(F2C/F1C)*F9)
      XMX1M=XM11*((2.*F8C/F1C)*F8-(F10C/F1C)*F9)
      XMX2M=-XM22*((F6C/F1C)*F13-(F5C/F1C)*F14+(F3C/F1C)*F9)
      IF(RAD2(K).EQ.0.AND.K.EQ.NUMELE) XMX1H=-XMX1H
      IF(RAD2(K).NE.0.AND.RAD1(K).EQ.RAD1(K-1)) XMX1H=-XMX1H
      IF(RAD2(K).NE.0.AND.RAD1(K).EQ.RAD1(K-1)) XMX2H=-XMX2H
      IF(IEND.EQ.1.AND.RAD1(K).EQ.RAD1(K-1)) Y(I)=XMX1H+XMX1M
      IF(IEND.EQ.2.AND.RAD1(K).EQ.RAD1(K+1)) Y(I)=XMX1H+XMX1M
      IF(IEND.EQ.1.AND.RAD2(K).EQ.RAD1(K-1)) Y(I)=XMX2H+XMX2M
      IF(IEND.EQ.2.AND.RAD2(K).EQ.RAD1(K+1)) Y(I)=XMX2H+XMX2M
      IF(RAD2(K).EQ.0) Y(I)=XMX1H+XMX1M
      ZETA=ZETA+(XINCRM/(X2-X1))
50    CONTINUE
      CALL GRAPH(X,Y)
      IF(II.EQ.7) GOTO 15
501   CONTINUE
555   FORMAT(/,3X,'AXIAL MOMENT AS A FUNCTION OF SHELL LENGTH
     $ (1=YES,2=NO)')
      GOTO 1
C
C
C
C
C  (7.)  TANGENTIAL MOMENT AS A FUNCTION OF THE ACTUAL SHELL LENGTH.
C
C
7000  CONTINUE
      WRITE(5,700)
      READ(5,*) LL
      IF(LL.EQ.2) GOTO 701
      I=0
      ZETA=0
      DO 70 I=2,L
      X(I)=ZETA*(X2-X1)
      CALL FACTOR(ZETA)
      XMX1H=((XH11*SIN(ALPHA))/XK)*((F9C/F1C)*F8-(F8C/F1C)*F9)
      XMX2H=-((XH22*SIN(ALPHA))/(2.*XK))*(F10-(2.*F4C/F1C)*F8
     $+(F2C/F1C)*F9)
      XMX1M=XM11*((2.*F8C/F1C)*F8-(F10C/F1C)*F9)
      XMX2M=-XM22*((F6C/F1C)*F13-(F5C/F1C)*F14+(F3C/F1C)*F9)
      IF(RAD2(K).EQ.0.AND.K.EQ.NUMELE) XMX1H=-XMX1H
      IF(RAD2(K).NE.0.AND.RAD1(K).EQ.RAD1(K-1)) XMX1H=-XMX1H
      IF(RAD2(K).NE.0.AND.RAD1(K).EQ.RAD1(K-1)) XMX2H=-XMX2H
      IF(IEND.EQ.1.AND.RAD1(K).EQ.RAD1(K-1)) Y(I)=XNU(K)*(XMX1H+XMX1M)
      IF(IEND.EQ.2.AND.RAD1(K).EQ.RAD1(K+1)) Y(I)=XNU(K)*(XMX1H+XMX1M)
      IF(IEND.EQ.1.AND.RAD2(K).EQ.RAD1(K-1)) Y(I)=XNU(K)*(XMX2H+XMX2M)
      IF(IEND.EQ.2.AND.RAD2(K).EQ.RAD1(K+1)) Y(I)=XNU(K)*(XMX2H+XMX2M)
      IF(RAD2(K).EQ.0) Y(I)=XNU(K)*(XMX1H+XMX1M)
      ZETA=ZETA+(XINCRM/(X2-X1))
70    CONTINUE
      CALL GRAPH(X,Y)
```

```
      IF(II.EQ.7) GOTO 15
 701  CONTINUE
 700  FORMAT(/,3X,'TANGENTIAL MOMENT AS A FUNCTION OF SHELL LENGTH
     $ (1=YES,2=NO)')
C
C
 15    IF(RAD2(K).EQ.0) RETURN
       IF(IEND.EQ.1) GOTO 2
       RETURN
       END
C
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C     SUBROUTINE PLATE CLACULATES DEFLECTION AND ROTATION              C
C    OF THE MERIDIAN AND MERIDIONAL MOMENT, CIRCUMFERENTIAL            C
C    MOMENT AND SHEAR STRESS RESULTANT AS FUNCTION OF THE              C
C    RADIUS FOR FLAT CIRCULAR PLATES.                                  C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C
C
C
C
      SUBROUTINE PLATE(X,Y)
      COMMON/ARRAYS/ITYPE(10),RAD1(10),RAD2(10),XLENT(10),
     $              PRES(10),THIC(10),XNU(10),E(10),LOAD(10)
      COMMON/CONST/XINCRM,L,XK,PI,K,D,NUMELE,RAD,PHICNE,PHITHC
      COMMON/FACT/F1,F2,F3,F4,F5,F6,F7,F8,F9,F10,F11,F12,
     $              F13,F14,F15,F16,F17,F18,F19,F20,F1C,F2C,
     $              F3C,F4C,F5C,F6C,F7C,F8C,F9C,F10C,F11C,
     $              F12C,F13C,F14C,F15C,F16C,F17C,F18C,F19C,F20C,
     $              G2,G5,G6
      DIMENSION X(L),Y(L)
C
C
C
      X(1)=0
      Y(1)=0
      READ(1,*) XM,XH
      IF(K.EQ.NUMELE) XM=-XM
      D=(E(K)*THIC(K)**3)/(12.*(1.-XNU(K)**2))
      II=0
 1    CONTINUE
      IF(II.EQ.4) GOTO 15
      WRITE(5,2)
 2    FORMAT(/,3X,'FLAT PLATE')
      II=II+1
      GOTO(1000,2000,3000,4000),II
C
C
C
```

```
C
C
C       (1.)  DELTA AS A FUNCTION OF RADIUS
C
C
1000   CONTINUE
       WRITE(5,100)
       READ(5,*) LL
       IF(LL.EQ.2) GOTO 101
       I=0
       ZETA=0
       DO 10 I=2,L
       X(I)=ZETA
       IF(ZETA.EQ.RAD1(K)) Y(I)=Y(I-1)
       IF(ZETA.EQ.RAD1(K)) GOTO 10
C       CALL FACTOR(ZETA)
C       DELPRE=(PRES(K)/(10*64.))*(RAD1(K)**2-(RAD1(K)-ZETA)**2)**2
       XKII=((RAD1(K)-ZETA)/RAD1(K))
       DELRM=((XM*RAD1(K)**2)/(2.*D*(1.+XNU(K))))*(1.-XKII**2)
       DELPRE=((PRES(K)*RAD1(K)**2*(1.-XKII**2))/(64.*D*PI))
      &*((5.+XNU(K))/(1.+XNU(K)))-XKII**2)
      .DELPRE=0
       Y(I)=DELPRE+DELRM
       ZETA=ZETA+XINCRM
10     CONTINUE
       CALL GRAPH(X,Y)
       IF(II.EQ.5) GOTO 15
101    CONTINUE
100    FORMAT(/,3X,'DELTA AS A FUNCTION OF RADIUS')
       GOTO 1
C
C
C
C
C
C
C       (2.)  ROTATION AS A FUNCTION OF THE RADIUS
C
C
2000   CONTINUE
       WRITE(5,200)
       READ(5,*) LL
       IF(LL.EQ.2) GOTO 201
       I=0
       ZETA=0
       DO 20 I=2,L
       X(I)=ZETA
C       IF(ZETA.EQ.RAD1(K)) Y(I)=Y(I-1)
C       IF(ZETA.EQ.RAD1(K)) GOTO 20
C       CALL FACTOR(ZETA)
C       DETAM=((XM*(RAD1(K)-ZETA))/D)*05
C       BETAPR=-((PRES(K)*(RAD1(K)-ZETA))/(6.*D))*(RAD1(K)**2
C      &-(RAD1(K)-ZETA)**2)
       XKII=((RAD1(K)-ZETA)/RAD1(K))
       BETAPR=((PRES(K)*RAD1(K)*XKII)/(16.*D*PI))
```

```
      WRITE(2,*) X(I),Y(I)
10    CONTINUE
      CALL INITT(120)
      CALL BINITT
      CALL CHECK(X,Y)
      CALL LINE(I)
      CALL SYMBL(0)
      CALL DSPLAY(X,Y)
C      CALL MOVABS(50,620)
C      CALL VLABEL(21,LABELY)
C      CALL MOVABS(410,50)
C      CALL HLABEL(15,LABELX)
C      CALL ROTATE(410,765,14,LILLEX)
C      CALL MOVABS(370,700)
C      CALL HLABEL(15,LILLEX)
C      CALL MOVABS(435,725)
C      CALL HLABEL(15,LJUST)
      CALL TINPUT(JUNK)
C      CALL BINITT
      CALL INITT(30)
      RETURN
      END
C
C
C
C
C
C
C
C
```

```
C
C       (4.)  TANGENTIAL MOMENT AS A FUNCTION OF RADIUS
C
C
400   CONTINUE
      WRITE(5,40C)
      READ(5,*) LL
      IF(LL.EQ.2) GOTO 401
      I=0
      ZETA=0
      DO 40 I=2,L
      X(I)=ZETA
      IF(ZETA.EQ.RAD1(K)) Y(I)=Y(I-1)
      IF(ZETA.EQ.RAD1(K)) GOTO 40
      XKII=((RAD1(K)-ZETA)/RAD1(K))
      RADMTP=(PRES(K)/(16.*PI))*(3.+XNU(K)-(1.+3.*XNU(K))*XKII**2)
      RADMMT=XM
      RADMTP=0
      Y(I)=RADMPT+RADMMT
      ZETA=ZETA+XINCRM
40    CONTINUE
      CALL GRAPH(X,Y)
      IF(II.EQ.5) GOTO 15
401   CONTINUE
400   FORMAT(/,3X,'TANGENTIAL MOMENT AS A FUNCTION OF RADIUS')
      GOTO 1
C
C
C
C
C
C
C
15    CONTINUE
      RETURN
      END
C
C
C
C
C
C
C
C
C
C
      SUBROUTINE GRAPH(X,Y)
      COMMON/CONST/XINCRM,L,XK,PI,K,J,NUMELE
      DIMENSION X(L),Y(L)
      X(1)=L-1
      Y(1)=L-1
      DO 10 I=1,L
```

```
      S*(((3.+XNU(K))/(1.+XNU(K)))-XKII**2)
       BETAM=(XM*RADI(K)*XKII)/(D*(1.+XNU(K)))
       BETAPR=0
       Y(I)=BETAPR+BETAM
       ZETA=ZETA+XINCRM
 20    CONTINUE
       CALL GRAPH(X,Y)
       IF(II.EQ.5) GOTO 15
 201   CONTINUE
 200   FORMAT(/,3X,'ROTATION AS A FUNCTION OF RADIUS')
       GOTO 1
C
C
C
C
C
C
C
C
C      (3.)  RADIAL MOMENT AS FUNCTION OF RADIUS
C
 300   CONTINUE
       WRITE(5,300)
       READ(5,*) LL
        IF(LL.EQ.2) GOTO 301
       I=0
       ZETA=0
       DO 30 I=2,L
       X(I)=ZETA
       IF(ZETA.EQ.RADI(K)) Y(I)=Y(I-1)
       IF(ZETA.EQ.RADI(K)) GOTO 30
C       CALL FACTOR(ZETA)
C       RADMP=(PRES(K)/10.)*(RADI(K)**2*(1.+XNU(K))-((RADI(K)-ZETA)**2
C      1*(3.+XNU(K))))
C       RADMM=XM*GB
       XKII=((RADI(K)-ZETA)/RADI(K))
       RADMP=(PRES(K)/(10.*PI))**(3.+XNU(K))*(1.-XKII**2)
       RADMM=XM
       RADMP=0
       ZETA=ZETA+XINCRM
       Y(I)=RADMP+RADMM
 30    CONTINUE
       CALL GRAPH(X,Y)
       IF(I.EQ.5) GOTO 15
 301   CONTINUE
 300   FORMAT(/,3X,'RADIAL MOMENT AS A FUNCTION OF RADIUS')
       GOTO 1
C
C
C
C
C
C
C
```

VITA

Brian Graham Smith was born in Dayton, Ohio on April 18, 1958. He entered the University of Tennessee in September 1977 and received a Bachelor of Science in Mechanical Engineering in December 1981. After completing a student exchange with the Fachhoschule des Rheinland Phalz, Koblenz West Germany, he returned to the University of Tennessee to pursue a Master's degree in Engineering Science in January 1983.

The author began working for the Oak Ridge National Laboratory in Oak Ridge, Tennessee in April of 1984 where he is presently employed.