CANADIAN THESES ON MICROFICHE

I.S.B.N.

THÈSES CANADIENNES SUR MICROFICHE

National Library of Canada
Collections Development Branch

Canadian Theses on
Microfiche Service

Ottawa, Canada
K1A 0N4

Bibliothèque nationale du Canada
Direction du développement des collections

Service des thèses canadiennes
sur microfiche

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us a poor photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de mauvaise qualité.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE

NL-339 (r. 82/08)

Canada

172 0-315-16081-0

National Library of Canada — Bibliothèque nationale du Canada

Canadian Theses Division — Division des thèses canadiennes

Ottawa, Canada
K1A 0N4

63990

PERMISSION TO MICROFILM — AUTORISATION DE MICROFILMER

• Please print or type — Écrire en lettres moulées ou dactylographier

Full Name of Author — Nom complet de l'auteur

NERISSA HERNANDEZ

Date of Birth — Date de naissance	Country of Birth — Lieu de naissance
AUGUST 17, 1958	PHILIPPINES

Permanent Address — Résidence fixe

514 ROSEBERRY STREET
WINNIPEG, MANITOBA
R3J 1T7

Title of Thesis — Titre de la thèse

ANALYSIS OF AXISYMMETRICALLY LOADED SHELLS OF REVOLUTION

University — Université

UNIVERSITY OF ALBERTA

Degree for which thesis was presented — Grade pour lequel cette thèse fut présentée

MSc

Year this degree conferred — Année d'obtention de ce grade	Name of Supervisor — Nom du directeur de thèse
1983	DR S. H. SIMMONDS

Permission is hereby granted to the NATIONAL LIBRARY OF CANADA to microfilm this thesis and to lend or sell copies of the film.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

L'autorisation est, par la présente, accordée à la BIBLIOTHÈQUE NATIONALE DU CANADA de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans l'autorisation écrite de l'auteur.

Date	Signature
April 8, 1983	Nerissa Hernandez

NL-91 (4/77)

THE UNIVERSITY OF ALBERTA

ANALYSIS OF AXISYMMETRICALLY LOADED SHELLS OF REVOLUTION

by

© NERISSA HERNANDEZ

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF MASTER OF SCIENCE

DEPARTMENT OF CIVIL ENGINEERING

EDMONTON, ALBERTA

SPRING 1983

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR NERISSA HERNANDEZ

TITLE OF THESIS ANALYSIS OF AXISYMMETRICALLY LOADED SHELLS OF REVOLUTION

DEGREE FOR WHICH THESIS WAS PRESENTED MASTER OF SCIENCE

YEAR THIS DEGREE GRANTED SPRING 1983

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

(SIGNED) Nerissa Hernandez

PERMANENT ADDRESS:

514 Roseberry Street

Winnipeg, Manitoba

R3J 1T7

DATED April 8, 1983

THE UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research, for acceptance, a thesis entitled ANALYSIS OF AXISYMMETRICALLY LOADED SHELLS OF REVOLUTION submitted by NERISSA HERNANDEZ in partial fulfilment of the requirements for the degree of MASTER OF SCIENCE.

....
Supervisor

....

....

Date............April 8, 1983...........

ABSTRACT

Two approaches to solving problems involving thin shells based on the standard methods of structural analysis are discussed. In the stiffness method, the governing shell equations are expanded into a Fourier series and reduced to a set of eight first order differential equations. A forward numerical integration technique is used to form the stiffness matrix and the particular solutions. In the flexibility method, the governing shell equations are simplified by limiting the analysis to axisymmetric shells of constant thickness. Closed form solutions are obtained for the flexibility coefficients for specific shell geometries. Particular solutions are approximated by the appropriate membrane solution.

A computer program was developed to perform the flexibility analysis based on the approach presented. The results are compared with the results from a program developed by Shazly (5) based on the stiffness method. The solutions from the two programs show excellent agreement.

ACKNOWLEDGEMENTS

The author wishes to thank Dr. S.H. Simmonds for his invaluable comments. Special thanks to Dr. D.W. Murray and Dr. J.S. Kennedy for serving the thesis committee.

Table of Contents

List of Tables

List of Figures

Figure | Page

NOMENCLATURE

a = radius of curvature of a sphere

$[A_i]$ = matrix coefficients which are a function of the geometric and material properties of the shell

[A] = Boolean Connectivity Matrix

$\{B_i\}$ = load vector coefficients

{C} = constants of integration vector

CA = constant parameter which is a function of the rigidities of the shell and the principal shell curvature

D = flexural rigidity

{D} = segment deformation vector

E = modulus of elasticity

F_o = fixed end forces

$\{F_i\}, \{F_o\}$ = primary and secondary force vectors respectively

[F] = segment flexibility matrix

$[\bar{F}]$ = structure flexibility matrix

h = shell thickness

H = horizontal force, positive in the direction towards the axis of revolution

H_v = fictitional horizontal force due to a vertical edge load at the top of a cone or sphere

$\{h_i\}$ = homogeneous solution vector used in the stiffness analysis

$[H_i]$ = transfer matrix arising from integrating $[A_i]$

[I] = identity matrix

K = extensional rigidity

[K] = segment stiffness matrix

$L(\ )$ = linear differential operator defined in Eqn. A.11

M_s, M_ϕ, M_θ = in-plane bending moments

$M_{\phi\theta}, M_{\theta\phi}, M_{s\theta}, M_{\theta s}$ = twisting moments

n = harmonic number

N_s, N_ϕ, N_θ = normal in-plane forces

$N_{\phi\theta}, N_{\theta\phi}, N_{s\theta}, N_{\theta s}$ = in-plane shear forces

$p_s, p_\phi, p_\theta, p_z$ = intensity of the load components in the directions s, ϕ, θ, z respectively

$[P_i]$ = particular solution vector used in the stiffness analysis

$\{q_0\}$ = structure particular solution used in the flexibility analysis

$\{Q_i\}$ = vector arising from the integration of $\{B_i\}$

Q_d, Qf = coefficients of $\{Q_i\}$

Q_s, Q_ϕ, Q_θ = transverse shear forces

r = radius of the parallel circle for the cylindrical segment

r_0 = radius of a parallel circle

r_1 = radius of curvature of a meridian

r_2 = length of the normal between any point on the midsurface and the axis of revolution

R_0 = curvature of a parallel circle

R_1 = first principal curvature $= 1/r_1$

R_2 = second principal curvature $= 1/r_2$

R = total vertical load acting on a segment due to the applied loads

s = coordinate which measures the distance along the shell meridian

S_i = effective transverse shear force

T_i = effective tangential shear force

[TT] = matrix relating the constants of integration to the redundant vector for the segment; a function of the shell geometry

[TA] = matrix relating the constants of integration to the particular solution displacement vector; a function of the geometric and material properties of the shell

u = displacement component in the circumferential direction

U = change of variable in terms of Q_ϕ and r_2 used to form the homogeneous solution in the flexibility analysis

v = displacement component in the meridional direction

V = change of variable in terms of r_1,v,w, used to form the homogeneous solution in the flexibility analysis

w = displacement component in the radial direction

$\{y_i\}$ = vector of the eight dependent variables in the stiffness analysis

z = coordinate which measures the distance in the direction normal to the midsurface toward the axis of revolution

α = angle between the outer edge of the sphere and the axis of revolution, or the semi-vertex angle for a cone

α_0 = angle between the inner edge of the sphere and the axis of revolution

α_T = coefficient of thermal expansion

β = parameter which is a function of ν, r, and h in the flexibility analysis, or, the meridional rotation in the stiffness analysis

γ = specific weight of the shell or the liquid weight density

$\gamma_{\phi\theta}$ = shear strain

δ, Δ = particular and homogeneous deformations respectively

Δ_H = horizontal displacement of a shell .

Δ_ϕ = meridional rotation of a shell

η = change of variable used to form the homogeneous solution for the cone, defined in Eqn. A.19

θ = coordinate which measures the angle in the circumferential direction

λ = dimensionless parameter which is a function of a/h and ν for the sphere, or the parameter in terms of h and the semi-vertex angle for the cone

ν = Poisson's ratio

ξ = dimensionless input parameter for the evaluation of the Kelvin functions for the cone

$\sigma_\phi, \sigma_\theta$ = meridional and circumferential stresses

$\tau_{\theta\phi}$ = shear stress

ϕ = coordinate which measures the angle between any point on the midsurface and the axis of revolution

$(\ )^{\cdot} = \dfrac{\partial}{\partial \phi}$

$(\ )' = \dfrac{\partial}{\partial \theta}$

$(\ )^{\circ} = \dfrac{\partial}{\partial s}$

1. INTRODUCTION

1.1 Introductory Remarks

A shell of revolution is a surface generated by rotating a plane curve about an axis lying in the same plane. Shells of revolution form part of such structures as pressure vessels, storage tanks, silos, nuclear containment structures, and cooling towers. Apart from their attractive appearance, the widespread use of such shells as structural elements is attributed to their efficiency in resisting load. This leads to thinner sections and reduced material costs.

The general theory of shells of revolution, originally developed by Flügge, applies to any type of meridian geometry with either constant or variable thickness, and subjected to any type of loading. However, for many practical applications the shell segments are of constant thickness and the loads are axisymmetric. The analysis of such shells can be simplified by separating the solution into two parts: firstly, the particular solution approximated by the membrane stresses due to the applied loads; and secondly, the bending stresses due to the edge effects. Moreover, if the shell segments are sufficiently long such that there is virtually no interaction between the edges of a shell segment, the computations can be simplified even further. This method is analogous to the method of consistent deformation in elastic frame analysis. And since accounting for the boundary effects involves evaluating the flexibility coefficients, this method of analysis will be referred to as the flexibility method.

1.2 The Objectives of the Study

The objectives of this study are:

1. To review the solutions to the general theory of shells of revolution;
2. To obtain solutions for the membrane stresses and flexibility influence coefficients in closed form for cylindrical, spherical, and conical segments under various axisymmetric loadings.
3. To incorporate these solutions into the computer program FLEXSHELL.
4. To evaluate the limitations of an approximation used in obtaining the solution for spherical segments known as Geckeler's assumption.

1.3 Structure of Thesis

The thirteen basic differential equations of shells of revolution are formulated in detail in Chapter 2. Chapter 3 presents the two solution techniques to solve these governing shell equations based on standard methods of structural analysis. The formulation of program FLEXSHELL based on the flexibility approach is presented in Chapter 4. An evaluation of the accuracy of the closed form solutions used in this approach is presented in Chapter 5. Finally, Chapter 6 consists of a brief summary and conclusions of the study. Detailed derivations, the program listing, sample input and output files, and the user's manual for program FLEXSHELL are found in the Appendices.

2. THEORY OF SHELLS OF REVOLUTION

2.1 Shell Geometry

As shown in Fig. 2.1, a shell is geometrically defined by its midsurface which bisects the shell thickness, h. A surface of revolution is generated by the rotation of a plane curve about an axis in its plane. This generating curve is called a meridian. Another term frequently used is the parallel circle, which is the intersection of the surface with a plane perpendicular to the axis of revolution. To specify an arbitrary point on the midsurface, two coordinates need be specified: θ, the angular distance of the point from the datum meridian, and ϕ, the angle between a normal to the shell and its axis of revolution. To measure the distance along a normal to the midsurface, a third coordinate z, may be specified. The radii of curvature of a shell of revolution are:

r_0 = radius of the parallel circle;

r_1 = radius of curvature of a meridian;

r_2 = length of the normal between any point on the midsurface and the axis of revolution.

The following relations can be derived from Fig. 2.1.

$$r_0 = r_2 \sin\phi \qquad 2.1(a)$$

$$ds = r_1 d\phi \qquad 2.1(b)$$

$$\therefore \frac{\partial}{\partial s} = \frac{1}{r_1}\frac{\partial}{\partial\phi} \qquad 2.1(c)$$

$$dr = ds \cos\phi \qquad 2.1(d)$$

$$dz = ds \sin\phi \qquad 2.1(e)$$

$$\frac{dr_2}{ds} = \frac{r_1 - r_2\cot\phi}{r_1} \qquad 2.1(f)$$

The internal stress resultants in Fig. 2.2, is determined by integrating the internal stresses through the shell thickness as follows

$$N_\phi = \int_{-h/2}^{h/2} \sigma_\phi (1+z/r_2)dz \qquad 2.2(a)$$

$$N_\theta = \int_{-h/2}^{h/2} \sigma_\theta (1+z/r_1)dz \qquad 2.2(b)$$

$$N_{\phi\theta} = \int_{-h/2}^{h/2} \tau_{\phi\theta} (1+z/r_2)dz \qquad 2.2(c)$$

$$N_{\theta\phi} = \int_{-h/2}^{h/2} \tau_{\theta\phi} (1+z/r_1)dz \qquad 2.2(d)$$

$$Q_\phi = \int_{-h/2}^{h/2} \tau_{\phi z} (1+z/r_2)dz \qquad 2.2(e)$$

$$Q_\theta = \int_{-h/2}^{h/2} \tau_{\theta z} (1+z/r_1)dz \qquad 2.2(f)$$

$$M_\phi = \int_{-h/2}^{h/2} z\sigma_\phi (1+z/r_2)dz \qquad 2.2(g)$$

$$M_\theta = \int_{-h/2}^{h/2} z\sigma_\theta (1+z/r_1)dz \qquad 2.2(h)$$

$$M_{\phi\theta} = \int_{-h/2}^{h/2} z\tau_{\phi\theta} (1+z/r_2)dz \qquad 2.2(i)$$

$$M_{\theta\phi} = \int_{-h/2}^{h/2} z\tau_{\theta\phi} (1+z/r_1)dz \qquad 2.2(j)$$

2.2 The Fundamental Assumptions

The fundamental equations of the general theory of shells of revolution first presented by Flügge (1) are based on the following set of assumptions:

1. Thin shell - the shell thickness is small in comparison to the other dimensions of the shell. Thus, the stresses on the z-face, and the twisting moments about the z-axis may be neglected.
2. Small deflection theory applies. The displacements of the shell due to the applied loads are sufficiently small that the equilibrium equations developed from the initial shell geometry do not change.
3. Material is linearly elastic, i.e., Hooke's law applies.
4. Plane sections remain plane after bending. i.e., the normals to the middle surface before bending remain normal after bending.
5. Deformations of the shell due to radial shears can be neglected.

Now, based on these set of assumptions and the shell geometry, the general theory of shells of revolution may be formulated by:

1. Determining the equilibrium of forces acting on the differential element shown in Fig. 2.2; (six equations with ten unknowns)
2. Establishing the strain-displacement relationships; (six equations with six unknowns)
3. Establishing the stress-strain relationships from

Hooke's Law; (three equations with six unknowns)

4. Transforming the stress-resultant equations into the force-displacement equations; (six equations with three unknowns)

5. Obtaining a complete formulation by combining the force-displacement equations with the equilibrium equations. (thirteen equations with thirteen unknowns)

2.3 Equations of Equilibrium

Consider the differential element shown in Fig. 2.2. From the summation of forces in each of the coordinate directions and moments about each of the coordinate axes, ϕ, θ, and z, the six equations of equilibrium are:

$$(r_0N_\phi)^{\cdot} + r_1(N_{\theta\phi})' - r_1N_\theta\cos\phi - r_0Q_\phi + r_0r_1p_\phi = 0 \qquad 2.3(a)$$

$$(r_0N_{\phi\theta})^{\cdot} + r_1(N_\theta)' + r_1N_{\theta\phi}\cos\phi - r_1Q_\theta\sin\phi + r_0r_1p_\theta = 0 \qquad 2.3(b)$$

$$r_1N_\theta\sin\phi + r_0N_\phi + r_1(Q_\theta)' + (r_0Q_\phi)^{\cdot} - r_0r_1p_z = 0 \qquad 2.3(c)$$

$$(r_0M_\phi)^{\cdot} + r_1(M_{\theta\phi})' - r_1M_\theta\cos\phi - r_0r_1Q_\phi = 0 \qquad 2.3(d)$$

$$(r_0M_{\phi\theta})^{\cdot} + r_1(M_\theta)' + r_1M_{\theta\phi}\cos\phi - r_0r_1Q_\theta = 0 \qquad 2.3(e)$$

$$\frac{M_{\phi\theta} - M_{\theta\phi}}{r_1} = \frac{N_{\phi\theta} - N_{\theta\phi}}{r_2} \qquad 2.3(f)$$

where

$$\frac{\partial(\ )}{\partial\phi} = (\ )^{\cdot}$$

$$\frac{\partial(\ )}{\partial\theta} = (\ )'$$

N_ϕ, N_θ = meridional and circumferential forces respectively;

$N_{\theta\phi}$, $N_{\phi\theta}$ = meridional and circumferential shear forces;

Q_ϕ, Q_θ = transverse shear forces;

M_ϕ, M_θ = meridional and circumferential moments, respectively;

$M_{\theta\phi}$, $M_{\phi\theta}$ = meridional and circumferential twisting moments, respectively.

Note that all forces and moments are expressed in units of force per unit length. The sign convention used is as shown in Fig. 2.2, where N_ϕ and N_θ are positive for tension along the meridian and circumference, respectively. M_ϕ and M_θ are positive when the outer shell surface is in compression.

2.4 Force-Displacement Equations

The deformation of a shell element consists of the change in length of the shell edges, $r_1 d\phi$ and $r_0 d\theta$, and of the change of the angle between these edges. In reference to Fig. 2.3, the midsurface strain-displacement relationships for a shell element are:

Meridional strain, $\epsilon_\phi = \dfrac{1}{r_1}(v^{\cdot} - w)$ 2.4(a)

Hoop strain, $\epsilon_\theta = \dfrac{1}{r_0}(u' + v\cos\phi - w\sin\phi)$ 2.4(b)

Shear strain, $\gamma_{\phi\theta} = \dfrac{v'}{r_0} + \dfrac{u^{\cdot}}{r_1} - \dfrac{u}{r_0}\cos\phi$ 2.4(c)

where

u = midsurface displacement component in the circumferential direction, positive in the direction of increasing θ.

v = midsurface displacement component in the meridional direction, positive in the direction of increasing ϕ.

w = midsurface displacement component in the radial direction, positive in the direction away from the centre of curvature.

Consider a point i at a distance z to the midsurface, i.e., $(r_1)_i = r_1 + z$, and $(r_2)_i = r_2 + z$. From Eqn. 2.1(a), the strains at point i are:

$$(\epsilon_\phi)_i = \frac{(v_i - w_i)}{(r_1 + z)} \quad 2.5(a)$$

$$(\epsilon_\theta)_i = \frac{(u'_i + v_i \cos\phi - w_i \sin\phi)}{(r_2 + z)\sin\phi} \quad 2.5(b)$$

$$(\gamma_{\theta\phi})_i = \frac{v_i'}{(r_2 + z)\sin\phi} + \frac{u_i}{r_1 + z} - \frac{u_i \cos\phi}{(r_2 + z)\sin\phi} \quad 2.5(c)$$

where

$$w_i = w \quad 2.6(a)$$

$$v_i = \frac{v}{r_1}(r_1 + z) - \frac{z\, w'}{r_1} \quad 2.6(b)$$

$$u_i = \frac{u}{r_2}(r_2 + z) - \frac{z\, w'}{r_2} \quad 2.6(c)$$

Hooke's law forms the basis for the formulation of the stress-strain equations.

$$\sigma_\phi = \frac{E}{(1-\nu^2)}(\epsilon_\phi + \nu \epsilon_\theta) \quad 2.7(a)$$

$$\sigma_\theta = \frac{E}{(1-\nu^2)}(\epsilon_\theta + \nu \epsilon_\phi) \quad 2.7(b)$$

$$\tau_{\theta\phi} = \frac{E}{2(1+\nu)} \gamma_{\theta\phi} \quad 2.7(c)$$

where E is the modulus of elasticity and ν is Poisson's ratio. Combining the strain-displacement relationships (Eqns. 2.5 and 2.6) and substituting these into the stress-strain equations (Eqns. 2.7), and finally, substituting these into the stress-resultant equations

(Eqns. 2.2) and integrating through the shell thickness, the force-displacement relationships are as follows:

$$N_\phi = K\left[\frac{\dot{v}+w}{r_1} + \frac{\nu(u'+v\cos\phi+w\sin\phi)}{r_0}\right]$$

$$+ \frac{D}{r_1^2}\,\frac{r_2-r_1}{r_2}\left[\frac{v-\dot{w}}{r_1}\,\frac{\dot{r}_1}{r_1} + \frac{\ddot{w}+w}{r_1}\right] \qquad 2.8(a)$$

$$N_\theta = K\left[\frac{u'+v\cos\phi+w\sin\phi}{r_0} + \frac{\nu(\dot{v}+w)}{r_1}\right]$$

$$- \frac{D}{r_0 r_1}\,\frac{r_2-r_1}{r_2}\left[-\frac{v}{r_1}\,\frac{r_2-r_1}{r_2}\cos\phi + \frac{w\sin\phi}{r_2} + \frac{w''}{r_0} + \frac{\dot{w}\cos\phi}{r_1}\right] \qquad 2.8(b)$$

$$N_{\phi\theta} = \frac{K(1-\nu)}{2}\left[\frac{\dot{u}}{r_1} + \frac{v'-u\cos\phi}{r_0}\right] + \frac{D}{r_1^2}\,\frac{1-\nu}{2}\,\frac{r_2-r_1}{r_2}\left[\frac{\dot{u}}{r_1}\,\frac{r_2-r_1}{r_2}\right.$$

$$\left. + \frac{u}{r_2}\,\frac{r_1-r_2}{r_2}\cot\phi + \frac{\dot{w}'}{r_0} - \frac{w'}{r_0}\,\frac{\dot{r}_1\cos\phi}{r_0}\right] \qquad 2.8(c)$$

$$N_{\theta\phi} = \frac{K(1-\nu)}{2}\left[\frac{\dot{u}}{r_1} + \frac{v'-u\cos\phi}{r_0}\right]$$

$$+ \frac{D}{r_0 r_1}\,\frac{1-\nu}{2}\,\frac{r_2-r_1}{r_2}\left[\frac{v'}{r_1}\,\frac{r_2-r_1}{r_2} - \frac{\dot{w}'}{r_1} + \frac{w'\cos\phi}{r_0}\right] \qquad 2.8(d)$$

$$M_\phi = D\left[\frac{1}{r_1^2}\left(\ddot{w} - \dot{w}\,\frac{\dot{r}_1}{r_1} - w\Big(\frac{r_1-r_2}{r_2}\Big)\right) - \frac{\dot{v}}{r_1 r_2} + \frac{v}{r_1^2}\,\frac{\dot{r}_1}{r_1}\right.$$

$$\left. + \frac{\nu w''}{r_0^2} + \frac{\nu \dot{w}\cos\phi}{r_0 r_1} - \frac{\nu u'}{r_0 r_1} - \frac{\nu v\cos\phi}{r_0 r_1}\right] \qquad 2.8(e)$$

$$M_\theta = D\left[\frac{w''}{r_0^2} + \frac{\dot{w}\cos\phi}{r_0 r_1} - \frac{w}{r_2^2}\,\frac{r_2-r_1}{r_1} - \frac{u'}{r_0 r_1} - \frac{v\cos\phi}{r_0 r_1}\,\frac{2r_2-r_1}{r_2}\right.$$

$$\left. + \frac{\nu}{r_1^2}\left(\ddot{w} - \dot{w}\,\frac{\dot{r}_1}{r_1}\right) - \frac{\nu\dot{v}}{r_1^2} + \frac{\nu v\dot{r}_1}{r_1^3}\right] \qquad 2.8(f)$$

$$M_{\phi\theta} = \frac{D(1-\nu)}{2}\left[\frac{2w''}{r_0 r_1} - \frac{2w'\cos\phi}{r_2} - \frac{u'}{r_1 r_2}\,\frac{2r_1-r_2}{r_1} + \frac{u}{r_2^2}\,\frac{(2r_1-r_2)\cot\phi}{r_1} - \frac{v'}{r_0 r_1}\right] \quad 2.8(g)$$

$$M_{\theta\phi} = \frac{D(1-\nu)}{2}\left[\frac{2w''}{r_0 r_1} - \frac{2w'\cos\phi}{r_0^2} - \frac{u'}{r_1 r_2} + \frac{u}{r_2^2}\cot\phi - \frac{w'}{r_0 r_1}\,\frac{(2r_2-r_1)}{r_2}\right] \quad 2.8(h)$$

Where the extensional rigidity K and the flexural rigidity D, are defined respectively as

$$K = \frac{Eh}{(1-\nu^2)}$$

$$D = \frac{Eh^3}{(1-\nu^2)}$$

There are now fourteen equations (Eqns. 2.3 and 2.8) with thirteen unknowns, N_ϕ, N_θ, $N_{\theta\phi}$, $N_{\phi\theta}$, M_ϕ, M_θ, $M_{\phi\theta}$, $M_{\theta\phi}$, Q_ϕ, Q_θ, u, v, w. Note that there is one equation too many. Since both sides of Eqn. 2.3(f) are small differences between small quantities which are almost equal, this equation may be discarded. Thus, there is now a balance of unknowns and equations. The classical method of solution would be to reduce these differential equations into a single eighth order equation in terms of one variable. This procedure tends to be too complicated and cumbersome to solve. Therefore, alternative solutions to these equations based on the standard methods of structural analysis will be presented in the following chapter.

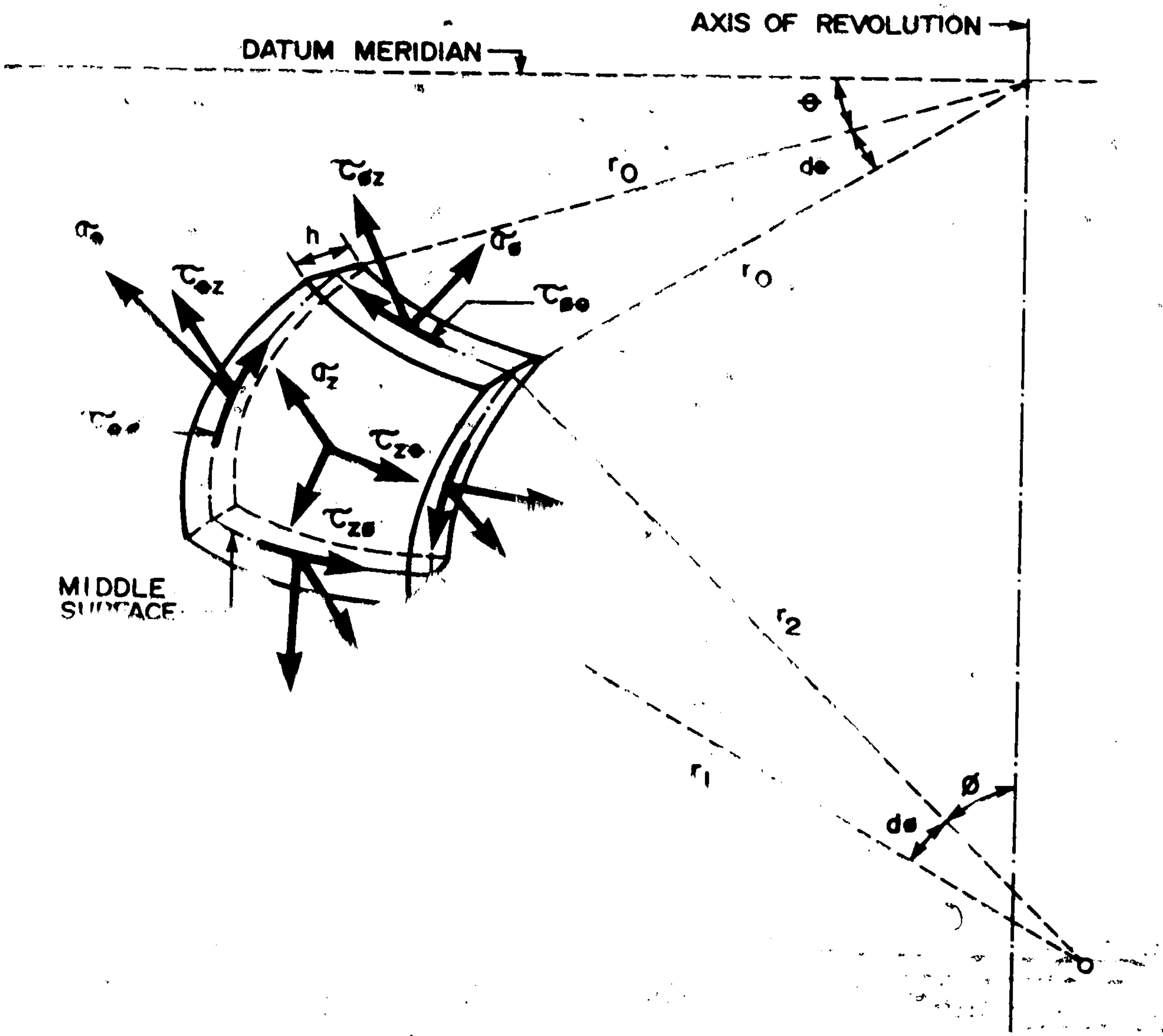


Figure 2.1 GEOMETRY OF SHELL

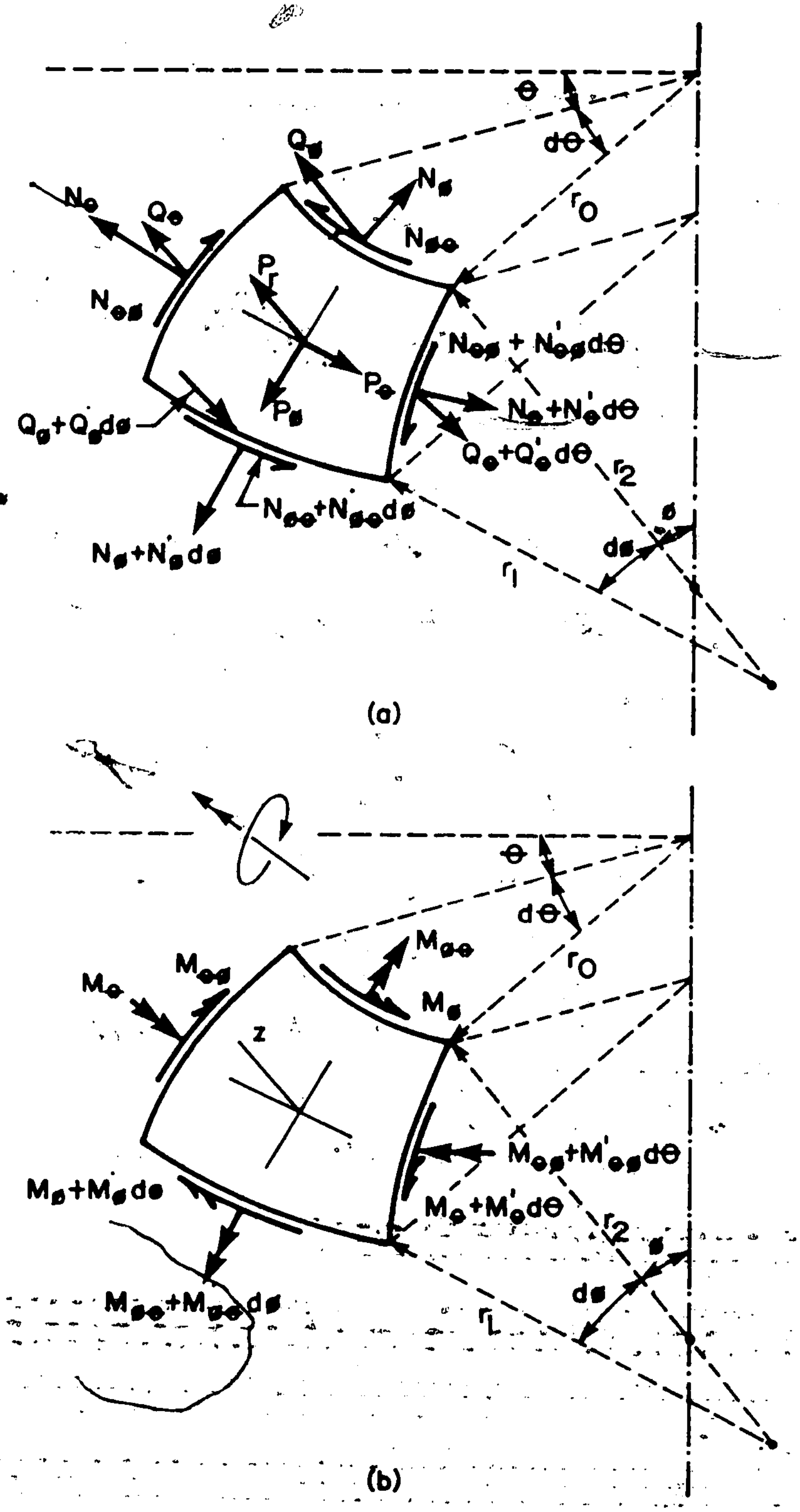


Figure 2.2 FORCES ACTING ON
SHELL MIDSURFACE

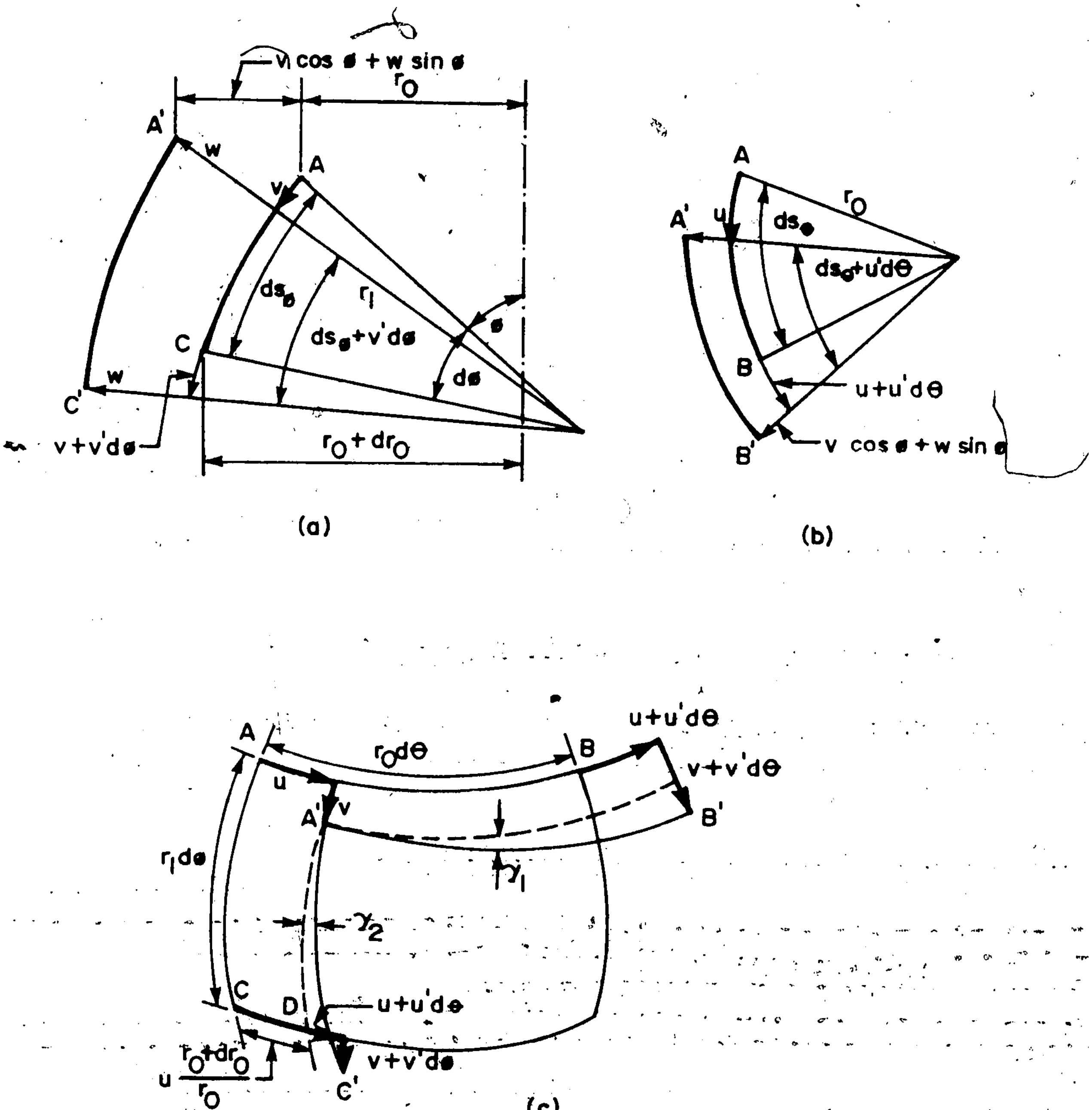


Figure 2.3 SHELL SEGMENTS BEFORE & AFTER DEFORMATION
(a) meridian
(b) parallel circle
(c) angle change

3. METHOD OF ANALYSIS

Structures that geometrically consist of several segments of shells of revolution can be analyzed by either of the two standard methods of structural analysis, namely: the stiffness method and the flexibility method. With the stiffness method, the stiffness matrix relating the forces and deformations at the edge of each shell segment are computed using the procedures outlined in Section 3.1. These element stiffness matrices are then superposed to form the structural stiffness matrix from which the segment edge deformations are computed. With the flexibility approach (Section 3.2), the flexibility influence coefficients for each shell segment are obtained. Equations of geometric compatibility at the segment boundaries are written to obtain the forces at segment junctions.

In this study, the use of the flexibility approach is explained in detail for structures with axisymmetric loading. The membrane solutions are used for the particular solution.

3.1 The Stiffness Approach

Program SASHELL analyzes a segmented shell structure based on the stiffness method. To establish the stiffness matrix and fixed end forces vector the basic shell equations must be solved numerically. But in order to do this, the shell equations must be reduced to a set of eight first order differential equations, corresponding to the eight natural boundary conditions of the shell segment by:

1. Expanding the equations using a Fourier series to eliminate the necessity of forming the equilibrium equations in the circumferential direction.
2. Introducing the auxilliary equations to eliminate the in-plane shear force in the circumferential direction and the meridional twisting moment.
3. Performing matrix operations to eliminate the forces in the circumferential direction.

Introducing the s coordinate, which measures the distance along the shell meridian, the five independent equations of equilibrium (Eqns. 2.3) become

$$r_1(r_0N_s)^\circ + r_1N_{\theta s}' - r_1N_\theta\cos\phi - r_0Q_s + r_0r_1p_s = 0 \qquad 3.1(a)$$

$$r_1(r_0N_{s\theta})^\circ + r_1N_\theta' + r_1N_{\theta s}\cos\phi - r_1Q_\theta\sin\phi + r_0r_1p_\theta = 0 \qquad 3.1(b)$$

$$r_1N_\theta\sin\phi + r_0N_s + r_1Q_\theta' + r_1(r_0Q_s)^\circ - r_0r_1p_z = 0 \qquad 3.1(c)$$

$$r_1(r_0M_s)^\circ + r_1M_{\theta s}' - r_1M_\theta\cos\phi - r_0r_1Q_s = 0 \qquad 3.1(d)$$

$$r_1(r_0M_{s\theta})^\circ + r_1M_\theta' + r_1M_{\theta s}\cos\phi - r_0r_1Q_\theta = 0 \qquad 3.1(e)$$

where

$$\frac{1}{r_1}\frac{\partial(\ )}{\partial\phi} = \frac{\partial(\ )}{\partial s} = (\ )^\circ \qquad 3.1(f)$$

The force-displacement equations (Eqns. 2.8) in terms of the s-coordinate are

$$N_s = K\left[v^\circ + \frac{w}{r_1} + \frac{\nu(u' + v\cos\phi + w\sin\phi)}{r_0}\right]$$

$$+ \frac{D}{r_1^2}\,\frac{r_2 - r_1}{r_2}\left[v - \frac{w^\circ}{r_1}\,r_1{}^\circ + w^\circ + \frac{w}{r_1}\right] \qquad 3.2(a)$$

$$N_\theta = K\left[\frac{u'+v\cos\phi+w\sin\phi}{r_0} + \nu v^\circ + \frac{w}{r_1}\right]$$

$$- \frac{D}{r_0 r_1}\frac{r_2-r_1}{r_2}\left[-\frac{v}{r_1}\frac{r_2-r_1}{r_2}\cos\phi + \frac{w\sin\phi}{r_2} + \frac{w''}{r_0} + w^\circ\cos\phi\right] \qquad 3.2(b)$$

$$N_{s\theta} = \frac{K(1-\nu)}{2}\left[u^\circ + \frac{v'-u\cos\phi}{r_0}\right] + \frac{D}{r_1^2}\frac{1-\nu}{2}\frac{r_2-r_1}{r_2}\left[u^\circ\frac{r_2-r_1}{r_2}\right.$$

$$\left. + \frac{u}{r_2}\frac{r_1-r_2}{r_2}\cot\phi + \frac{r_1 w'^\circ}{r_0} - \frac{w'}{r_0}\frac{r_1\cos\phi}{r_0}\right] \qquad 3.2(c)$$

$$N_{\theta s} = \frac{K(1-\nu)}{2}\left[u^\circ + \frac{v'-u\cos\phi}{r_0}\right]$$

$$+ \frac{D}{r_0 r_1}\frac{1-\nu}{2}\frac{r_2-r_1}{r_2}\left[\frac{v'}{r_1}\frac{r_2-r_1}{r_2} - w'^\circ + \frac{w'\cos\phi}{r_0}\right] \qquad 3.2(d)$$

$$M_s = D\left[w^{\circ\circ} - w^\circ\frac{r_1^\circ}{r_1} - w\frac{(r_1-r_2)}{r_2 r_1^2} - \frac{v^\circ}{r_2} + \frac{v}{r_1^2} r_1\right.$$

$$\left. + \frac{\nu w''}{r_0^2} + \frac{\nu w^\circ\cos\phi}{r_0} - \frac{\nu u'}{r_0 r_1} - \frac{\nu v\cos\phi}{r_0 r_1}\right] \qquad 3.2(e)$$

$$M_\theta = D\left[\frac{w''}{r_0^2} + \frac{w^\circ\cos\phi}{r_0} - \frac{w}{r_2^2}\frac{r_2-r_1}{r_1} - \frac{u'}{r_0 r_1} - \frac{v\cos\phi}{r_0 r_1}\frac{2r_2-r_1}{r_2}\right.$$

$$\left. + \nu w^{\circ\circ} - \nu w^\circ\frac{r_1^\circ}{r_1} - \frac{\nu v^\circ}{r_1} + \frac{\nu v r_1^\circ}{r_1^2}\right] \qquad 3.2(f)$$

$$M_{s\theta} = \frac{D(1-\nu)}{2}\left[\frac{2w'^\circ}{r_0} - \frac{2w'\cos\phi}{r_2} - \frac{u^\circ}{r_2}\frac{2r_1-r_2}{r_1}\right.$$

$$\left. + \frac{u}{r_2^2}\frac{(2r_1-r_2)\cot\phi}{r_1} - \frac{v'}{r_0 r_1}\right] \qquad 3.2(g)$$

$$M_{\theta s} = \frac{D(1-\nu)}{2}\left[\frac{2w'^\circ}{r_0} - \frac{2w'\cos\phi}{r_0^2} - \frac{u^\circ}{r_2}\right.$$

$$+ \frac{u}{r_2^2}\cot\phi - \frac{v'}{r_0 r_1} \frac{(2r_2 - r_1)}{r_2}\Big] \qquad 3.2(h)$$

3.1.1 Fourier Series

For any variable, say F(x,y) being arbitrary functions of x and y, may be represented in the form

$$F = \sum_{n=0}^{\infty} F_n(x)\cos(ny) + \sum_{n=1}^{\infty} F_n(x)\sin(ny) \qquad 3.3$$

where n is the harmonic number and variable F_n is now a function of x only. Similarly, the load components p_s, p_θ, and p_z, and forces N_s, N_θ, $N_{s\theta}$, $N_{\theta s}$, M_s, M_θ, $M_{\theta s}$, $M_{s\theta}$, Q_s, and Q_θ, and displacement components u, v, and w, may be expressed as a Fourier series, where the variable components become a function of s only. The first and second series in each expression represent the portions of the variables which are respectively symmetric and anti-symmetric with respect to the meridian passing through the line $\theta = 0$.

$$p_s = \sum_{n=0}^{\infty} p_{sn}(s)\cos(n\theta) + \sum_{n=1}^{\infty} p_{sn}(s)\sin(n\theta) \qquad 3.4(a)$$

$$p_\theta = \sum_{n=0}^{\infty} p_{\theta n}(s)\cos(n\theta) + \sum_{n=1}^{\infty} p_{\theta n}(s)\sin(n\theta) \qquad 3.4(b)$$

$$p_z = \sum_{n=0}^{\infty} p_{zn}(s)\cos(n\theta) + \sum_{n=1}^{\infty} p_{zn}(s)\sin(n\theta) \qquad 3.4(c)$$

$$N_s = \sum_{n=0}^{\infty} N_{sn}(s)\cos(n\theta) + \sum_{n=1}^{\infty} N_{sn}(s)\sin(n\theta) \qquad 3.4(e)$$

$$N_\theta = \sum_{n=0}^{\infty} N_{\theta n}(s)\cos(n\theta) + \sum_{n=1}^{\infty} N_{\theta n}(s)\sin(n\theta) \qquad 3.4(f)$$

$$N_{s\theta} = \sum_{n=0}^{\infty} N_{s\theta n}(s)\cos(n\theta) + \sum_{n=1}^{\infty} N_{s\theta n}(s)\sin(n\theta) \qquad 3.4(g)$$

$$N_{\theta s} = \sum_{n=0}^{\infty} N_{\theta s n}(s)\cos(n\theta) + \sum_{n=1}^{\infty} N_{\theta s n}(s)\sin(n\theta) \qquad 3.4(h)$$

$$Q_s = \sum_{n=0}^{\infty} Q_{sn}(s)\cos(n\theta) + \sum_{n=1}^{\infty} Q_{sn}(s)\sin(n\theta) \qquad 3.4(i)$$

$$Q_\theta = \sum_{n=0}^{\infty} Q_{\theta n}(s)\cos(n\theta) + \sum_{n=1}^{\infty} Q_{\theta n}(s)\sin(n\theta) \qquad 3.4(j)$$

$$M_s = \sum_{n=0}^{\infty} M_{sn}(s)\cos(n\theta) + \sum_{n=1}^{\infty} M_{sn}(s)\sin(n\theta) \qquad 3.4(k)$$

$$M_\theta = \sum_{n=0}^{\infty} M_{\theta n}(s)\cos(n\theta) + \sum_{n=1}^{\infty} M_{\theta n}(s)\sin(n\theta) \qquad 3.4(l)$$

$$M_{s\theta} = \sum_{n=0}^{\infty} M_{s\theta n}(s)\cos(n\theta) + \sum_{n=1}^{\infty} M_{s\theta n}(s)\sin(n\theta) \qquad 3.4(m)$$

$$M_{\theta s} = \sum_{n=0}^{\infty} M_{\theta s n}(s)\cos(n\theta) + \sum_{n=1}^{\infty} M_{\theta s n}(s)\sin(n\theta) \qquad 3.4(n)$$

$$u = \sum_{n=0}^{\infty} u_n(s)\cos(n\theta) + \sum_{n=1}^{\infty} u_n(s)\sin(n\theta) \qquad 3.4(o)$$

$$v = \sum_{n=0}^{\infty} v_n(s)\cos(n\theta) + \sum_{n=1}^{\infty} v_n(s)\sin(n\theta) \qquad 3.4(p)$$

$$w = \sum_{n=0}^{\infty} w_n(s)\cos(n\theta) + \sum_{n=1}^{\infty} w_n(s)\sin(n\theta) \qquad 3.4(q)$$

For an arbitrary applied load expressed as a Fourier series of order N, there are 2N+1 terms that represent each component of the load; (n = 0,1,2,...,N) for the symmetric series and (n = 1,2,3,...,N) for the anti-symmetric series. For each value of n, the s-dependent variables with subscript n (Eqn. 3.4) can be substituted into the basic shell equations (Eqns. 3.1 and 3.2), because the sequences $\sin(n\theta)$ and $\cos(n\theta)$ are linearly independent.

Differentiations with respect to θ can be performed and the terms grouped according to the common factors, $\cos(n\theta)$ and $\sin(n\theta)$. Since the coefficient of each of these factors must be zero, each factor produces a separate equation. For example, for any n, the cosine terms in Eqn. 3.1(a) become

$$r_0 N_{sn}^{\circ}\cos(n\theta) + \cos\phi N_{sn}\cos(n\theta) + nN_{\theta sn}\cos(n\theta) - \cos\phi N_{\theta n}\cos(n\theta) - \frac{r_0}{r_1}Q_{sn}\cos(n\theta) + R_0 p_{sn}\cos(n\theta) = 0 \qquad 3.5$$

which, upon factoring out the common term, yields

$$r_0 N_{sn}^{\circ} + \cos\phi N_{sn} + nN_{\theta sn} - \cos\phi N_{\theta n} - \frac{r_0}{r_1}Q_{sn} + r_0 p_{sn} = 0 \qquad 3.6$$

Similarly, for the sine terms, Eqn. 3.1(a) become

$$r_0 N_{sn}^{\circ} + \cos\phi N_{sn} - nN_{\theta sn} - \cos\phi N_{\theta n} - \frac{r_0}{r_1}Q_{sn} + r_0 p_{sn} = 0 \qquad 3.7$$

Let R_0, R_1, R_2 be defined as shell curvature, i.e.

$$R_0 = \frac{1}{r_0}$$

$$R_1 = \frac{1}{r_1}$$

$$R_2 = \frac{1}{r_2}$$

Thus, for the nth set of equations, the five independent equilibrium equations derived from Eqns. 3.1 become

$$N_{sn}^{\circ} + R_0\cos\phi N_{sn} \pm nR_0 N_{\theta sn} - R_0\cos\phi N_{\theta n} - R_1 Q_{sn} + p_{sn} = 0 \qquad 3.8(a)$$

$$N_{s\theta n}^{\circ} + R_0\cos\phi N_{s\theta n} \mp nR_0 N_{\theta n} + R_0\cos\phi N_{\theta sn} - R_2 Q_{\theta n} + p_{\theta n} = 0 \qquad 3.8(b)$$

$$R_2 N_{\theta n} + R_1 N_{sn} \pm nR_0 Q_{\theta n} + Q_{sn}^{\circ} + R_0\cos\phi Q_{sn} - p_{zn} = 0 \qquad 3.8(c)$$

$$M_{s,n}^{\circ} + R_0\cos\phi M_{s,n} \pm nR_0M_{\theta s,n} - R_0\cos\phi M_{\theta,n} - Q_{s,n} = 0 \qquad 3.8(d)$$

$$M_{s\theta,n}^{\circ} + R_0\cos\phi M_{s\theta,n} \mp nR_0M_{\theta,n} + R_0\cos\phi M_{\theta s,n} - Q_{\theta,n} = 0 \qquad 3.8(e)$$

and the eight force-displacement equations obtained from Eqn. 3.2 become

$$N_{s,n} = [DR_1(R_1-R_2)r_1^{\circ}]\beta_n - [D(R_1-R_2)]\beta_n^{\circ} + [K(R_1+\nu R_2) + DR_1^2(R_1-R_2)]w_n + [\nu KR_0\cos\phi - DR_1^2(R_1-R_2)r_1^{\circ}]v_n + [K + DR_1(R_1-R_2)]v_n^{\circ} \pm [\nu nKR_0]u_n \qquad 3.9(a)$$

$$N_{\theta,n} = [DR_0(R_1-R_2)\cos\phi]\beta_n + [K(R_2+\nu R_1) + D(R_1-R_2)(R_0^2n^2-R_2^2)]w_n + [KR_0\cos\phi - DR_0R_2\cos\phi(R_1-R_2)]v_n + [\nu K]v_n^{\circ} \pm [nKR_0]u_n \qquad 3.9(b)$$

$$N_{s\theta,n} = 0.5(1-\nu)\{\pm[nDR_0(R_1-R_2)]\beta_n \pm [nDR_0^2\cos\phi(R_1-R_2)]w_n \mp [nKR_0 + nDR_0R_1(R_1-R_2)]v_n - [KR_0\cos\phi - DR_0\cos\phi(R_1-R_2)^2]u_n + [K + D(R_1-R_2)^2]u_n^{\circ}\} \qquad 3.9(c)$$

$$N_{\theta s,n} = 0.5(1-\nu)\{\pm\{nDR_0(R_1-R_2)]\beta_n \mp [nDR_0^2\cos(\&v{>}(R_1-R_2)]w_n \mp [nKR_0 + nDR_0R_1(R_1-R_2)]v_n - [KR_0\cos\phi]u_n + [K]u_n^{\circ} \qquad 3.9(d)$$

$$M_{s,n} = [DR_1r_1^{\circ} - \nu DR_0\cos\phi]\beta_n - [D]\beta_n^{\circ} + [DR_1(R_1-R_2) - \nu DR_0^2n^2]w_n - [DR_1^2r_1^{\circ}]v_n + [D(R_1-R_2)]v_n^{\circ} \mp [\nu nDR_0R_2]u_n \qquad 3.9(e)$$

$$M_{\theta,n} = [\nu DR_1r_1^{\circ} - DR_0\cos\phi]\beta_n - [\nu D]\beta_n^{\circ} + [Dn^2R_0^2 + DR_2(R_1-R_2)]w_n - [\nu DR_1r_1^{\circ} + DR_0\cos\phi(R_1-R_2)]v_n \mp [nDR_0R_2]u_n \qquad 3.9(f)$$

$$M_{s\theta,n} = 0.5(1-\nu)\{\pm[2nDR_0]\beta_n \pm [2nDR_0^2\cos\phi]w_n \mp [nDR_0R_1]v_n - [DR_0\cos\phi(R_1-2R_2)]u_n + [D(R_1-2R_2)]u_n^{\circ}\} \qquad 3.9(g)$$

$$M_{\theta s,n} = 0.5(1-\nu)\{\pm[2nDR_0]\beta_n \pm [2nDR_0^2\cos\phi]w_n \mp [nDR_0R_1]v_n + [DR_0R_2\cos\phi]u_n - [DR_2]u_n^{\circ}\}$$

where β_n and β_n° is an auxilliary varible which will be defined in the following section. Note that there are two sets of equations, grouped according to the cosine and sine terms. The final solution is obtained by solving each set separately and superimposing the two solutions.

3.1.2 Auxiliary Equations

The quantities in the natural boundary conditions on the edges of a shell segment are the four displacement components, the rotation of the meridian (β), the radial displacement (w), the meridional displacement (v), and the circumferential displacement (u), and the four corresponding forces, the meridional moment (M_s), the effective transverse shear force (S_s), the normal in-plane meridional force (N_s), and the effective tangential shear force (T_s). Three of these variables β, T_s, and S_s do not appear in the basic shell equations. They may be introduced by setting up the so-called auxilliary equations which express these variables in terms of in-plane shear forces in the circumferential direction and the meridional twisting moment.

Consider the side view of the top edge of the shell element shown on Fig. 3.1 with two adjacent elements of length $ds = r_0 d\theta$. (Note that $ds = r_2 d\theta$ for very small ds) The moments acting on the infinitesimal element ds can be replaced by a set of statically equivalent forces F_n and F_t (5), such that

$$F_n = M_{s\theta}$$

$$F_t = F_n d\theta$$

From the figure, superimposing these forces with the transverse force Q_s, and the in-plane shear force, $N_{s\theta}$, respectively, yields and expression for the Kirchhoff shears, S_s and T_s

$$S_s = Q_s + R_0 M'_{s\theta}$$

$$T_s = N_{s\theta} - R_2 M_{s\theta}$$

Expanding these into a Fourier series yield

$$S_{sn} = Q_{sn} \pm nR_0 M_{s\theta n} \qquad 3.10$$

$$T_{sn} = N_{s\theta n} - R_2 M_{s\theta n} \qquad 3.11$$

Using the geometrical relations in Eqns. 2.1 and 3.1(f), the derivatives of these forces with respect to the coordinate s may be written as

$$S_{sn}^{\circ} = Q_{sn}^{\circ} \pm nR_0 M_{s\theta n}^{\circ} \mp nR_0^2 \cos\phi M_{s\theta n} \qquad 3.12$$

$$T_{sn}^{\circ} = N_{s\theta n}^{\circ} - R_2 M_{s\theta n}^{\circ} - R_2(R_1 - R_2)\cot\phi M_{s\theta n} \qquad 3.13$$

Also, by superimposing Figs. 3.2(a) and (b), the angle by which an element of the meridian rotates during deformation may be expressed in terms of the displacement components as follows,

$$\beta_n = -w^{\circ} + R_1 v \qquad 3.14$$

3.1.3 Reduction of the Shell Equations

Rewriting Eqn. 3.10 to form an expression for Q_{sn}, and substituting this into Eqns. 3.8(a) and (d) respectively, yields

$$N_{sn}^{\circ} = R_1 S_{sn} - R_0 \cos\phi N_{sn} \mp nR_0 R_1 M_{s\theta n} + R_0 \cos\phi N_{\theta n} \mp nR_0 N_{\theta sn} - p_{sn} \qquad 3.15$$

$$M_{sn}^{\circ} = S_{sn} - R_0 \cos\phi M_{sn} + R_0 \cos\phi M_{\theta n} \mp nR_0 (M_{\theta sn} + M_{s\theta n}) \qquad 3.16$$

Rewriting Eqns. 3.11, 3.13, and 3.8(e) to form expressions for $N_{s\theta n}$, $N_{s\theta n}^{\circ}$, and $Q_{\theta n}$, yields

$$N_{s\theta n} = T_{sn} + R_2 M_{s\theta n}$$

$$N_{s\theta n}^{\circ} = T_{sn}^{\circ} + R_2 M_{s\theta n}^{\circ} + R_2(R_1 - R_2)\cot\phi M_{s\theta n}$$

$$Q_{\theta n} = M_{s\theta n} + R_0\cos\phi M_{s\theta n} \mp nR_0 M_{\theta n} + R_0\cos\phi M_{\theta s n}$$

Substituting the above expressions into Eqn. 3.8(b), and using the relation, $R_0\sin\phi = R_2$, yields

$$T_{sn}^{\circ} = -R_0\cos\phi(R_1 - R_2)M_{s\theta n} - R_0\cos\phi T_{sn} \pm nR_0 N_{\theta n} - R_0\cos\phi N_{\theta s n} \mp nR_0R_2M_{\theta n} + R_0R_2\cos\phi M_{\theta s n} - p_{\theta n} \qquad 3.17$$

Finally, rewriting Eqn. 3.12 to form an expression for Q_{sn}°, and substituting this, in addition to the expressions for $Q_{\theta n}$ and Q_{sn} derived earlier, into Eqn. 3.8(a), gives

$$S_{sn}^{\circ} = -R_2N_{\theta n} - R_1N_{sn} + n^2R_0^2M_{\theta n} \mp nR_0^2\cos\phi(M_{\theta s n} + M_{s\theta n}) - R_0\cos\phi S_{sn} + p_{zn} \qquad 3.18$$

Eqns. 3.15 to 3.18 may be written symbolically as

$$M_{sn}^{\circ} = F_{20}(M_{sn}, S_{sn}, M_{\theta n}, M_{\theta s n}, M_{s\theta n})$$

$$S_{sn}^{\circ} = F_{21}(S_{sn}, N_{sn}, M_{\theta n}, M_{\theta s n}, M_{s\theta n}, N_{\theta n}, p_{zn})$$

$$N_{sn}^{\circ} = F_{22}(S_{sn}, N_{sn}, M_{s\theta n}, N_{\theta n}, N_{\theta s n}, p_{sn})$$

$$T_{sn}^{\circ} = F_{23}(T_{sn}, M_{\theta n}, M_{\theta s n}, N_{\theta n}, N_{\theta s n}, p_{\theta n})$$

or, in matrix form,

$$\{F_s^{\circ}\} = [B_1 \; B_2]\begin{Bmatrix} F_s \\ F_\theta \end{Bmatrix} + \{B_3\} \qquad 3.19$$

where

$$\langle F_s \rangle = \langle M_{s\theta n} \; S_{sn} \; N_{sn} \; T_{sn} \rangle$$

$$\langle F_s^{\circ} \rangle = \langle M_{s\theta n}^{\circ} \; S_{sn}^{\circ} \; N_{sn}^{\circ} \; T_{sn}^{\circ} \rangle$$

$$\langle F_\theta \rangle = \langle M_{\theta n} \; M_{\theta s n} \; M_{s\theta n} \; N_{\theta n} \; N_{\theta s n} \rangle$$

and the coefficients of $[B_1 \; B_2]$ is a function of the geometric and material properties of the shell; and $\{B_3\}$ is the load vector. These matrices are defined in Table 3.1. The plus and minus signs relate to the two sets of equations, grouped according to the cosine and sine terms in the Fourier series expansion.

To form expressions for the displacement variables, manipulate the force-displacement equations as follows

Let

$$CA_1 = K + DR_1(R_1-R_2) \quad 3.20(a)$$

$$CA_2 = K + DR_2(R_1-R_2) \quad 3.20(b)$$

Multiply Eqn. 3.9(a) by (R_1-R_2),

$$N_{sn}(R_1-R_2) = [DR_1(R_1-R_2)^2r_0{}^2]\beta_n - [D(R_1-R_2)^2]\beta_n{}^\circ + [K(R_1+\nu R_2) + DR_1^2(R_1-R_2)](R_1-R_2)w_n + [\nu DR_0\cos\phi - DR_1^2(R_1-R_2)](R_1-R_2)v_n + [CA_1(R_1-R_2)]v_n{}'' \pm [\nu KnR_0(R_1-R_2)]u_n$$

and multiply Eqn. 3.9(e) by CA_1/D,

$$CA_1M_{sn}/D = [R_1r_1{}^\circ - \nu R_0\cos\phi]CA_1\beta_n - CA_1\beta_n{}^\circ + [R_1(R_1-R_2) - \nu n^2R_0^2]CA_1w_n - CA_1R_1^2r_1{}^\circ v_n + [CA_1(R_1-R_2)]v_n{}^\circ \mp [\nu nR_0R_2CA_1]u_n$$

Subtracting the first from the second expression, and simplifying by means of Eqns. 3.20 yields,

$$\beta_n{}^\circ = \{-CA_1M_{sn}/D + (R_1-R_2)N_{sn} + [R_1r_1{}^\circ CA_2 - \nu R_0\cos\phi CA_1]\beta_n - [CA_1\nu n^2R_0^2 + \nu KR_2(R_1-R_2)]w_n - [\nu KR_0\cos\phi(R_1-R_2) + R_1^2r_1{}^\circ CA_2]v_n \pm [\nu nR_0R_1CA_2]u_n\}/CA_2 \quad 3.21$$

Similarly, subtracting Eqn 3.9(a) from the product of (R_1-R_2) and Eqn. 3.9(e), and simplify the expression using Eqns. 3.20 yields

$$v_n{}^\circ = \{-(R_1-R_2)M_{sn} + N_{sn} - [\nu DR_0\cos\phi(R_1-R_2)]\beta_n -$$

$$[\nu D n^2 R_0^2(R_1-R_2) + R_1 CA_2 + \nu K R_2]w_n - [\nu K R_0 \cos\phi]v_n \mp$$

$$[\nu n R_0 CA_2]u_n\}/CA_2 \qquad 3.22$$

Rewriting Eqn. 3.14 yields

$$w_n{}^\circ = v_n R_1 - \beta_n \qquad 3.23$$

Finally, substituting Eqn. 3.9(g) into 3.11, and rewriting the equation to form an expression for $N_{s\theta n}$, then substituting this into Eqn. 3.9(c), and simplifying,

$$u_n{}^\circ = \{2T_{s,n}/(1-\nu) \mp [DnR_0(R_1-3R_2)]\beta_n \mp [DnR_0^2\cos\phi(R_1-3R_2)]w_n$$

$$\pm [nCA_1R_0 - DnR_0R_1R_2]v_n + [R_0\cos\phi CA_3]u_n\}/CA_3 \qquad 3.24$$

where

$$CA_3 = K + D(R_1^2-3R_1R_2+3R_2^2) \qquad 3.25$$

Eqns. 3.21 to 3.25 may be written symbolically as

$$\beta_n{}^\circ = F_{24}(\beta_n, w_n, v_n, M_{s,n}, N_{s,n})$$

$$w_n{}^\circ = F_{25}(\beta_n, v_n)$$

$$v_n{}^\circ = F_{26}(\beta_n, w_n, v_n, u_n, M_{s,n}, N_{s,n})$$

$$u_n{}^\circ = F_{27}(\beta_n, w_n, v_n, u_n, T_{s,n})$$

or, in matrix form,

$$\{D^\circ\} = [A_1 \ \ A_2]\begin{Bmatrix} D \\ F_s \end{Bmatrix} \qquad 3.26$$

where {D} and {D°} consists of displacements β_n, w_n, u_n, and v_n, and their derivatives with respect to the coordinate s, respectively; $[A_1 \ A_2]$ is a function of the geometric and material properties of the shell, defined in Table 3.2. Again, the plus-minus signs relate to the set of equations, grouped according to the sine and cosine terms in the Fourier series.

The vector <F_o> which appears in Eqn. 3.19, is formed by writing the force-displacement equations in the following order, 3.9(f), (h), (g), (b), (d). In matrix form,

$$\{F_6\} = [B_4 \; B_5]\begin{Bmatrix} D^\circ \\ D \end{Bmatrix} \qquad 3.27$$

where {D} and {D°} are defined as before. The cofficients of $[B_4 \; B_5]$ is a function of the geometric and material properties of the shell, defined in Tables 3.3.

Substituting Eqn. 3.27 into 3.19 yields

$$\{F_s^\circ\} = [B_1]\{F_s\} + [B_2]([B_4]\{D^\circ\} + [B_5]\{D\}) + \{B_3\}$$

Simplifying,

$$\{F_s^\circ\} = [A_3]\{D\} + [A_4]\{F_s\} + \{B_3\} \qquad 3.28$$

where

$$[A_3] = [B_2][B_4][A_1] + [B_2][B_5]$$

$$[A_4] = [B_1] + [B_2][B_4][A_2]$$

Combining Eqns. 3.26 and 3.28 to form a single matrix equation yields

$$\begin{Bmatrix} D^\circ \\ F_s^\circ \end{Bmatrix} = \begin{bmatrix} A_1 & A_2 \\ A_3 & A_4 \end{bmatrix}\begin{Bmatrix} D \\ F_s \end{Bmatrix} + \begin{Bmatrix} 0 \\ B_3 \end{Bmatrix} \qquad 3.29$$

Matrix equation 3.29 relates, at any point, ~~the~~ eight fundamental dependent variables, that appear in the natural boundary conditions of shells of revolution, and their derivatives with respect to the independent variable s.

3.1.4 Solution of the Governing System of Equations

To establish the stiffness matrix, the eight first order differential equations expanded into a Fourier series, represented by matrix Eqn. 3.29, must be solved numerically. In general, Eqn. 3.29 can be written as

$$\{y_s^\circ\} = [A_s]\{y_s\} + \{B_s\} \qquad 3.30$$

where $\{y_s\}$ and $\{y_s^\circ\}$ are vectors of eight dependent variables, four displacement components and four corresponding forces, and their derivatives, respectively. $[A_s]$ is the coefficient matrix relating the variables and their derivatives consisting only of functions of the material properties and geometry of the shell. $\{B_s\}$ is a function of the applied loads.

The general solution of Eqn. 3.30 consists of two parts: the homogeneous solution and the particular solution. From Eqn. 3.30, the form of the homogeneous part is

$$\{h_s^\circ\} = [A_s]\{h_s\} \qquad 3.31$$

and the particular solution is

$$\{P_s^\circ\} = [A_s]\{P_s\} + \{B_s\} \qquad 3.32$$

Now, consider the solution of Eqn. 3.31 for a segment in the region $j \geq s \geq i$. Let the eight arbitrary constants of integration be the eight boundary conditions at edge i, and denote these values by $\{C\}$, then

$$\{h_i\} = \{C\} \qquad 3.33$$

Substituting Eqn. 3.33 into 3.31, for $s = i$,

$$\{h_i^\circ\} = [A_i]\{C\} \qquad 3.34$$

Integrating this numerically, as an initial boundary value problem, allows the value of h_s at any point in the region to be determined as

$$\{h_s\} = [H_s]\{C\} \qquad 3.35$$

where $[H_s]$ represents the matrix arising from the integration of $[A_s]$ along the meridian. $\{C\}$ is a vector of arbitrary constants of integration. For Eqn. 3.35 to reduce to Eqn. 3.33 when s = i, $[H_i]$ must be the identity matrix, i.e.,

$$[H_i] = [I] \tag{3.36}$$

Eqn. 3.36 may be considered to be a 'boundary condition' on the numerical integration of $[H_s]$.

Now, turn to solve Eqn. 3.32, which for s = i may be written as

$$\{P_i^{\circ}\} = [A_i]\{C*\} + \{B_s\} \tag{3.37}$$

where {C*} represents an arbitrary set of initial values of $\{P_i\}$. Integration yields

$$\{P_s\} = [H_s]\{C*\} + \{Q_s\} \tag{3.38}$$

where $[H_s]$ is defined as before, $\{Q_s\}$ is a vector arising from the integration of $\{B_s\}$. Since the particular solution is any solution which satisfies the inhomogeneous equations, it is adequate to select

$$\{C*\} = 0$$

Hence, Eqn. 3.38 reduce to

$$\{P_s\} = \{Q_s\} \tag{3.39}$$

Therefore, the final solution is formed by superimposing the two solutions, Eqns. 3.35 and 3.39.

$$\{y_s\} = [H_s]\{C\} + \{Q_s\} \tag{3.40}$$

3.1.5 Segment Stiffness Matrix

For s = j, Eqn. 3.40 becomes

$$\{y_j\} = [H_j]\{y_i\} + \{Q_j\} \tag{3.41}$$

where each column vector of $[H_j]$ represents the variables at 'j' corresponding to each unit variable applied at 'i' in the absence of any external loads. $\{Q_j\}$ represents the variables at 'j' corresponding to zero displacements, D, and forces, F at 'i' in the presence of the external loads. Thus, Eqn. 3.41 can be expanded into

$$\begin{Bmatrix} D_j \\ F_j \end{Bmatrix} = \begin{bmatrix} H_1 & H_2 \\ H_3 & H_4 \end{bmatrix} \begin{Bmatrix} D_i \\ F_i \end{Bmatrix} + \begin{Bmatrix} Q_d \\ Qf \end{Bmatrix} \qquad 3.42$$

where Q_d and Qf are the displacements and forces from the particular solution respectively. The total matrix in Eqn. 3.42 is usually referred to as a 'transfer matrix'. Expanding Eqn. 3.42 into two equations

$$\begin{Bmatrix} D_i \\ D_j \end{Bmatrix} = \begin{bmatrix} I & 0 \\ H_1 & H_2 \end{bmatrix} \begin{Bmatrix} D_i \\ F_i \end{Bmatrix} + \begin{Bmatrix} 0 \\ Q_d \end{Bmatrix} = [Y_1] \begin{Bmatrix} D_i \\ F_i \end{Bmatrix} + \begin{Bmatrix} 0 \\ Q_d \end{Bmatrix} \qquad 3.43$$

and

$$\begin{Bmatrix} F_i \\ F_j \end{Bmatrix} = \begin{bmatrix} 0 & I \\ H_3 & H_4 \end{bmatrix} \begin{Bmatrix} D_i \\ F_i \end{Bmatrix} + \begin{Bmatrix} 0 \\ Qf \end{Bmatrix} = [Y_2] \begin{Bmatrix} D_i \\ F_i \end{Bmatrix} + \begin{Bmatrix} 0 \\ Qf \end{Bmatrix} \qquad 3.44$$

Solving for $\langle D_i \ F_i \rangle$ and substituting into 3.44 yields

$$\begin{Bmatrix} F_i \\ F_j \end{Bmatrix} = [Y_2][Y_1]^{-1} \begin{Bmatrix} D_i \\ D_j - Q_d \end{Bmatrix} + \begin{Bmatrix} 0 \\ Qf \end{Bmatrix} \qquad 3.45$$

and

$$\begin{Bmatrix} F_i \\ F_j \end{Bmatrix} = [K] \begin{Bmatrix} D_i \\ D_j \end{Bmatrix} + \begin{Bmatrix} F_{o\,i} \\ F_{o\,j} \end{Bmatrix} \qquad 3.46$$

where the coefficients of [K] represent the forces at each shell edge due to a unit displacement at each end while all other displacements are restrained. This matrix is known as the stiffness matrix, $\{F_o\}$ represents the fixed end forces.

3.1.6 Stiffness Matrix Sign Convention

In the derivation of the element stiffness matrix and the fixed end stresses, the sign convention used corresponds to that generally used in shell theory as given in Fig. 2.2. As a result, the stiffness matrix will have some negative elements on the main diagonal. This can be corrected by adapting the so-called 'stiffness matrix sign convention'. This sign convention is shown in Fig. 3.3. It can be seen that the positive direction of the top normal in-place force N_s, the top tangential shear force T_s, the bottom moment M_s, and the bottom transverse shear S_s have been changed to the opposite direction.

3.2 The Flexibility Approach

The solution to the basic shell equations may be split into two parts, namely: the particular solution, which can be simplified to the membrane solution with negligible loss of accuracy, and the homogeneous solution which considers the bending stresses. This procedure is analogous to the flexibility method of analysis for a statically indeterminate structure. Program FLEXSHELL was developed based on this approach. To simplify the shell equations and limit the particular solutions, the following assumptions will be made:

1. Loads are axisymmetric, i.e., $\partial/\partial\theta = 0$, $p_\theta = 0$,

thus, $\partial\phi = d\phi$;

2. The shell segment has uniform thickness; and,
3. z (from Eqns. 2.5 and 2.6 is small compared with the radii of curvature, i.e., $r_1+z \simeq r_1$ and $r_2+z \simeq r_2$.

Thus, the equations of equilibrium become

$$\frac{d(r_0 N_\phi)}{d\phi} - r_1 N_\theta \cos\phi - Q_\phi r_0 + r_0 r_1 p_\phi = 0 \qquad 3.47(a)$$

$$\frac{d(r_0 Q_\phi)}{d\phi} + N_\theta r_1 \sin\phi + r_0 N_\phi - r_0 r_1 p_z = 0 \qquad 3.47(b)$$

$$-\frac{d(r_0 M_\phi)}{d\phi} + r_1 M_\theta \cos\phi + Q_\phi r_0 r_1 = 0 \qquad 3.47(c)$$

and the force-displacement relations become

$$N_\phi = K\left[\frac{1}{r_1}\left(\frac{dv}{d\phi} - w\right) + \frac{\nu}{r_0}(v\cos\phi - w\sin\phi)\right] \qquad 3.48(a)$$

$$N_\theta = K\left[\frac{\nu}{r_1}\left(\frac{dv}{d\phi} - w\right) + \frac{1}{r_0}(v\cos\phi - w\sin\phi)\right] \qquad 3.48(b)$$

$$M_\phi = -D\left[\frac{1}{r_1}\frac{d}{d\phi}\left(\frac{1}{r_1}\frac{dw}{d\phi}\right) + \frac{\nu\cos\phi}{r_0 r_1}\frac{dw}{d\phi}\right] \qquad 3.48(c)$$

$$M_\theta = -D\left[\frac{\nu}{r_1}\frac{d}{d\phi}\left(\frac{1}{r_1}\frac{dw}{d\phi}\right) + \frac{\cos\phi\, dw}{r_0 r_1\, d\phi}\right] \qquad 3.48(d)$$

The method of analysis is outlined as follows:

1. Determine the particular solution forces and the deformations at the edges of the shell due to the applied loads;
2. Establish the flexibility matrix;
3. Solve for the edge forces and moments necessary to restore the incompatibilities of the deformations between adjoining elements;

4. Determine the final stresses by superimposing the particular solution stresses and the stresses due to the incompatibilities.

3.2.1 The Particular Solution

As mentioned earlier, the particular solution is approximated by the membrane solution. The membrane theory of shells approximates the solution to Eqns. 3.47 and 3.48 by neglecting the bending components, based on the assumption that the displacements due to the membrane stresses do not induce any appreciable bending. Thus, Eqn. 3.47 and 3.48 reduce to two equations with two unknowns as shown:

$$(r_0N_\phi)^{\cdot} - r_1N_\theta\cos\phi + r_0r_1p_\phi = 0 \qquad 3.49(a)$$

$$r_1N_\theta\sin\phi + r_0N_\phi + r_0r_1p_z = 0 \qquad 3.49(b)$$

The in-plane forces N_ϕ and N_θ are obtained more simply from the vertical and normal equilibrium of the statically determinate shell segment under the applied loads. Since the radii of curvature r_1 and r_2 vary in form depending on the type of shell of revolution, so does the form of the membrane solution.

1. Cylinder

$$N_s = -\int_s p_s ds \qquad 3.50(a)$$

$$N_\theta = -p_z r \qquad 3.50(b)$$

2. Sphere

$$N_\phi = \frac{-R}{2\pi r_0\sin\phi} \qquad 3.51(a)$$

$$N_\theta = \frac{\pm R}{2\pi r_1 \sin^2\phi} \mp p_z r_2 \qquad 3.51(b)$$

3. Cone

$$N_s = \frac{-R}{2\pi s \cos\alpha} \qquad 3.52(a)$$

$$N_\theta = \mp p_z r_2 \qquad 3.52(b)$$

where R is the total vertical load, positive when directed toward the supports; p_z is the component of the external load per unit area normal to the shell surface in the direction towards the axis of revolution. The upper and lower signs relate to Figs. (a) and (b) respectively, of Tables 3.4 and 3.6. The expression for the membrane in-plane forces for the spherical, cylindrical, and conical segments, subjected to various loading conditions shown in Tables 3.4 to 3.6 were derived from Eqns. 3.50 to 3.52. The solution due to the thermal effects were obtained from Billington(3).

3.2.2 The Homogeneous Solution

Consider the vertical equilibrium of a shell element, then

$$2\pi r_0 N_\phi \sin\phi + 2\pi r_0 Q_\phi \cos\phi + R = 0$$

from which

$$N_\phi = -Q_\phi \cot\phi - \frac{R}{2\pi r_0 \sin\phi} \qquad 3.53$$

where R is defined as before. Note that the second term is the membrane force which can be evaluated separately as shown earlier. Therefore, the homogeneous solution is obtained by solving the simplified shell equations (Eqns. 3.47 to 3.48) ignoring all load terms. Thus, the homogeneous solution for the meridional force is

$$N_\phi = -Q_\phi \cot\phi \qquad 3.54$$

Substituting this into Eqn. 3.47(b), ignoring the load term p_z, and using the relation,

$$r_0 = r_2 \sin\phi \qquad 3.55$$

then

$$N_\theta = -\frac{r_2}{r_1}\frac{dQ_\phi}{d\phi} \qquad 3.56$$

Let

$$U = r_2 Q_\phi \qquad 3.57$$

$$V = \frac{1}{r_1}\left[v + \frac{dw}{d\phi}\right] \qquad 3.58$$

Eqns. 3.54 and 3.56 become

$$N_\phi = -\frac{1}{r_2} U \cot\phi \qquad 3.59$$

$$N_\theta = -\frac{1}{r_1}\frac{dU}{d\phi} \qquad 3.60$$

Rearranging Eqns. 3.48(a) and 3.48(b), and substituting Eqn. 3.55 yields,

$$\frac{dv}{d\phi} - w = \frac{r_1}{Eh}(N_\phi - \nu N_\theta) \qquad 3.61$$

$$v\cot\phi - w = \frac{r_2}{Eh}(N_\theta - \nu N_\phi) \qquad 3.62$$

from which w may be eliminated to yield,

$$\frac{dv}{d\phi} - v\cot\phi = \frac{1}{Eh}[(r_1+\nu r_2)N_\phi - (r_2+\nu r_1)N_\theta] \qquad 3.63$$

Differentiating Eqn. 3.62, and combining with Eqn. 3.63 gives,

$$v + \frac{dw}{d\phi} = r_1 V = \frac{\cot\phi}{Eh}[(r_1+\nu r_2)N_\phi - (r_2+\nu r_1)N_\theta]$$

$$-\frac{d}{d\phi}\left[\frac{r_2}{Eh}(N_\theta-\nu N_\phi)\right] \qquad 3.64$$

Substituting Eqns. 3.59 and 3.60 into Eqn. 3.64 yields one equation with U and V terms only.

$$\frac{r_2}{r_1^2}\frac{d^2U}{d\phi^2}+\frac{1}{r_1}\left[\frac{d}{d\phi}\left(\frac{r_2}{r_1}\right)+\frac{r_2\cot\phi}{r_1}-\frac{r_2}{r_1 h}\frac{dh}{d\phi}\right]\frac{dU}{d\phi}$$

$$-\frac{1}{r_1}\left[\frac{r_1\cot\phi}{r_2}-\nu-\frac{\nu}{h}\frac{dh}{d\phi}\cot\phi\right]U = EhV \qquad 3.65$$

Substituting Eqns. 3.57 and 3.58 into Eqns. 3.48(c) and 3.48(d),

$$M_\theta = -D\left[\frac{V}{r_2}\cot\phi+\frac{\nu}{r_1}\frac{dV}{d\phi}\right] \qquad 3.66$$

$$M_\phi = -D\left[\frac{\nu V\cot\phi}{r_2}+\frac{1}{r_1}\frac{dV}{d\phi}\right] \qquad 3.67$$

Substituting these two equations and Eqn. 3.57 into Eqn. 3.47(c) yields,

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

$$-\frac{1}{r_1}\left[\nu-\frac{3\nu\cot\phi}{h}\frac{dh}{d\phi}+\frac{r_1}{r_2}\cot^2\phi\right]V = -\frac{U}{D} \qquad 3.68$$

Eqns. 3.65 and 3.68 permit a closed form solution of the equations of shells of revolution. The solution of these equations may be further simplified by applying the geometrical properties of each shell.

For the cylindrical segment with $r_1 = \infty$ and $r_0 = r_2 = r$, these equations reduce to the form (See Appendix A for details)

$$\frac{d^4\Delta_H}{ds^4} + 4\beta^4\Delta_H = 0 \tag{3.69(a)}$$

where

$$\beta^4 = \frac{3(1-\nu^2)}{r^2h^2} \tag{3.69(b)}$$

for which the solution can be expressed in closed form as

$$\Delta_H = e^{\beta s}(C_1\cos\beta s + C_2\sin\beta s) + e^{-\beta s}(C_3\cos\beta s + C_4\sin\beta s) \tag{3.70}$$

For the conical segment with r_0 = s sinα, $r_1 = \infty$, r_2 = s tanα, and $\phi = \pi/2 - \alpha$, the closed form solution in terms of the Kelvin functions ber, bei, ker, kei as shown in detail in Appendix A is

$$Q_s = \frac{1}{s}(C_1\mathrm{ber}_2\xi + C_2\mathrm{bei}_2\xi + C_3\mathrm{ker}_2\xi + C_4\mathrm{kei}_2\xi) \tag{3.71}$$

where

$$\lambda^4 = \frac{12(1-\nu^2)}{h^2\tan^2\alpha} \tag{3.72}$$

$$\xi = 2\lambda\sqrt{s} \tag{3.73}$$

For a spherical segment with $r_1 = r_2 = a$ and r_0 = a sinϕ, Eqns. 3.65 and 3.68 become

$$\frac{d^2Q_\phi}{d\phi^2} + \cot\phi\frac{dQ_\phi}{d\phi} - (\cot^2\phi - \nu)Q_\phi = EhV \tag{3.74}$$

$$\frac{d^2V}{d\phi^2} + \cot\phi\frac{dV}{d\phi} - (\cot^2\phi - \nu)V = \frac{-a^2Q_\phi}{D} \tag{3.75}$$

Assume that the bending effects are significant over only a short distance from their point of introduction. Thus, Eqns. 3.74 and 3.75 can be reduced to one differential equation in terms of a single variable, for which the closed form solution is a function of

$$e^{\pm\lambda\phi}\{\cos\lambda\phi, \sin\lambda\phi\}$$

where λ is large and dimensionless. Note that each time the solution is differentiated with respect to ϕ, the result is a multiple of the large parameter λ. Consequently, the second derivative will be two orders of λ greater than the solution itself and so on. Therefore,

$$\frac{d^2Q_\phi}{d\phi^2} \gg \frac{dQ_\phi}{d\phi} \gg Q_\phi$$

So all lower order derivatives with respect to ϕ may be neglected in the formulation of the final solution. This assumption was first introduced by Geckeler in 1926. Hence, it will be referred to as Geckeler's assumption (2,9). Thus, Eqns. 3.74 and 3.75 reduce to

$$\frac{d^2Q_\phi}{d\phi^2} = EhV \qquad 3.76$$

$$\frac{d^2V}{d\phi^2} = -a^2Q_\phi \qquad 3.77$$

Combining these to eliminate V,

$$\frac{d^4Q_\phi}{d\phi^4} + 4\lambda^4Q_\phi = 0 \qquad 3.78$$

where

$$\lambda^4 = 3(1-\nu^2)\frac{a^2}{h^2} \qquad 3.79$$

The final solution is

$$Q_\phi = e^{\lambda\phi}(C_1\cos\lambda\phi + C_2\sin\lambda\phi) + e^{-\lambda\phi}(C_3\cos\lambda\phi + C_4\sin\lambda\phi) \qquad 3.80$$

where C_1, C_2, C_3, and C_4 are arbitrary constants of integration. The limitations of this approximation will be discussed in Chapter 5.

3.2.3 Segment Flexibility Matrix

The construction of the flexibility matrix for the cylindrical, conical, and spherical segments will be discussed in this section. Note that for the spherical segment, Geckeler's assumption will be used as indicated. By definition, the flexibility matrix coefficient, say F(i,j), is the deformation of the segment at i due to a unit value of the load applied to the segment at j.

Only those deformations which violate continuity and the corresponding forces which produce these deformations need be identified in the formulation of the flexibility matrix. For axisymmetric loading, these are the horizontal displacement Δ_H and the meridional rotation Δ_ϕ and the corresponding forces H and M_ϕ at each discontinuous edge of the shell. Combining Eqn. 3.76 and 3.67 and again for the sphere neglecting the lower order differentials with respect to ϕ, the expression for the meridional moment becomes

$$M_\phi = \frac{-D}{Eha}\frac{d^3Q_\phi}{d\phi^3} \qquad 3.81$$

and from the geometry of the shell, the horizontal force, H , can be expressed as a function of the meridional force N_ϕ. Thus, for a spherical segment

$$H = N_\phi \cos\phi \qquad 3.82$$

Consistent with the sign conventions shown in Fig. 3.4 expressions for the moment and horizontal force at each shell edge may be expressed in terms of the homogeneous solution, as shown in matrix form below for the spherical segment (Eqn. 3.74)

Let

$\phi_1 = e^{\lambda\phi}\cos\lambda\phi \qquad \theta_1 = e^{\lambda\phi}(\cos\lambda\phi + \sin\lambda\phi)$

$\phi_2 = e^{\lambda\phi}\sin\lambda\phi \qquad \theta_2 = e^{\lambda\phi}(\cos\lambda\phi - \sin\lambda\phi)$

$\phi_3 = e^{-\lambda\phi}\cos\lambda\phi \qquad \theta_3 = e^{-\lambda\phi}(\cos\lambda\phi + \sin\lambda\phi)$

$\phi_4 = e^{-\lambda\phi}\sin\lambda\phi \qquad \theta_4 = e^{-\lambda\phi}(\cos\lambda\phi - \sin\lambda\phi)$

then

$$\begin{Bmatrix} H^i \\ M_\phi^i \\ H^j \\ M_\phi^j \end{Bmatrix} = \begin{bmatrix} \sin\alpha & 0 & 0 & 0 \\ 0 & a/2\lambda & 0 & 0 \\ 0 & 0 & \sin\alpha & 0 \\ 0 & 0 & 0 & a/2\lambda \end{bmatrix} \begin{bmatrix} \phi_1^i & \phi_2^i & \phi_3^i & \phi_4^i \\ \theta_1^i & \theta_2^i & \theta_4^i & \theta_3^i \\ \phi_1^j & \phi_2^j & \phi_3^j & \phi_4^j \\ \theta_1^j & \theta_2^j & \theta_4^j & \theta_3^j \end{bmatrix} \begin{Bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{Bmatrix}$$

or simply,

$$\{V\} = [T_1][T_2]\{C\}$$

Multiplying matrices $[T_1]$ and $[T_2]$ simplifies to

$$\{V\} = [TT]\{C\} \qquad 3.83$$

Similarly, expressions for the deformations at each shell edge may be expressed in terms of the homogeneous solution as follows:

$$\begin{Bmatrix} \Delta_H^i \\ \Lambda_\phi^i \\ \Delta_H^j \\ \Delta_\phi^j \end{Bmatrix} = \frac{1}{Et} \begin{bmatrix} \lambda\sin\alpha & 0 & 0 & 0 \\ 0 & 2\lambda^2 & 0 & 0 \\ 0 & 0 & \lambda\sin\alpha & 0 \\ 0 & 0 & 0 & 2\lambda^2 \end{bmatrix} \begin{bmatrix} -\theta_2^i & \theta_1^i & \theta_3^i & \theta_4^i \\ -\theta_2^i & \theta_1^i & \theta_4^i & -\theta_3^i \\ -\theta_2^j & \theta_1^j & -\theta_3^j & \theta_4^j \\ -\phi_2^j & \phi_1^j & \phi_4^j & -\phi_3^j \end{bmatrix} \begin{Bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{Bmatrix}$$

or simply

$$\{\Delta\} = [T_3][T_4]\{C\}$$

Multiplying matrices $[T_3]$ and $[T_4]$ yields

$$\{\Delta\} = [TA]\{C\} \qquad 3.84$$

Combining Eqn. 3.83 and 3.84 yields

$$\{\Delta\} = [TA][TT]^{-1}\{V\} \quad 3.85(a)$$

$$\{\Delta\} = [F]\{V\} \quad 3.85(b)$$

where [F] is the segment flexibility matrix, such that

$$[F] = [TA][TT]^{-1} \quad 3.85(c)$$

Similarly, the flexibility matrices for the cylindrical and conical segments are constructed using the homogeneous solutions, Eqns. 3.70 and 3.71 respectively, as shown in detail in Appendix B.

The base segment is considered to be a circular plate supported on a Winkler type foundation, whose stiffness is expressed as the subgrade modulus, k (4). The segment flexibility matrix was developed in the same manner as the spherical, cylindrical, and conical segments, based on the asymptotic solution to the fourth order plate equation.

$$\left(\frac{d^2}{dr^2} + \frac{1}{r}\frac{d}{dr}\right)\left(\frac{d^2w}{dr^2} + \frac{1}{r}\frac{dw}{dr}\right) = \frac{q - kw}{D}$$

where w is the deformation component, r is the radius of the circular plate, q is the load term, and D is the flexural rigidity.

TABLE 3.1 Coefficients of Matrices B_1, B_2 and Load Vector B_3 in Eqn. 3.19

$R_o \cos\phi$	$\mp R_o n$	$\mp R_o n$		
$R_o^2 n^2$	$\mp R_o^2 n \cos\phi$	$\mp R_o^2 n \cos\phi$	$- R_2$	
		$\mp R_o R_1 n$	$R_o \cos\phi$	$\mp R_o n$
$\mp R_o R_2 n$	$R_o R_2 \cos\phi$	$-R_o \cos\phi$	$\pm R_o n$	$-R_o \cos\phi$

Matrix B_2

$-R_o \cos\phi$			
	$-R_o \cos\phi$	$- R_1$	
	R_1	$-R_o \cos\phi$	
			$-R_o \cos\phi$

Matrix B_1

P_r
$-P_s$
$-P_\theta$

Vector B_3

TABLE 3.2 Coefficients of Matrix A_1 and A_2 in Eqn. 3.26

$R_1 r_2^o - \nu R_o \cos\phi \frac{CA_1}{CA_2}$	$-\frac{\nu K R_2 (R_1-R_2)}{CA_2}$ $- \nu R_o^2 n^2 \frac{CA_1}{CA_2}$	$-R_1^2 r_1^o$ $-\frac{\nu K R_o \cos\phi (R_1-R_2)}{CA_2}$	$\mp \nu R_0 R_1 n$
-1		R_1	
$-\nu D \frac{R_o \cos\phi (R_1-R_2)}{CA_2}$	$-R_1 - \frac{\nu K R_2}{CA_2}$ $-\frac{\nu D R_o^2 n^2 (R_1-R_2)}{CA_2}$	$-\frac{\nu K R_o \cos\phi}{CA_2}$	$\mp \nu R_o n$
$\mp\frac{D R_o n (R_1-3R_2)}{CA_3}$	$\mp\frac{D R_o^2 n \cos\phi (R_1-3R_2)}{CA_3}$	$\pm \frac{R_o n CA_1}{CA_3}$ $- \frac{D n R_o R_1 R_2}{CA_3}$	$R_o \cos\phi$

Matrix A_1

$-\frac{CA_1}{DCA_2}$		$\frac{R_1 - R_2}{CA_2}$	
$-\frac{(R_1 - R_2)}{CA_2}$		$\frac{1}{CA_2}$	
			$(\frac{2}{1-\nu}) \frac{1}{CA_3}$

Matrix A_2

TABLE 3.3 (a) Coeffiecients of Matrix B_4 in Eq. 3.27

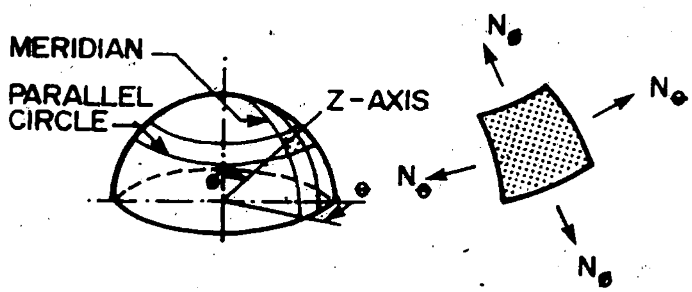
$-\nu D$			
			$-(\frac{1-\nu}{2}) D R_2$
			$(\frac{1-\nu}{2})\{D(R_1-R_2\}$
		νK	
			$(\frac{1-\nu}{2}) K$

Matrix B_4

$\nu DR_1 r_1^o - DR_o \cos\phi$	$-DR_o^2 n^2 - DR_2(R_1 - R_2)$	$-\gamma DR_1^2 r_1^o$ $-DR_o \cos\phi\,(R_1 - R_2)$	$\mp DR_o R_1 n$
$\pm(1-\nu)DR_o n$	$\mp(1-\nu)DR_o^2 n \cos\phi$	$\mp(\frac{1-\nu}{2})\,R_o R_2 n$	$(\frac{1-\nu}{2})DR_o R_2 \cos\phi$
$\pm(1-\nu)DR_o n$,	$(1-\nu)DR_o^2 n \cos\phi$	$\mp(\frac{1-\nu}{2})DR_o R_1 n$	$-(\frac{1-\nu}{2})DR_o \cos\phi\,(R_1 - 2R_2)$
$DR_o \cos\phi\,(R_1 - R_2)$	$K(R_2 + \nu R_1)$ $+D(R_1 - R_2)\,(R_o^2 n^2 - R^2)$	$R_o \cos\phi\{K - DR_2\,(R_1 - R_2)\}$	$\pm KR_o n$
$\mp(\frac{1-\nu}{2})DR_o n\,(R_1 - R_2)$	$\mp(\frac{1-\nu}{2})DR_o^2 n \cos\phi(R_1 - R_2)$	$\mp(\frac{1-\nu}{2})R_o n[K - D(R_1 - R_2)^2]$	$-(\frac{1-\nu}{2})KR_o \cos\phi$

TABLE 3.3 (b) Coefficients of Matrix B_5 in Eqn. 3.27

Table 3.4 MEMBRANE SOLUTION FOR A SPHERICAL SEGMENT



— UPPER AND LOWER SIGNS RELATE TO Figs. (a) AND (b) RESPECTIVELY

— $\psi_a = \alpha_0$
$\psi_b = \alpha$

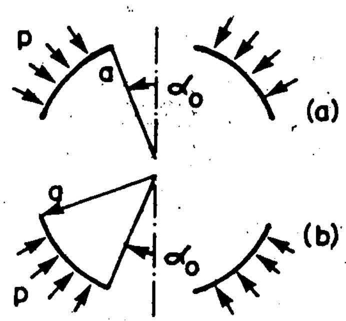
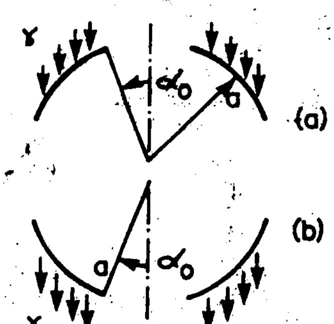
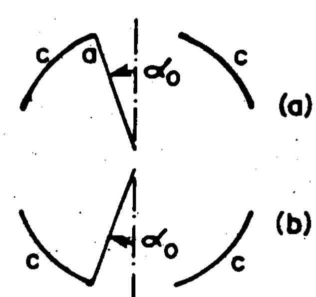
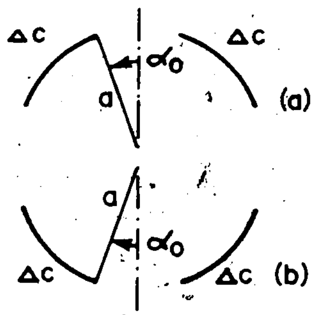
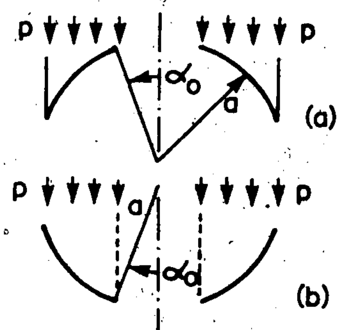
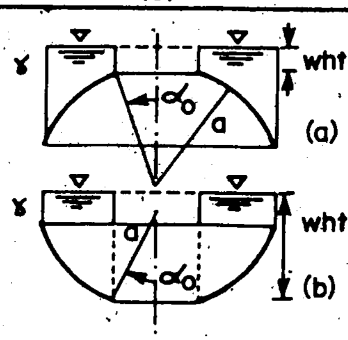
LOAD CASE	IN-PLANE FORCES & DEFORMATIONS
 (1) & (3)	$N_\phi = \mp \frac{pa}{2} \left(1 - \frac{\sin^2 \psi}{\sin^2 \phi}\right)$ $N_\theta = - \frac{pa}{2} \left(1 + \frac{\sin^2 \psi}{\sin^2 \phi}\right)$
 (2)	$N_\phi = \mp \gamma h a \frac{(\cos \psi - \cos \phi)}{\sin^2 \phi}$ $N_\theta = \gamma h a \left[\frac{(\cos \psi - \cos \phi)}{\sin^2 \phi} \mp \cos \phi \right]$
 (4)	$\Delta_H = \mp C \alpha_T a \sin \phi$

Table 3.4 (cont'd)

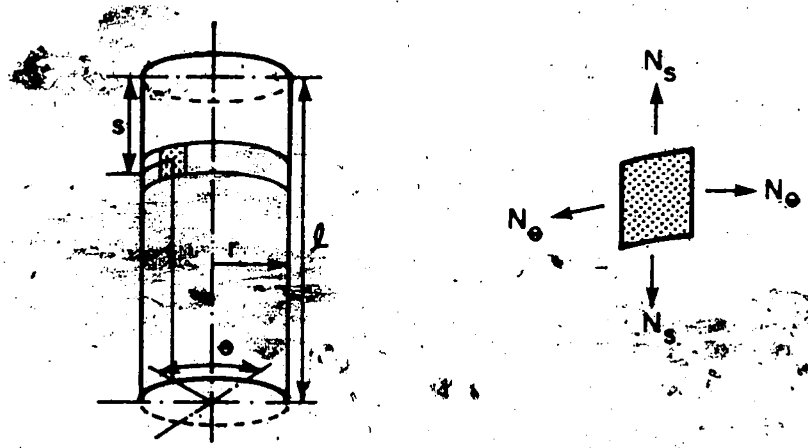
LOAD CASE	IN-PLANE FORCES & DEFORMATIONS
 (5)	$M_\phi = \frac{\Delta c \, \alpha_T E h^2}{12(1-\nu)}$ $M_\theta = M_\phi$
 (6)	$N_\phi = \mp \frac{pa}{2}\left(1 - \frac{\sin^2\alpha_0}{\sin^2\phi}\right)$ $N_\theta = \pm \frac{pa}{2}\left[\left(1 - \frac{\sin^2\alpha_0}{\sin^2\phi}\right) \mp 2\cos^2\phi\right]$
 (7)	$N_\phi = \mp \gamma a\left[\frac{(wht \pm a\cos\alpha_0)}{2}\left(1 - \frac{\sin^2\alpha_0}{\sin^2\phi}\right) \pm \frac{a}{3}\,\frac{(\cos^3\phi - \cos^3\alpha_0)}{\sin^2\phi}\right]$ $N_\theta = \mp \gamma a\left[\frac{(wht \pm a\cos\alpha_0)}{2}\left(1 + \frac{\sin^2\alpha_0}{\sin^2\phi}\right) \mp \frac{a}{3}\,\frac{(\cos^3\phi - \cos^3\alpha_0)}{\sin^2\phi} \mp a\cos\phi\right]$

NOTE:

$$\Delta_H = \frac{a \sin\phi}{Eh}\,(N_\theta - \nu N_\phi)$$

$$\Delta_\phi = \frac{\cot\phi}{Eh}\,(1+\nu)(N_\theta - N_\phi) - \frac{d}{d\phi}\,[N_\theta - \nu N_\phi]\,\frac{1}{Eh}$$

Table 3.5 MEMBRANE SOLUTION FOR A CYLINDRICAL SEGMENT



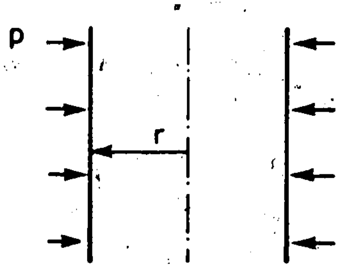
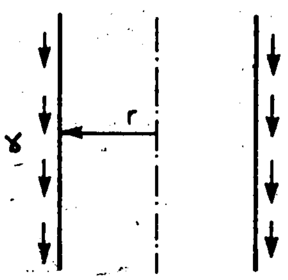
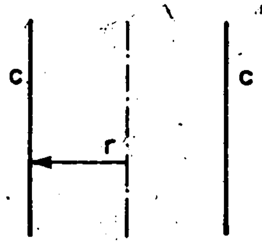
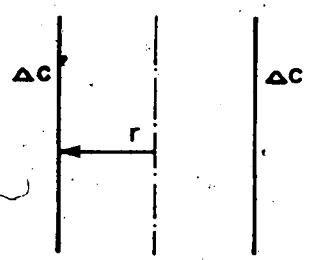
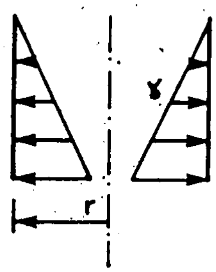
LOAD CASE	IN-PLANE FORCES & DEFORMATIONS
 (1) & (3)	$N_s = 0$ $N_\theta = -pr$
 (2)	$N_s = -\gamma hs$ $N_\theta = 0$
 (4)	$\Delta_H = -c\alpha_T r$

Table 3.5 (cont'd)

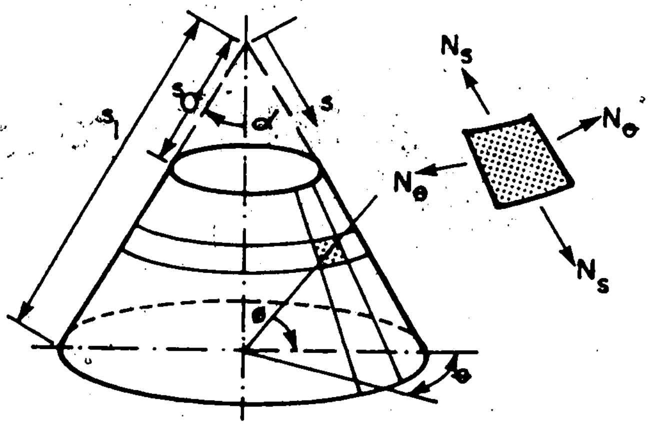
LOAD CASE	IN-PLANE FORCES & DEFORMATION
 (5)	$M_s = \dfrac{\Delta c \alpha_T E h^2}{12(1-\nu)}$ $M_\theta = M_s$
 (7)	$N_s = 0$ $N_\theta = \gamma r s$

NOTE:

$$\Delta_H = \frac{r}{Eh}\left(N_\theta - \nu N_s\right)$$

$$\Delta_\phi = \frac{-d}{ds}\left(N_\theta - \nu N_s\right)$$

Table 3.6 MEMBRANE SOLUTION FOR A CONICAL SEGMENT



– UPPER AND LOWER SIGNS RELATE TO Figs (a) AND (b) RESPECTIVELY

– $y_{(a)} = s_0$

$y_{(b)} = s_1$

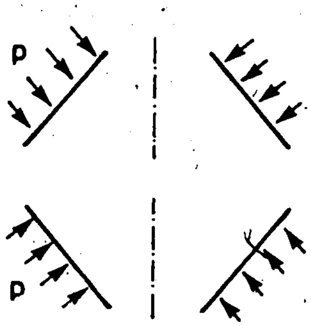
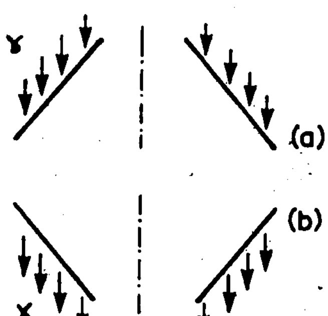
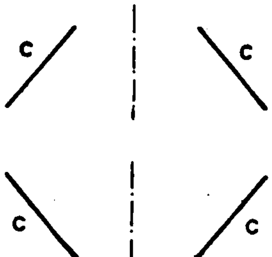
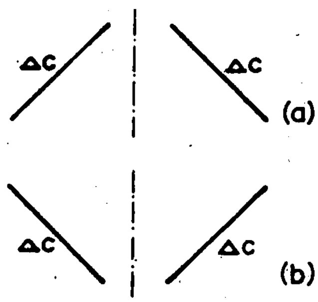
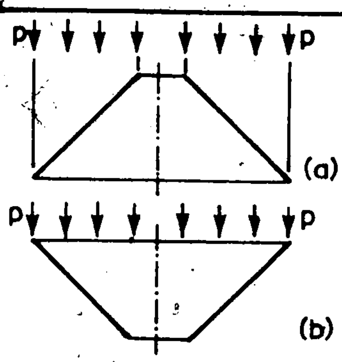
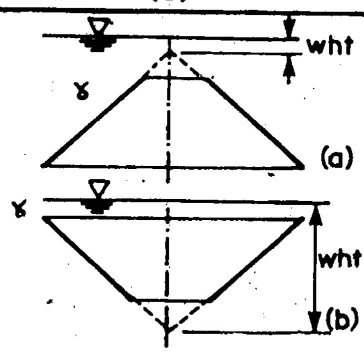
LOAD CASE	IN-PLANE FORCES & DEFORMATIOS
 (1) & (3)	$N_s = -p \tan\alpha \frac{(s^2-y^2)}{2s}$ $N_\theta = -ps \tan\alpha$
 (2)	$N_s = \mp \frac{\gamma h (s^2-y^2)}{2s \cos\alpha}$ $N_\theta = \mp \gamma hs \tan\alpha \sin\alpha$
 (4)	$\Delta_H = -c\alpha_T s \sin\alpha$

Table 3.6 (cont'd)

LOAD CASE	IN-PLANE FORCES & DEFORMATIONS
 (5)	$M_s = \dfrac{\Delta c \alpha_T E h^2}{12(1-\nu)}$ $M_\theta = M_s$
 (6)	$N_s = \mp p \tan\alpha \dfrac{(s^2 - y^2)}{2s}$ $N_\theta = \mp ps \sin^2\alpha \tan\alpha$
 (7)	$N_s = \dfrac{-\gamma s \tan\alpha}{6} \left[3\,wht\left(1 - \dfrac{y^2}{s^2}\right) \pm 2 s \cos\alpha \left(1 - \dfrac{y^3}{s^3}\right)\right]$ $N_\theta = -\gamma s \tan\alpha\,[wht \pm s\cos\alpha]$

NOTE :

$$\Delta H = \frac{s \sin\alpha}{Eh}(N_\theta - N_s)$$

$$\Delta_\theta = \frac{\tan\alpha}{Eh}\left[(1+\nu)(N_s - N_\theta) - \frac{1}{s}\frac{d}{ds}(N_\theta - \nu N_s)\right]$$

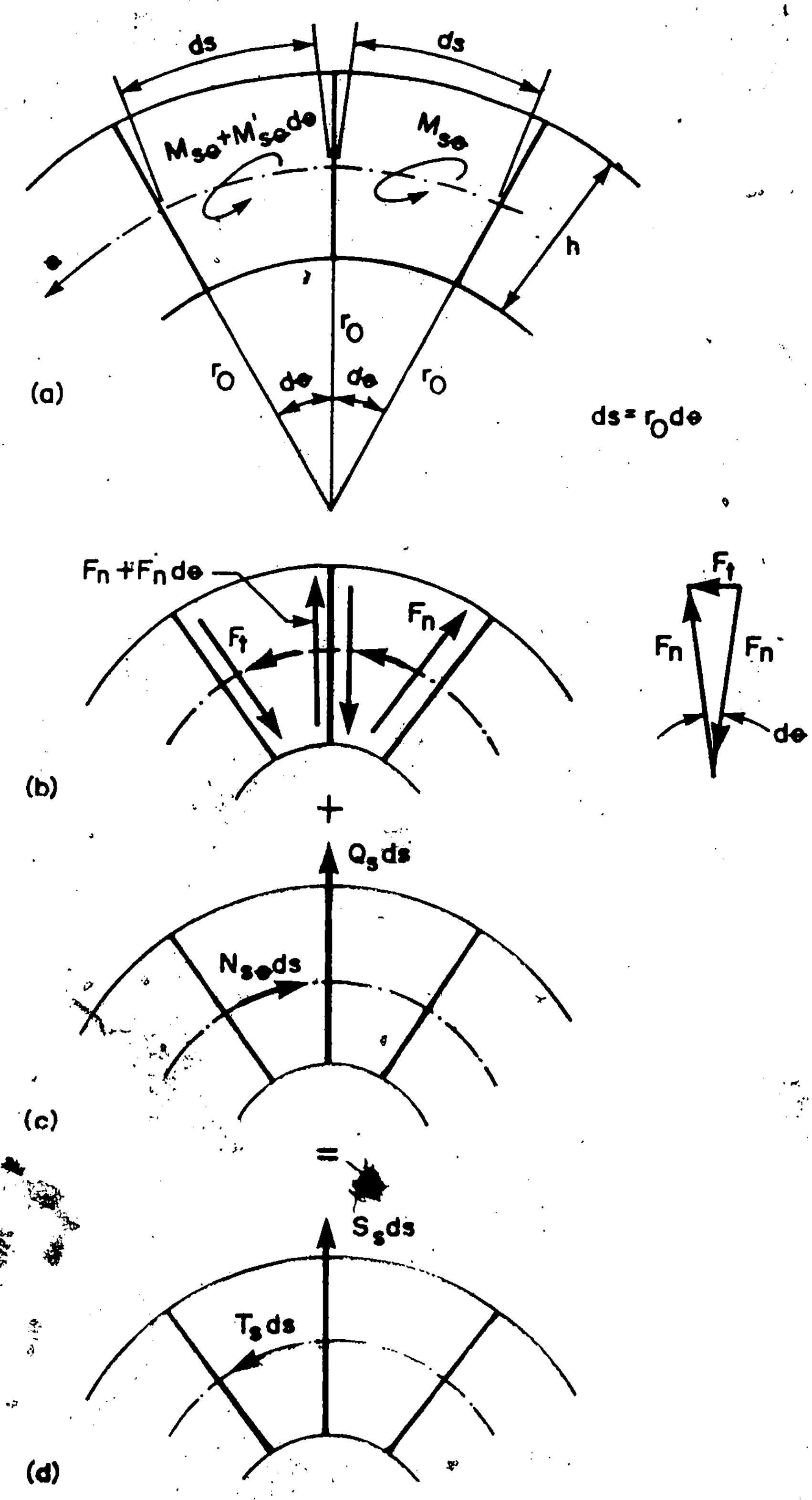


Figure 3.1 EFFECTIVE SHEARING FORCES EXPRESSED AS A FUNCTION OF THE IN-PLANE SHEAR AND THE TWISTING MOMENT

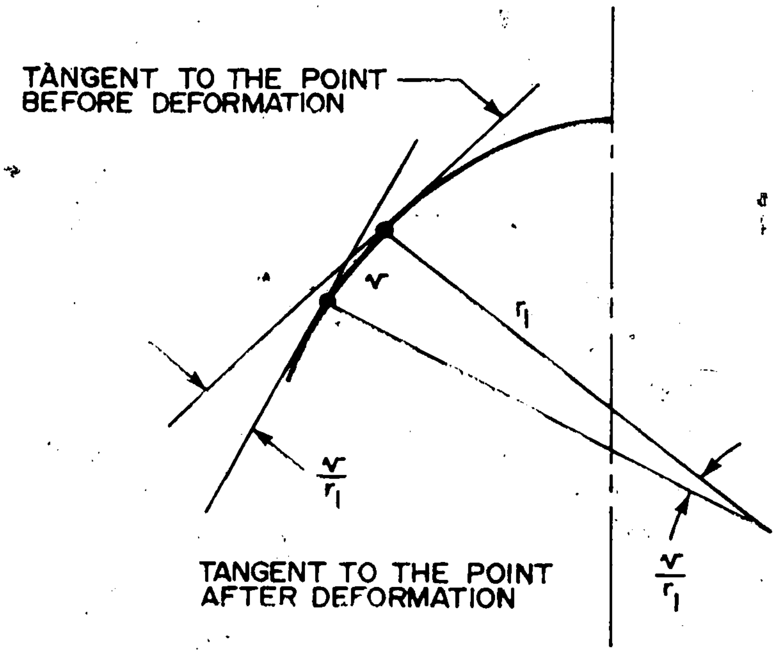


Figure 3.2(a) MERIDIONAL ROTATION β DUE TO DISPLACEMENT v

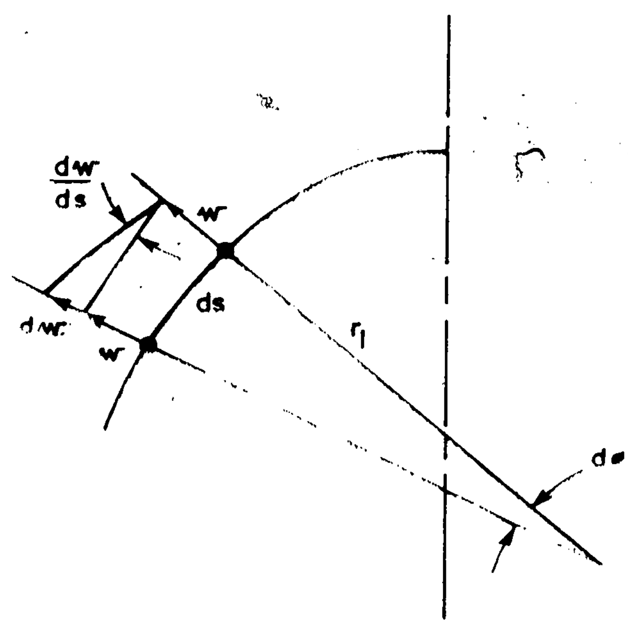


Figure 3.2(b) MERIDIONAL ROTATION β DUE TO DISPLACEMENT w

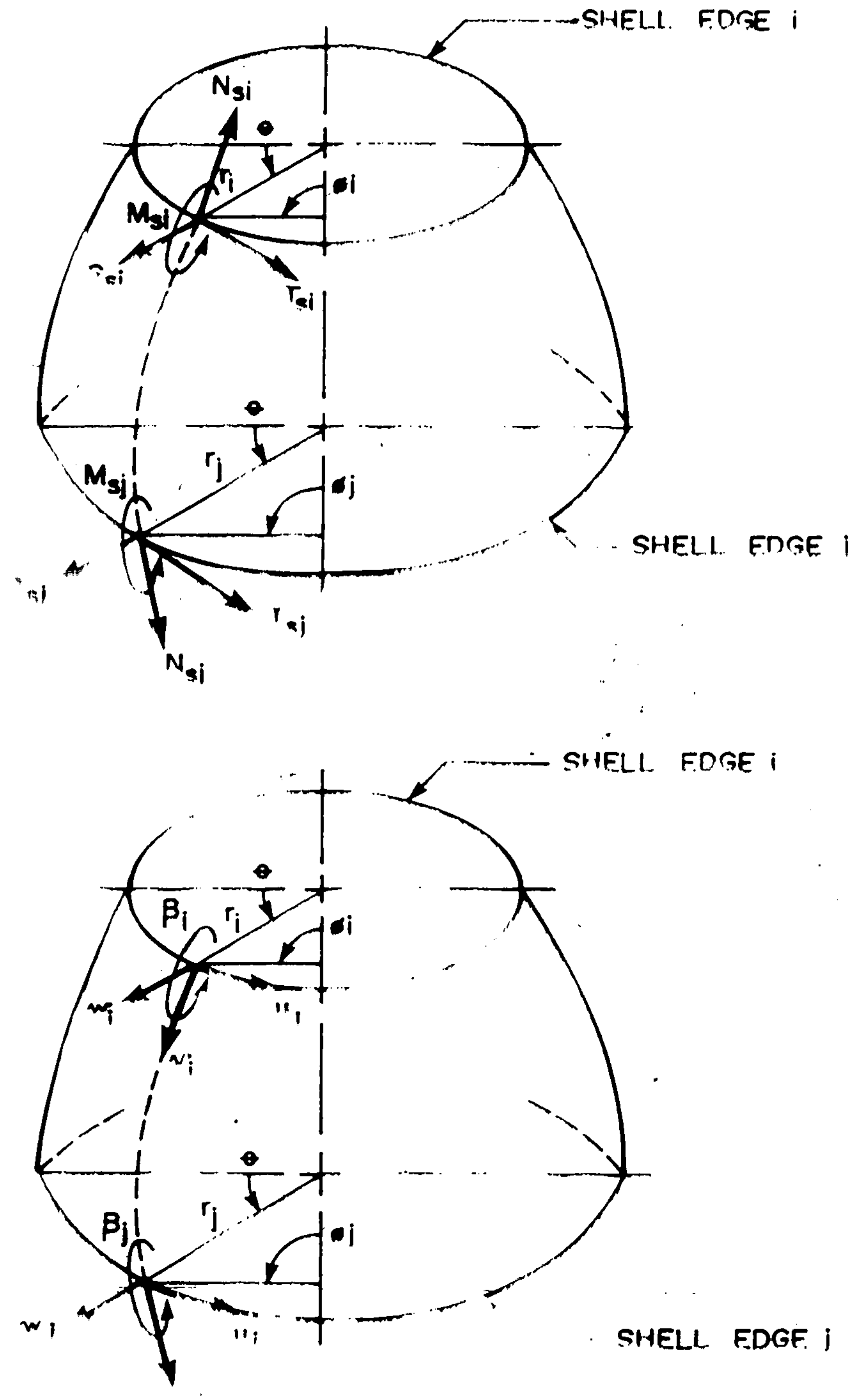


Figure 3.3 SASHELL STIFFNESS MATRIX SIGN CONVENTION

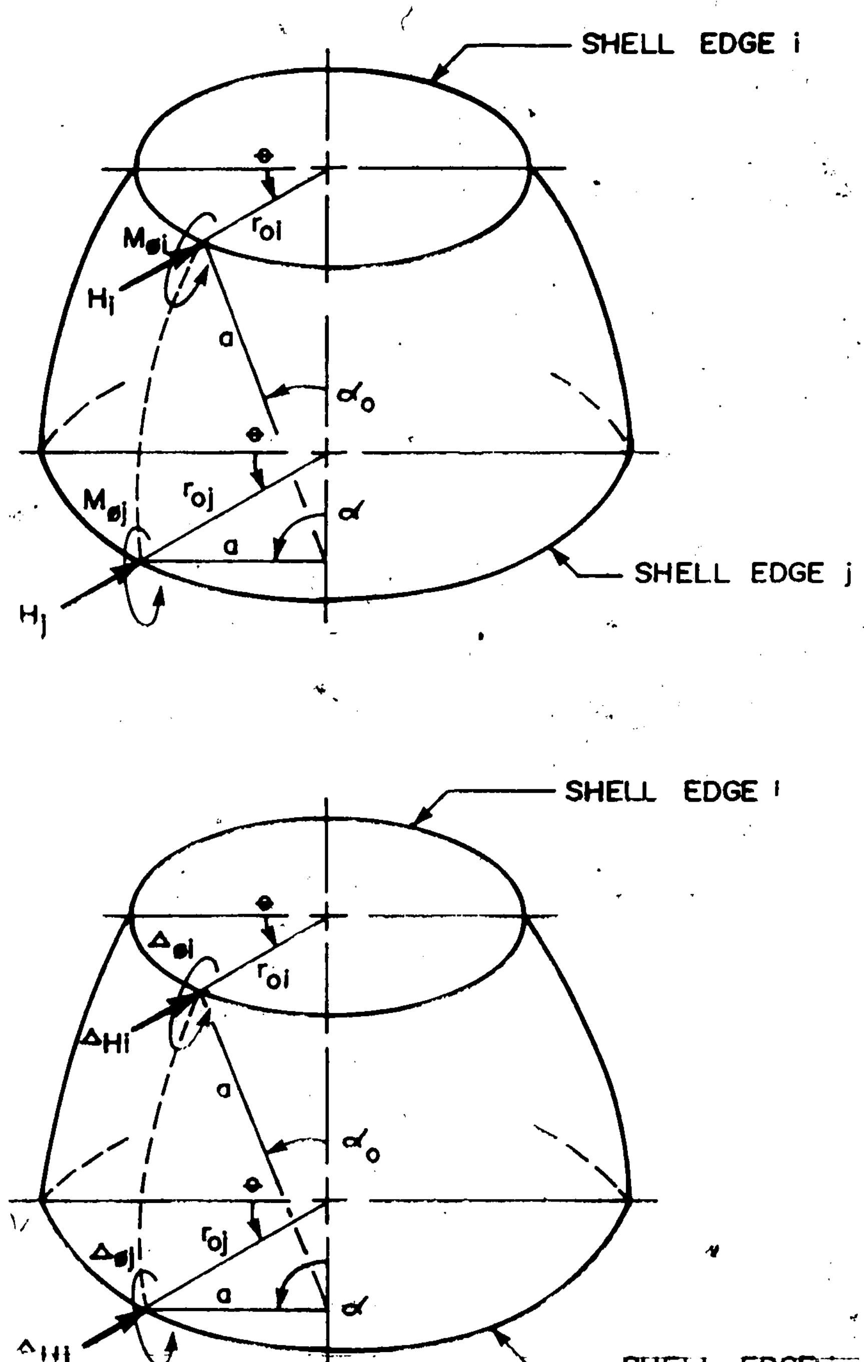


Figure 3.4 FLEXSHELL FLEXIBILITY MATRIX SIGN CONVENTION

4. FLEXSHELL FORMULATION

Program FLEXSHELL analyzes a multi-shell structure based on the flexibility approach described in Section 3.2. The program listed in Appendix C is a modification of an earlier version developed by Murray, et al(4) for the analysis of the Gentilly type containment structure. The capability of the program has since been increased for a wider variety of multi-shell problems.

The 'long' sphere was replaced by a 'short' spherical segment. And three additional segments were introduced such as: the 'short' conical segment, the 'short' inverted spherical and conical segments. (Fig. 4.1) Furthermore, two load cases were added, namely: the liquid pressure loading and the snow load, which is a uniform pressure over a horizontal projection of the shell segment. Consequently, a significant portion of the original version of the program was recoded. Specifically,

1. The membrane solution forces developed in Chapter 3, shown in detail in Tables 3.4 to 3.6, were coded directly into the program. The upper and lower signs relate to the dome and inverted dome configurations.
2. The construction of the flexibility matrix was coded from the product

   $$[TT]^{-1}[TA]$$

   whose coefficients were coded directly into the program. Provisions were made for the dome and the inverted dome configurations and also for the closed spherical and conical segments.

3. The calculation of the particular solution displacements were coded using the expressions for Δ_H and Δ_ϕ, shown in Tables 3.4 to 3.6, using the membrane solution forces already coded into the program (Step 1).
4. The special homogeneous solutions (4), which considers the effect of the vertical edge load was replaced.
5. Finally, the calculation of the final stress resultants and displacements were coded to conform to the 'short' segments that were introduced.

The logic flow of program FLEXSHELL is as follows:

1. Define the segment connectivities;
2. Satisfy the rigid body motion requirement by evaluating the effect of the vertical load components on the segment;
3. Evaluate the joint eccentricity effects;
4. Establish the segment flexibility matrix;
5. Solve the compatibility equations;
6. Solve for the final stress resultant values by superimposing the particular solution stresses with the stresses due to bending.

4.1 Definitions and Notations

Segments are defined with reference to a coordinate starting at the line of symmetry at the apex of a structure, and traversing the midsurface of the shell segment in a counter clockwise sense, until again reaching the symmetry line at the base of the structure. The coordinate for branches which do not fall in this primary circuit may be defined in the same manner, starting at the free edge, increasing in a counter clockwise sense. Therefore, with the exception of the last segment in the primary circuit, the 'bottom' of each segment is always supported by the 'top' of the adjacent segment. For reasons which will be explained later, the segments must be numbered sequentially in such manner that any segment always has a higher number than any of the segments which it supports.

Consider a shell segment cut by a vertical plane shown in Fig. 4.1, the sign conventions consistent throughout the program are as follows:

1. Moments and rotations are positive as shown in the figure.
2. Horizontal forces, displacements, and eccentricities are positive in the direction towards the line of symmetry, also known as the axis of revolution.
3. Vertical forces are positive downward.
4. For base segments, vertical displacements are positive downward, whereas, vertical eccentricities are positive upwards.

Note that all forces and moments are expressed per unit length.

4.2 Connectivity Matrix

The connectivity matrix is established by satisfying the geometric compatibility requirements between adjacent segments. This is accomplished by forming the algebraic summation of the horizontal displacements and meridional rotations at adjacent edges of the shell segments. To express these equations in matrix form, it is necessary to number the segments as described earlier, to ensure the consistency in the order of assembly of the end deformations. Furthermore, associated with each segment is a flag indicating the presence or absence of a connection at the 'top' and 'bottom' of the segment, input as IR and JR, respectively. The connection between segments is specified by IDCO(I,1) and IDCO(I,2), which is the number of the 'top' segment and the adjacent 'bottom' segment, respectively. The compatibility equations expressed in matrix form is

$$[A]\{\Delta\}_t = \{0\} \qquad 4.1$$

where [A] is the Boolean connectivity matrix expressing the compatibility requirements between segment deformations (4); $\{\Delta\}_t$ is the total segment deformation vector.

4.3 Vertical Edge Load

This section demonstrates how the rigid body motion of the shell structure which have been ignored up to this point is taken into account. Loads from the 'top' segment may be transmitted to the segment below it as a vertical edge load P, as shown in Fig. 4.2. Unlike the cylindrical segment which can carry this load by membrane action alone, for the case of the spherical and conical shells, a horizontal force H_v must be added vectorially, so that a resultant force N_ϕ is formed (1,6,14). This horizontal force must be compensated later by subtracting this value from the real horizontal loads PSF(N,1) and PSF(N,3) acting on segment N.

4.4 Shell Eccentricity

Since segments at a joint may not always end at the same point, a horizontal segment eccentricity may be specified in the input data. This results in the eccentricity of the edge horizontal and vertical loads, which in turn produces a moment which must be added to the existing moments PSF(N,2) and PSF(N,4) at the edges of segment N. This moment is automatically calculated in the program.

4.5 The Particular Solution

The particular solution is approximated by the membrane solution. The computation of the membrane in-plane forces N_ϕ and N_θ, for the spherical and conical segments are incorporated into function subprograms FN1, FN2, FN3, and FN4, respectively. The equations used in these subprograms are found in Tables 3.4 and 3.6. The solution for the cylindrical segment, found in Table 3.5, is simple enough, that a separate subroutine is not necessary. The particular solution displacements PSD are obtained from evaluating the equations for Δ_H and Δ_ϕ found at the end of Tables 3.4 to 3.6. These computations are incorporated into subroutines PCYLIN, PDOME, and PCONE, respectively. The particular solutions for the base segment derived in (4) are incorporated into subroutine PBASE.

4.6 The Flexibility Matrix

As derived earlier, the flexibility matrix for a shell of revolution may be expressed as follows

$$[F] = [TA][TT]^{-1}$$

These matrix operation is performed by subroutines CYLIN, DOME, CONE, and BASE, for the cylindrical, spherical, conical, and base segments respectively.

The first step is to initialize the coefficients of [TA] and [TT]. For subroutine CONE, this necessitates the use of another subroutine MMKEL2, which computes the Kelvin functions of order 2 using published recurrence formulas (10). Subroutine MMKEL2 in turn calls up a system-dependent subroutine which evaluates the Kelvin functions of order zero and one and their derivatives. Secondly, a check is made if the segment is inverted or not. By definition, an inverted spherical or conical segment is that which forms a cup-like shape as shown in Figs. (b) of Tables 3.4 and 3.6. If the segment is inverted, subroutine ROWEX is called. This performs row interchanges in the [TA] and [TT] to conform to the inverted configuration. Furthermore, a check is made whether the segment is a closed spherical or conical dome.

If so, the four by four flexibility matrix degenerates into a two by two matrix. The next step is to invert the [TT] matrix which is performed by subroutine TTINV which is capable of inverting a four by four or a degenerated two by two matrix. Finally, the matrix multiplication

$$[TA][TT]^{-1}$$

is performed, thus forming the flexibility matrix.

4.7 Matrix Formulation of the Solution Procedure

Let $[F]_i$ be the flexibility matrix of segment i, then from Eqn. 3.85(b),

$$\{\Delta\}_i = [F]_i\{V\}_i \qquad 4.2$$

Similarly, for the entire structure, the equations are

$$\{\Delta\} = [F]\{V\} \qquad 4.3$$

where the end displacements $\{\Delta\}_i$, end forces $\{V\}_i$, and flexibility matrix $[F]_i$ of element i are assembled into the global matrices $\{\Delta\}$, $\{V\}$, and $[F]$, respectively, in the order consistent with the sequence of segment numbering.

The particular solution displacements and the vertical edge load displacements $\{\delta\}$ in the corresponding order as $\{\Delta\}$. The total displacement vector is

$$\{\Delta\}_t = \{\Delta\} + \{\delta\} \qquad 4.4$$

Substituting Eqn. 4.4 into 4.1,

$$[A](\{\Delta\} + \{\delta\}) = \{0\} \qquad 4.5$$

and multiplying Eqn. 4.3 by [A],

$$[A][F]\{V\} = [A]\{\Delta\} \qquad 4.6$$

Let {q} be a set of relative displacements in terms of the homogeneous solution $\{\Delta\}$ such that

$$\{q\} = [A]\{\Delta\} \qquad 4.7$$

From a general theorem in structural analysis (13), if a set of forces $\{V\}$ is associated with a set of displacements $\{v\}$, and if in another coordinate system, the same set of forces may be described as $\{U\}$, and their associated displacements as $\{u\}$, then the work done in the two systems must be identical when undergoing equivalent displacements, i.e.,

$$\langle u\rangle\{U\} = \langle v\rangle\{V\}, \qquad 4.8$$

similarly,

$$\langle q\rangle\{Q\} = \langle \Delta\rangle\{V\}, \qquad 4.9$$

where $\{V\}$ are the forces associated with displacements $\{\Delta\}$ and $\{Q\}$ are the redundant forces associated with the relative displacements $\{q\}$. Substituting the transpose of Eqn. 4.7 into 4.9,

$$\langle \Delta\rangle[A]^T\{Q\} = \langle \Delta\rangle\{V\} \qquad 4.10(a)$$

$$\langle \Delta\rangle([A]^T\{Q\} - \{V\}) = 0 \qquad 4.10(b)$$

since Eqn. 4.9 must be true for all $\langle \Delta\rangle$, Eqn. 4.10(b) becomes

$$\{V\} = [A]^T\{Q\}, \qquad 4.11$$

Substituting Eqn. 4.11 into 4.6 yields

$$[A][F][A]^T\{Q\} = -[A]\{\delta\} \qquad 4.12(a)$$

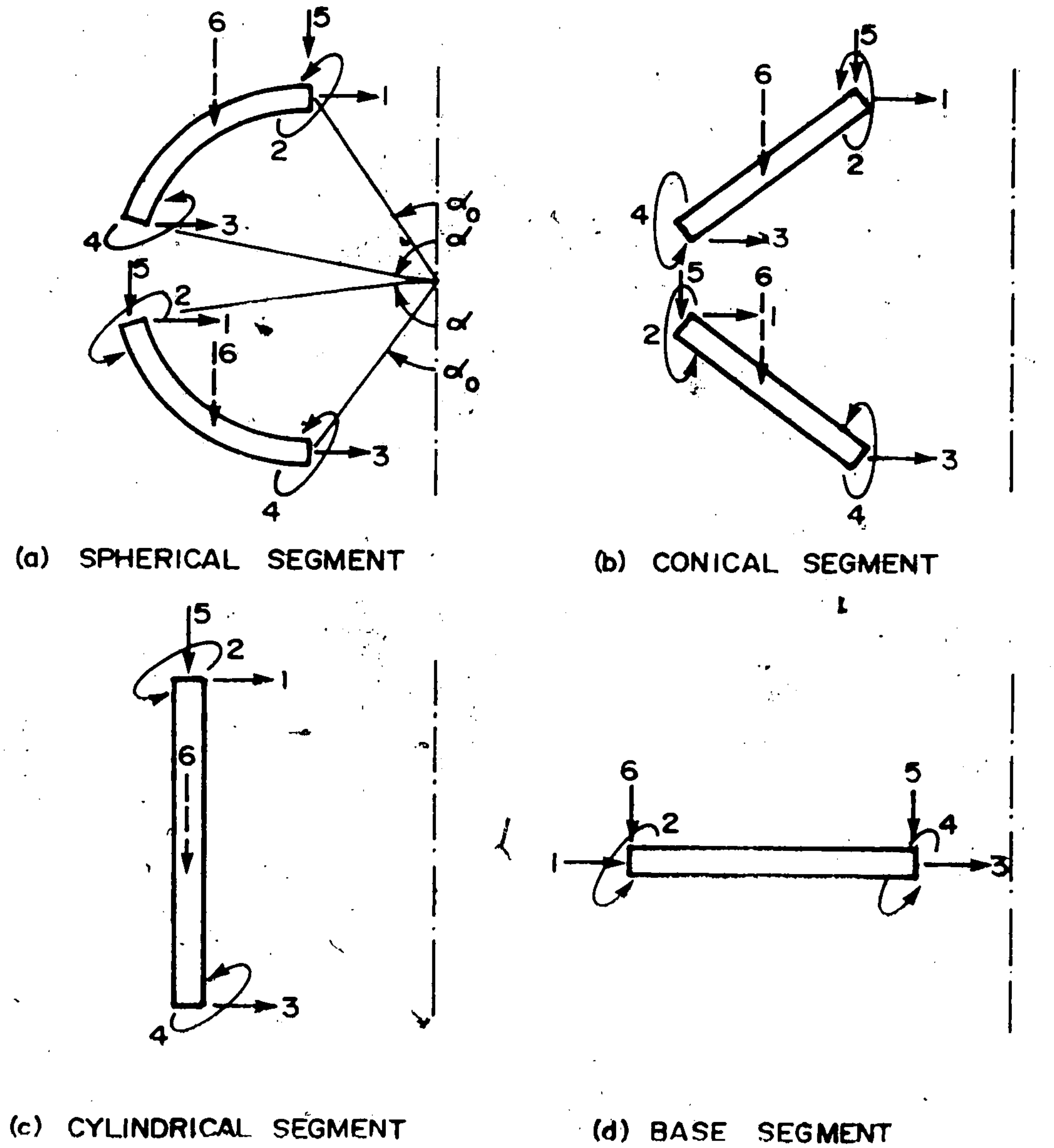
$$[\bar{F}]\{Q\} = \{q_o\} \qquad 4.12(b)$$

where the structure flexibility matrix and structure particular solution displacements, respectively, are

$$[\bar{F}] = [A][F][A]^T \qquad 4.13$$

$$\{q_0\} = -[A]\{\delta\} \qquad 4.14$$

The set of simultaneous equations (Eqns. 4.12(b)), can then be solved for the redundants Q, by Gauss elimination. Once evaluated, the value of the redundants may be back-substituted into Eqn. 4.11, to find the edge forces V, which in turn can be substituted into Eqn. 4.3, to find the displacements Δ. The final solution can then be obtained by superimposing these values on the particular solution.



ARRAYS: P B F — PARTICULAR SOLUTION BASE FORCES
P S D — PARTICULAR SOLUTION DISPLACEMENTS
P S F — PARTICULAR SOLUTION SEGMENT FORCES
S F — SEGMENT FORCES

Figure 4.1 SUBSCRIPTING OF SEGMENT ARRAYS FOR FLEXSHELL

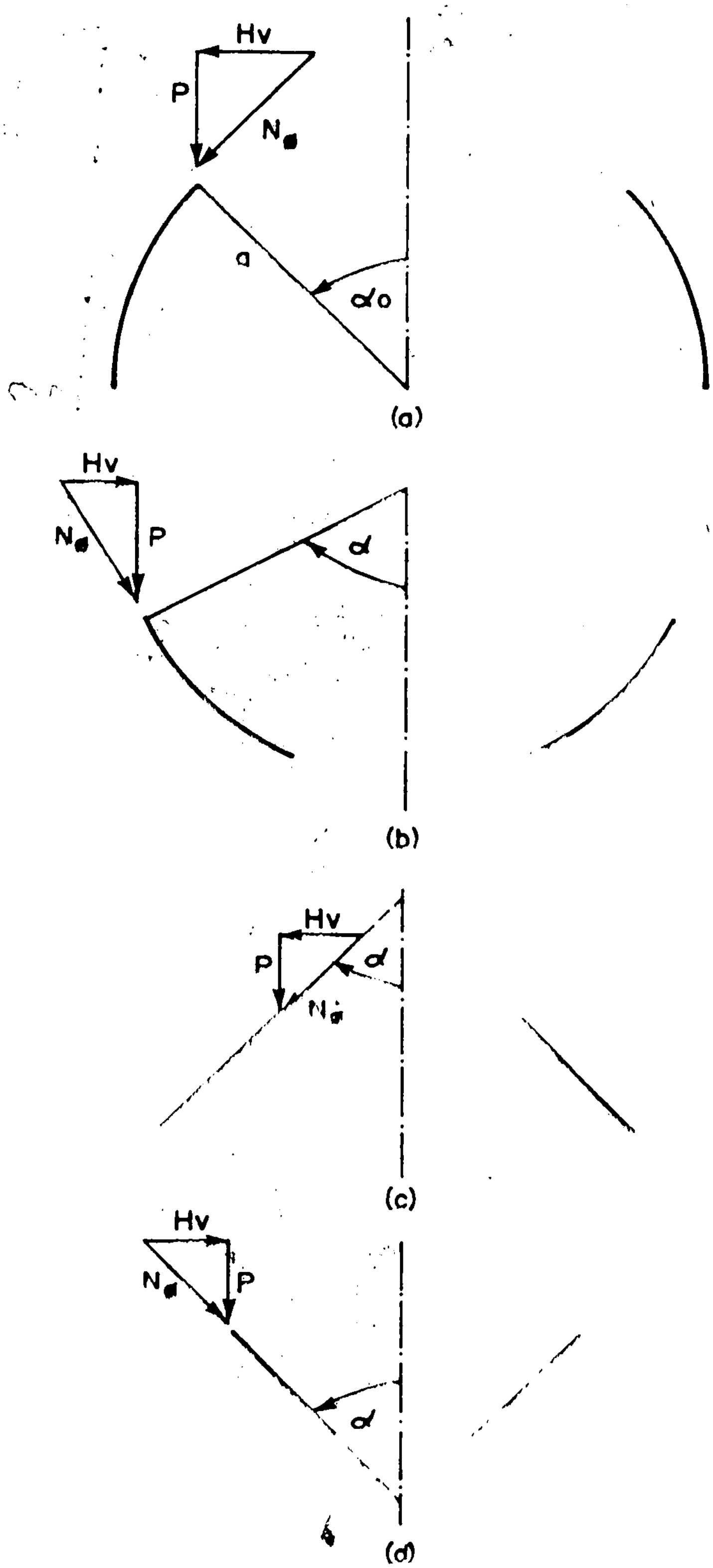


Figure 4.2 VERTICAL EDGE LOAD EFFECT

5. EVALUATION OF THE FLEXIBILITY APPROACH

The flexibility and stiffness methods are two basic approaches in the analysis of statically indeterminate structures. For a given problem, both methods will give identical solutions. Any differences which may be observed are due to the approximations used in the formulation and are not inherent in the methods themselves. Basic limitations such as axisymmetry of the loads and geometry, if introduced, will limit the scope of the application of each method.

The simplifications used in formulating the equations on which program SASHELL is based are consistent with the elastic theory. Hence, any errors in the solution are due to the manner in which the equations are solved in SASHELL. These may result from the lack of stability of the numerical integration technique used, and, for nonaxisymmetric loading, the rate of convergence of the Fourier expansions could be a factor. It should also be noted that the elastic theory is based on an element that has two edge boundaries along the meridian. So in the formulation of the stiffness matrix, four boundary conditions must be imposed at each edge of the shell in order to evaluate the eight constants of integration. Thus, the closed, spherical or conical segment which has only one edge results in a singular stiffness matrix. This problem may be overcome with the introduction of a small hole at the apex, say [illegible]

On the other hand, the solutions obtained from program FLEXSHELL are based on closed form solutions. The degree of simplifications introduced in obtaining these solutions depend on the shell type. With cylindrical shells, no simplifications are necessary. Hence, the solution should be identical to that from program SASHELL. With conical shells, some error may be introduced depending on the semi-vertex angle, α due to difficulties in evaluating the Kelvin functions and their derivatives. Using Geckeler's approximation in obtaining the closed form solution for the spherical shell will likely result in greater error, particularly as the values of [illegible] or α become small.

Closed form solutions for each segment were obtained for 'short' segments, that is, two constants of integration are evaluated at each edge of the shell. However, unlike program SASHELL the analysis of closed, spherical or conical segments offers no difficulties. In such case, only the constants of integration corresponding to the end furthest from the apex are evaluated. Thus, imposing [illegible] conditions at the apex are not required.

In order to evaluate the reliability of the two methods used in the programs, both programs were run on identical problems and the results are compared. Comments as to which is considered [illegible] for a specific problem are [illegible]

5.1 Cylindrical Segment

Fig. 5.1 illustrates the distribution of the bending moment and circumferential force along the length of the cylinder, under a hydrostatic pressure with γ = 62.4 pcf, obtained from programs FLEXSHELL and SASHELL. The meridional force N_s is equal to zero in this case. As expected, the two solutions are identical. Similarly, upon investigating the solutions for the other load cases, dead load, snow load, and uniform pressure or prestressing, both programs yield identical results.

5.2 Conical Segment

There are physical limits that must be imposed on the semi-vertex angle α for conical segments. As α approaches zero, the cone degenerates into a line and no shell action is possible. On the other hand, as α approaches 90°, the cone becomes a circular plate.

With the formulation of SASHELL, the latter case presents no problem since the basic shell equations reduce to the plate equation when α = 90°. However, in FLEXSHELL, it is necessary to impose a limit on the range of values of ξ for which the Kelvin functions and their derivatives are evaluated,

$$\xi = 2\lambda\sqrt{s}$$

where
$$\lambda^4 = \frac{12(1-\nu^2)}{h^2\tan^2\alpha}$$

So when α = 90°, ξ approaches zero and when α = 0°, ξ approaches ∞. Thus, this dimensionless parameter used in the evaluation of the Kelvin functions and their derivatives are limited as follows:

$$0 < \xi \leq 119.0$$

Consequently, a limit on the range of values of α is imposed. For instance, when s/h = 500, α must be greater than 26°, and when s/h = 100, α must be greater than 5.5°. Note that α can be very close to, but not equal to 90°, say 89.5°.

Fig. 5.2 illustrates the distribution of the in-plane forces N_ϕ and N_θ and the meridional bending moment M_ϕ along the conical segment, under snow load, p = 1 ksf, according to programs FLEXSHELL and SASHELL. As anticipated, both programs yield identical results since no simplifications were made in the formulation of the closed form solution for this segment. The solutions for the other load cases are also identical.

5.3 Spherical Segment

Fig. 5.3 illustrates the distribution of the in-plane forces N_ϕ and N_θ and the meridional bending moment M_ϕ along the spherical segment for α = 10° and α = 80°, under dead load with γ = 150 pcf, according to programs FLEXSHELL and SASHELL. It is observed that there is a greater discrepancy between the solutions for small values of the angle α, say 10°, than for large values of α, say 80°. This observation is confirmed with the investigation of the solutions for several values of α with α_0 = 0°, as shown in Fig. 5.4. The solutions for different values of a/h were also compared, and a greater discrepancy is observed for small a/h values. These observations may be explained as follows. As the angle α become small, the bending effects become more significant over a large portion of the segment, Clearly, this violates the basis of Geckeler's assumption; hence, the approximation for the spherical segment becomes less accurate, For the same reason, a greater discrepancy is observed for small values of $(\alpha-\alpha_0)$ with $\alpha_0 \neq 0°$, as illustrated in Fig. 5.5. Note that no discrepancies are observed for $\alpha = 90°$. Since Geckeler's assumption requires that the dimensionless parameter λ, which appears in the closed form solution for the spherical segment, must be large, that is, a/h is large, the approximation becomes less accurate for small values of a/h. In order to be able to predict the discrepancy between the two solutions for a specific set of geometry and material property, Figs. 5.4 and 5.5 were combined to develop Figs. 5.6 to 5.8 for various load cases.

Fig. 5.6 compares the meridional forces in a spherical segment under various load cases as given by the two computer programs. As illustrated in the figure, the solutions obtained from FLEXSHELL show excellent agreement with those obtained from SASHELL, the maximum difference being 1/200 of 1 percent. For the liquid pressure loading, the solutions become identical as α approaches 90°, and smaller discrepancies ar observed with increasing values of $\lambda(\alpha-\alpha_0)$. In general, for any α with $\lambda(\alpha-\alpha_0) = 15$, the solutions differ by less than 5 percent.

Fig. 5.7 compares the circumferential forces on a spherical segment under various loads. Similarly, the solutions become identical as α approaches 90°, and smaller discrepancies are observed for high values of $\lambda(\alpha-\alpha_0)$. The discrepancies between the solutions for $\lambda(\alpha-\alpha_0) = 10$ due to dead load and snow load and due to uniform pressure are less than 5 and 10 percent respectively. For the liquid pressure, a greater discrepancy is observed, up to 10 and 5 percent discrepancy is observed for $\lambda(\alpha-\alpha_0) = 43$ and $\lambda(\alpha-\alpha_0) = 60$ respectively.

Fig. 5.8 compares the meridional bending moment in the spherical segment under various loads. As with the in-plane forces, the bending moments from both programs become identical as α approaches 90°, and smaller discrepancies are observed with increasing values of $\lambda(\alpha-\alpha_0)$. A discrepancy of less than 5 percent for the uniform pressure, dead load, and snow load is observed. However, a discrepancy of less than 10 percent is observed for any α with $\lambda(\alpha-\alpha_0) = 25$, and 5 percent for $\lambda(\alpha-\alpha_0) = 45$.

5.4 Application of Program FLEXSHELL

Having investigated program FLEXSHELL for simple shell segments, a multi-shell structure consisting of a combination of cylindrical, spherical, and conical segments was analyzed to demonstrate the capabilities of the program. Again, the results obtained from FLEXSHELL are compared with those from SASHELL. The same example problem is used in the FLEXSHELL user's manual (Appendix C) to illustrate the input and output files. Fig. 5.9 illustrates an Intze tank consisting of cylindrical, spherical, conical and base segments, and ring beams which are modelled as cylindrical segments. Intze tanks are mainly used for water storage, and they are typically constructed as prestressed concrete. The material properties which were used are:

$$E = 0.5804 \times 10^{9} \text{ psf}$$

$$\nu = 0.167$$

$$\gamma = 150.0 \text{ pcf}$$

The base of the tank is considered to be fully fixed so the base segment was given a high modulus of elasticity,

$$E(\text{base}) = 1.0 \times 10^{20} \text{ psf}$$

The results of the analysis of the Intze tank under dead load, given by both programs are shown graphically in Figs. 5.10 to 5.12. It is apparent from these figures that the solutions show excellent agreement between the in-plane forces N_ϕ and N_θ, and meridional bending moment M_ϕ, for each segment. Some discrepancy is observed in the solution for both spherical segments in the vicinity of the apex. This may be introduced by the singularity formed at this location in program SASHELL, or due to inaccuracies introduced by using Geckeler's assumption for small values of the angle α in FLEXSHELL. Nevertheless, both solutions show good correlation and the same is anticipated for the other load cases.

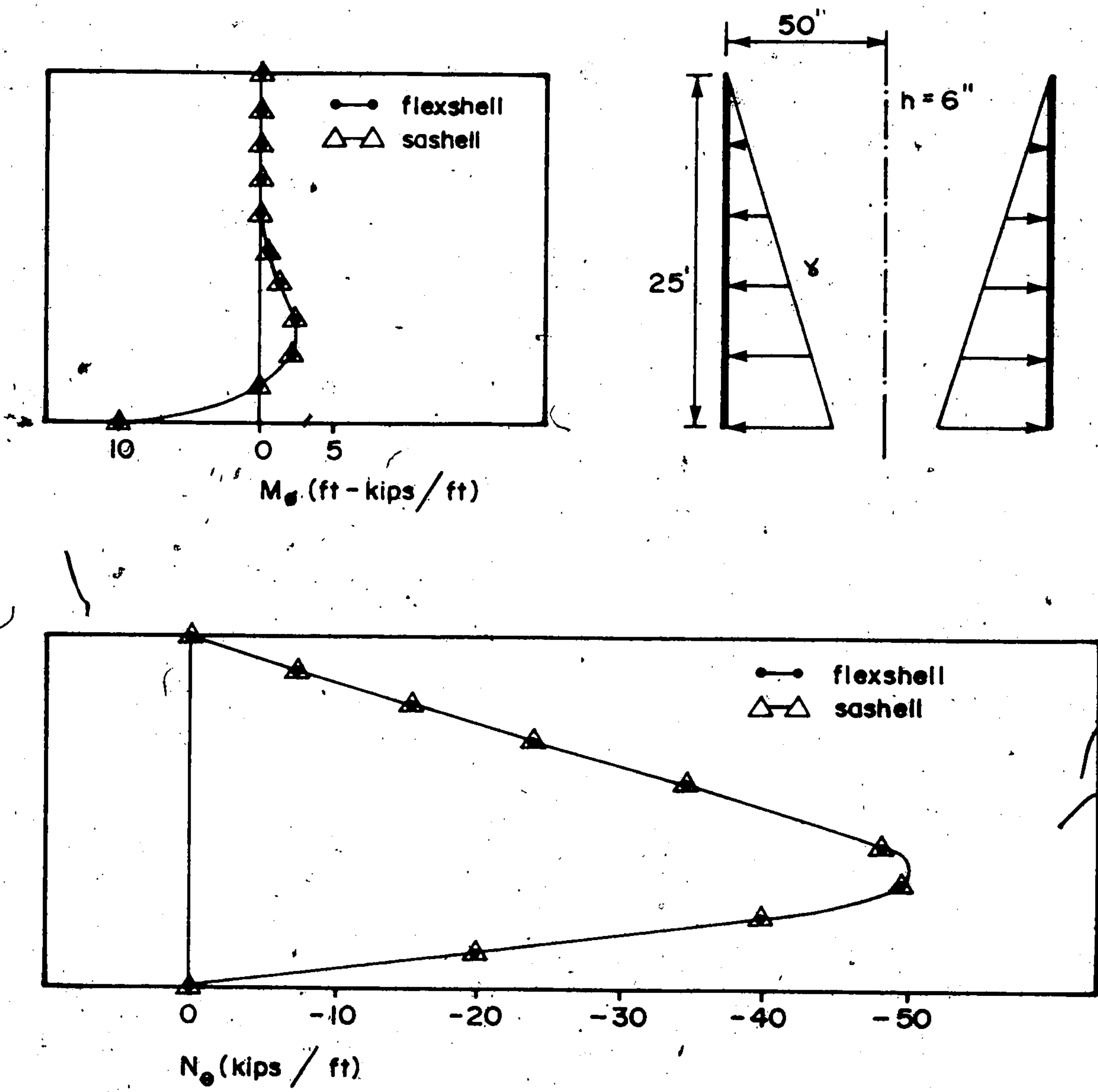


Figure 5.1 CIRCUMFERENTIAL FORCE & MOMENT ALONG A CYLINRICAL SEGMENT UNDER HYDROSTATIC LOAD, γ = 62.4 pcf.

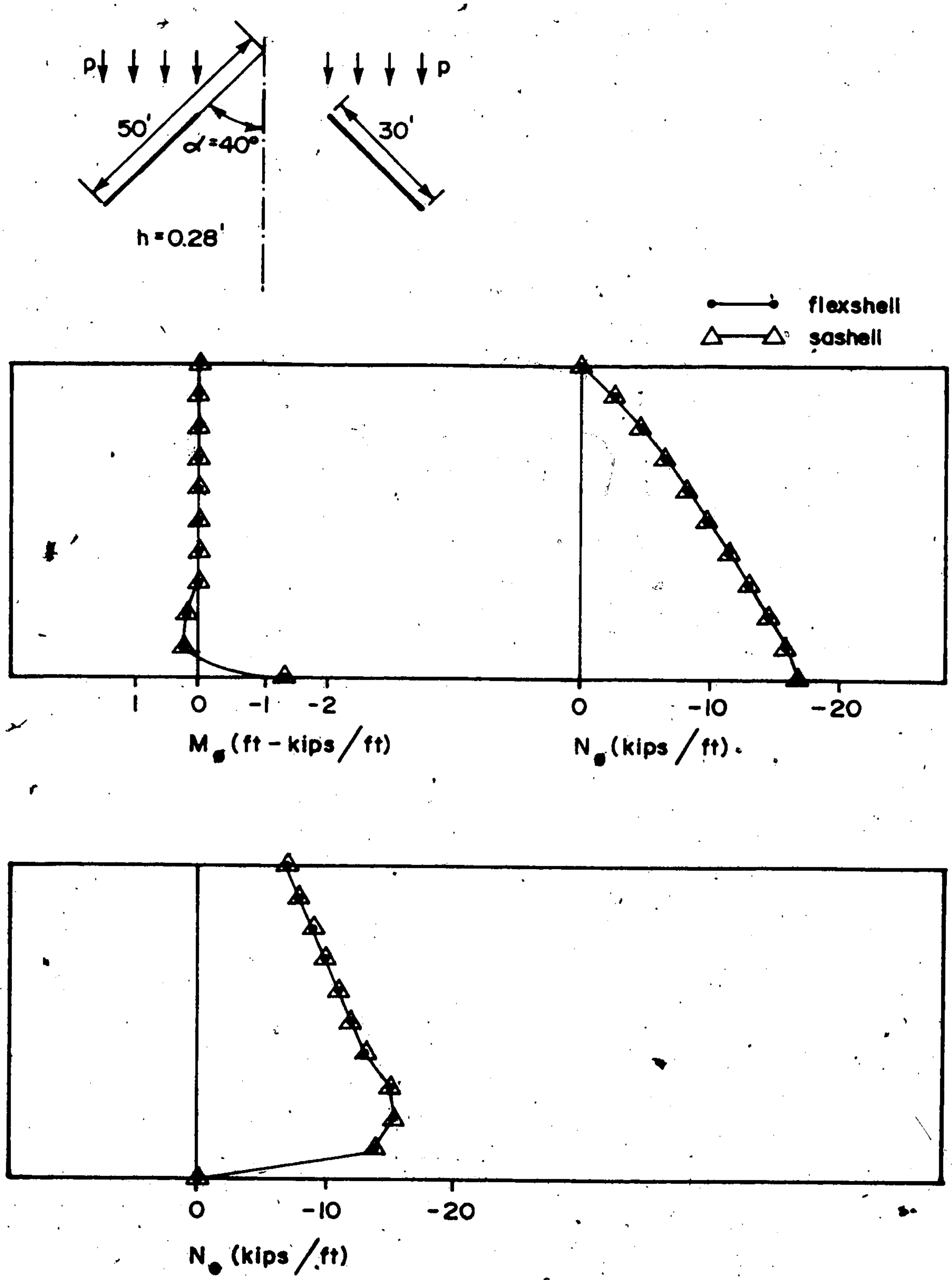


Figure 5.2 IN - PLANE FORCES & MOMENTS ALONG A CONICAL SEGMENT UNDER SNOWLOAD, p = 1 ksf.

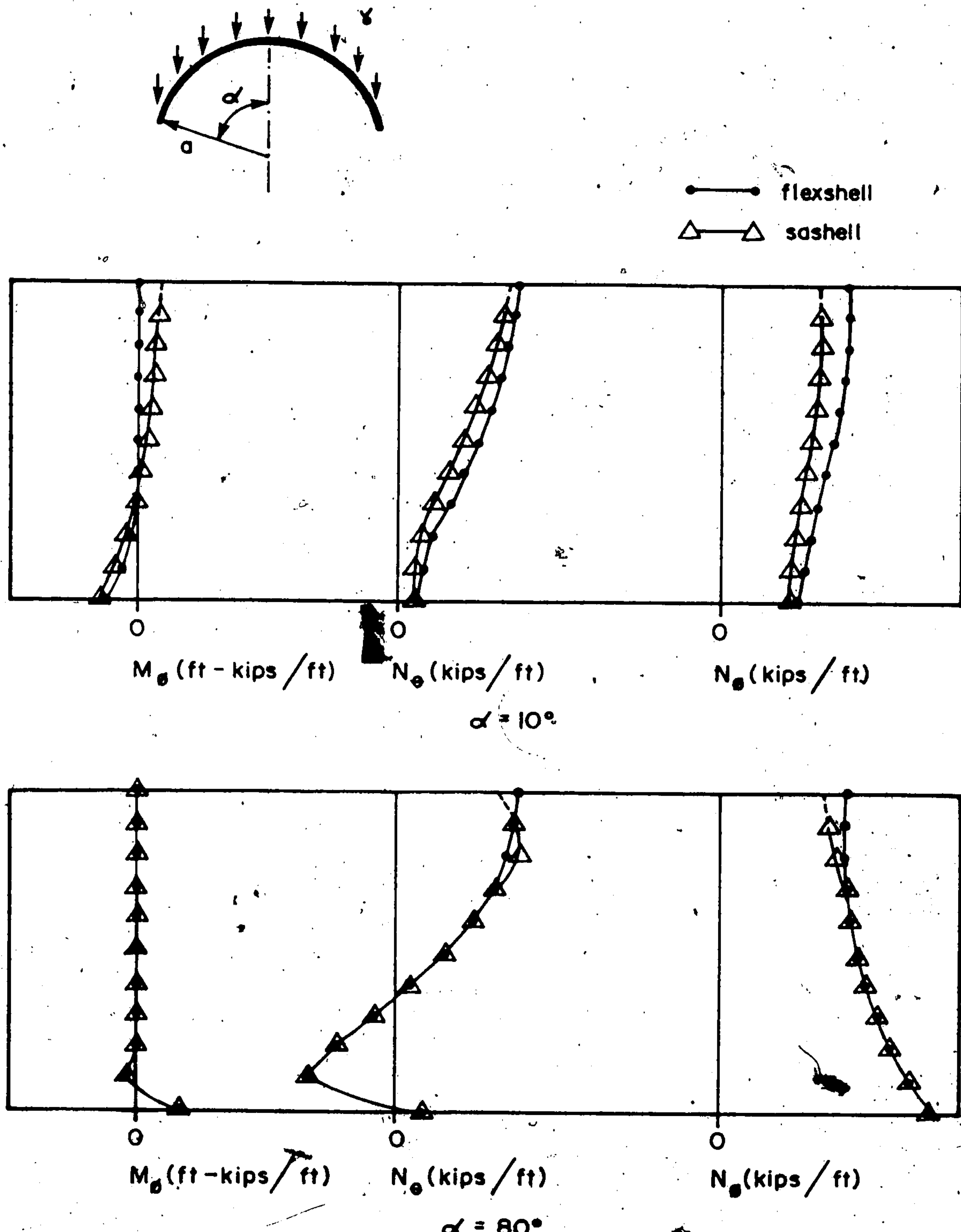


Figure 5.3 IN - PLANE FORCES & MOMENTS ALONG A SPHERICAL SEGMENT UNDER A DEADLOAD, $\gamma = 150$ pcf

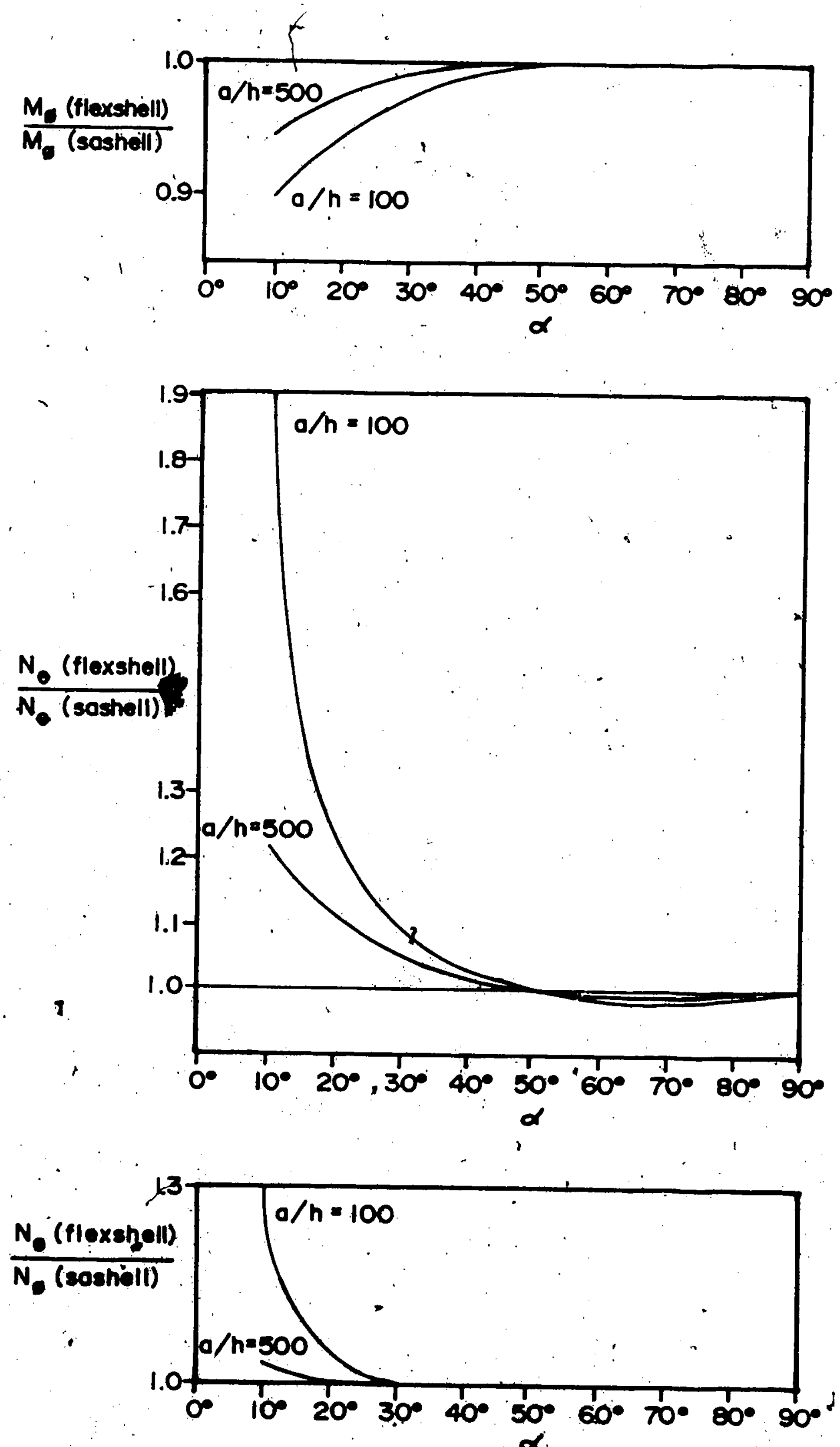


Figure 5.4 COMPARISON OF THE STRESS RESULTANTS FOR THE SPHERICAL SEGMENT IN TERMS OF a/h

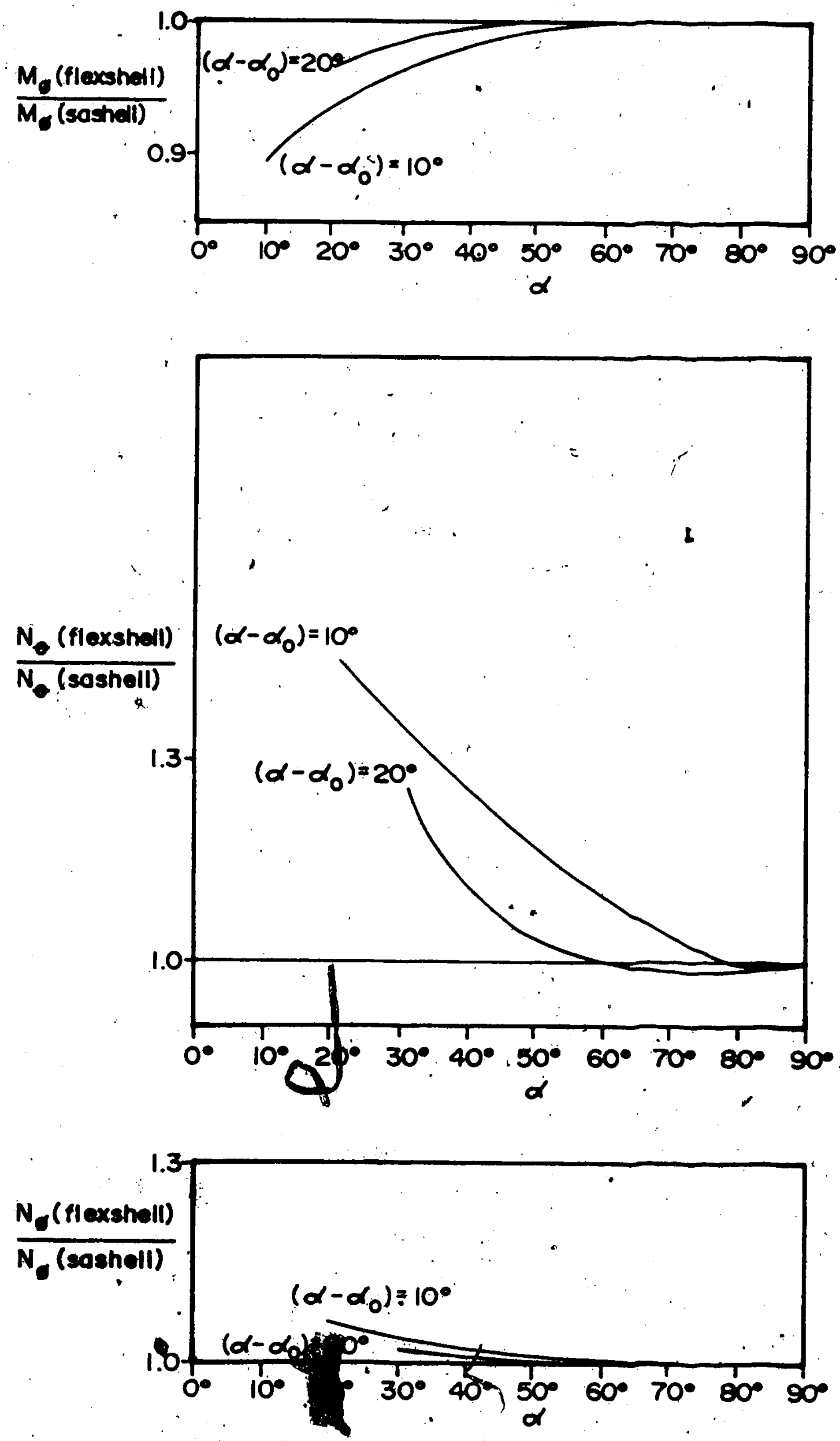


Figure 5.5 COMPARISON OF THE STRESS RESULTANTS FOR THE SPHERICAL SEGMENT IN TERMS OF $(\alpha - \alpha_0)$

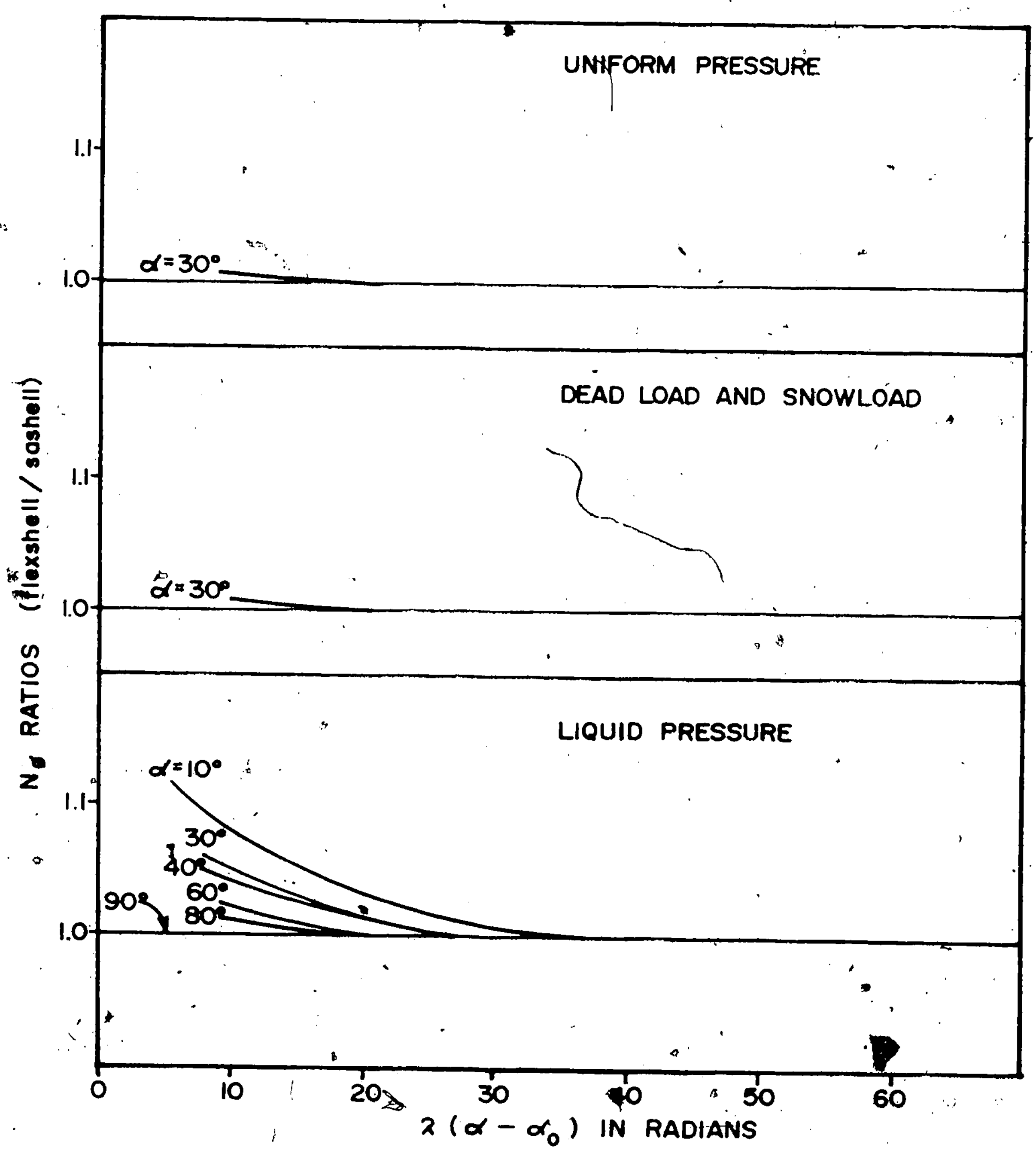


Figure 5.6 COMPARISON OF THE MERIDIONAL FORCE, N_ϕ FOR THE SPHERICAL SEGMENT UNDER VARIOUS LOADS

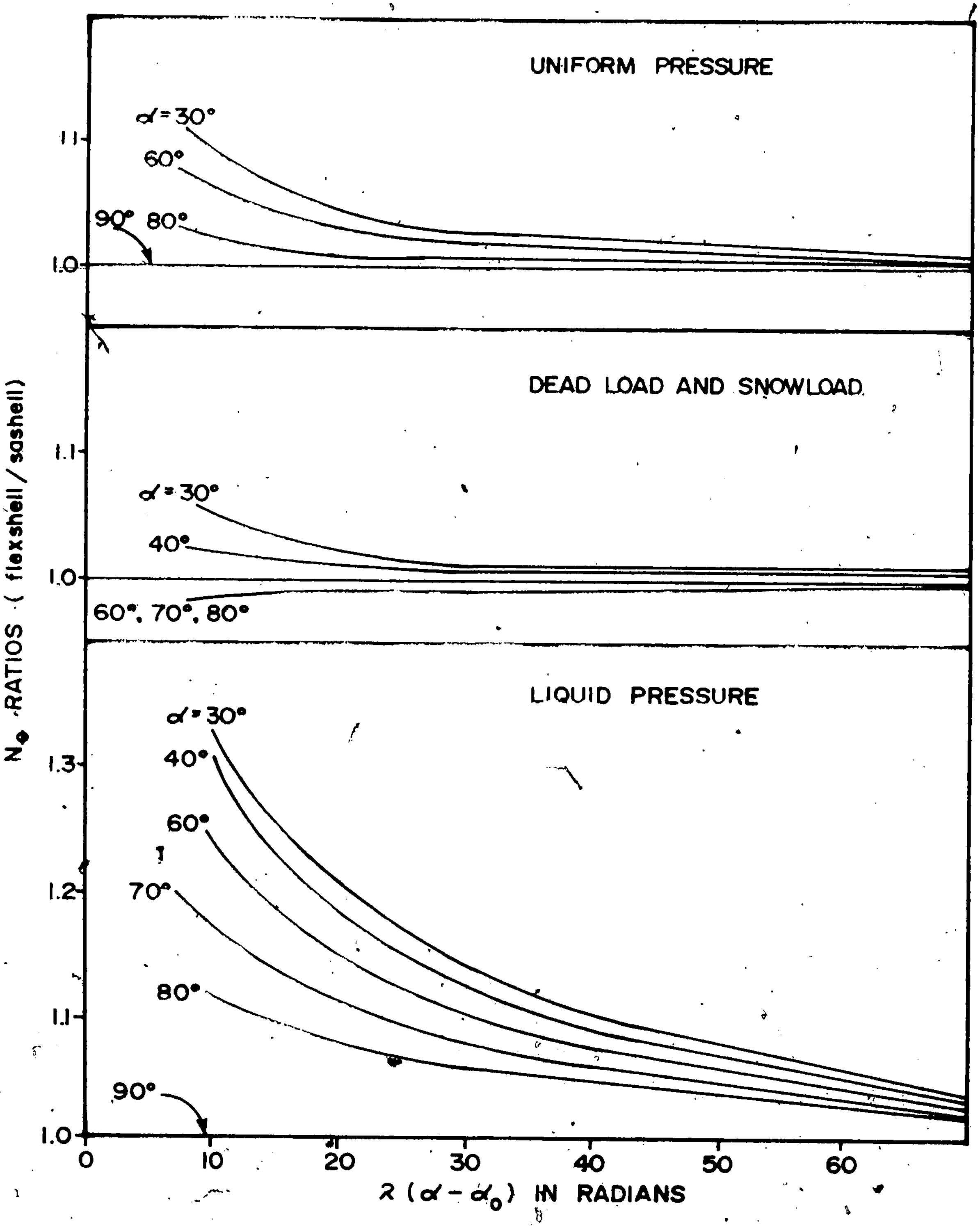


Figure 5.7 COMPARISON OF THE CIRCUMFERENTIAL FORCE, N_ϕ FOR THE SPHERICAL SEGMENT UNDER VARIOUS LOADS

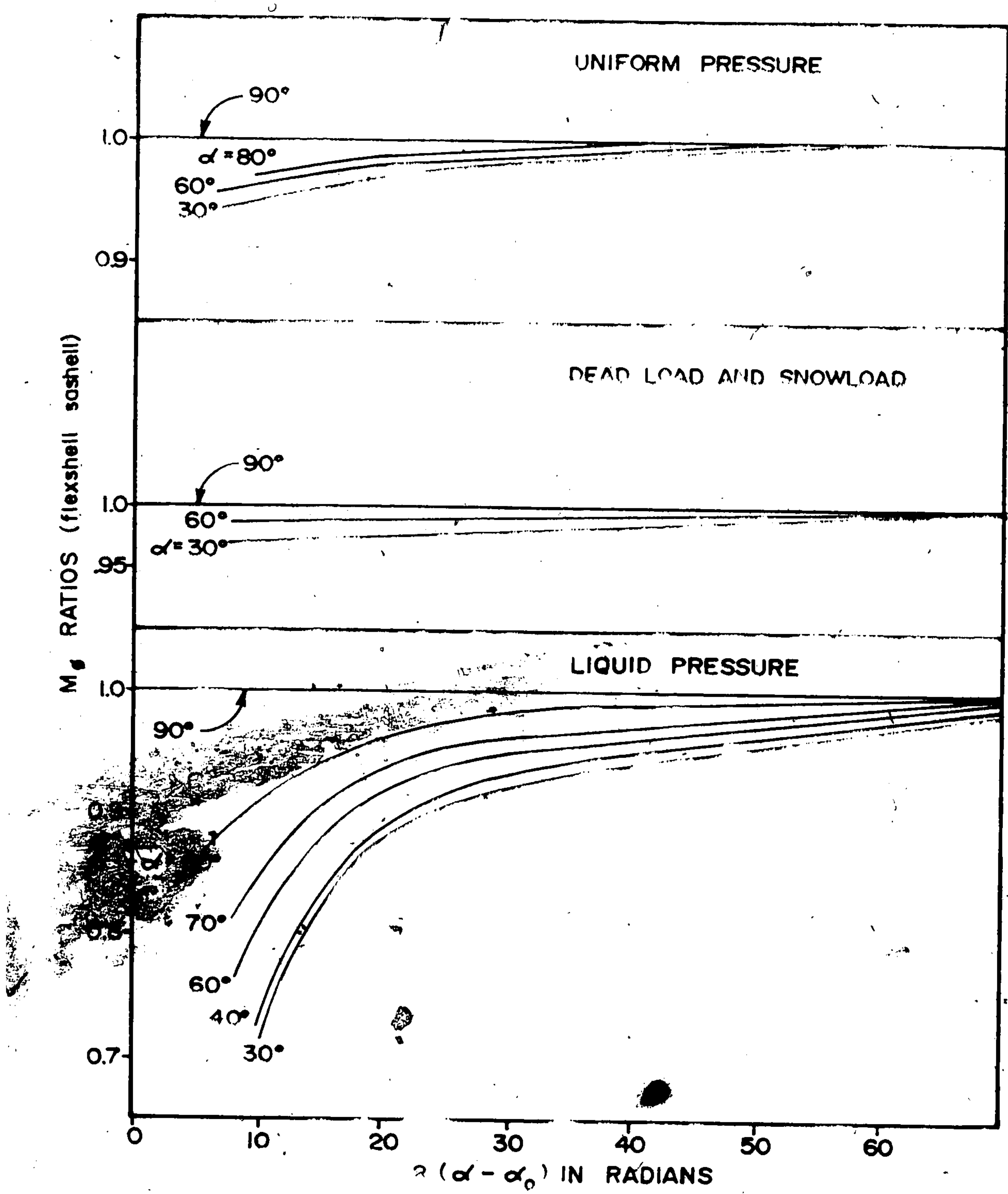


Figure 5.8 COMPARISON OF THE BENDING MOMENT, M_ϕ FOR THE SPHERICAL SEGMENT UNDER VARIOUS LOADS

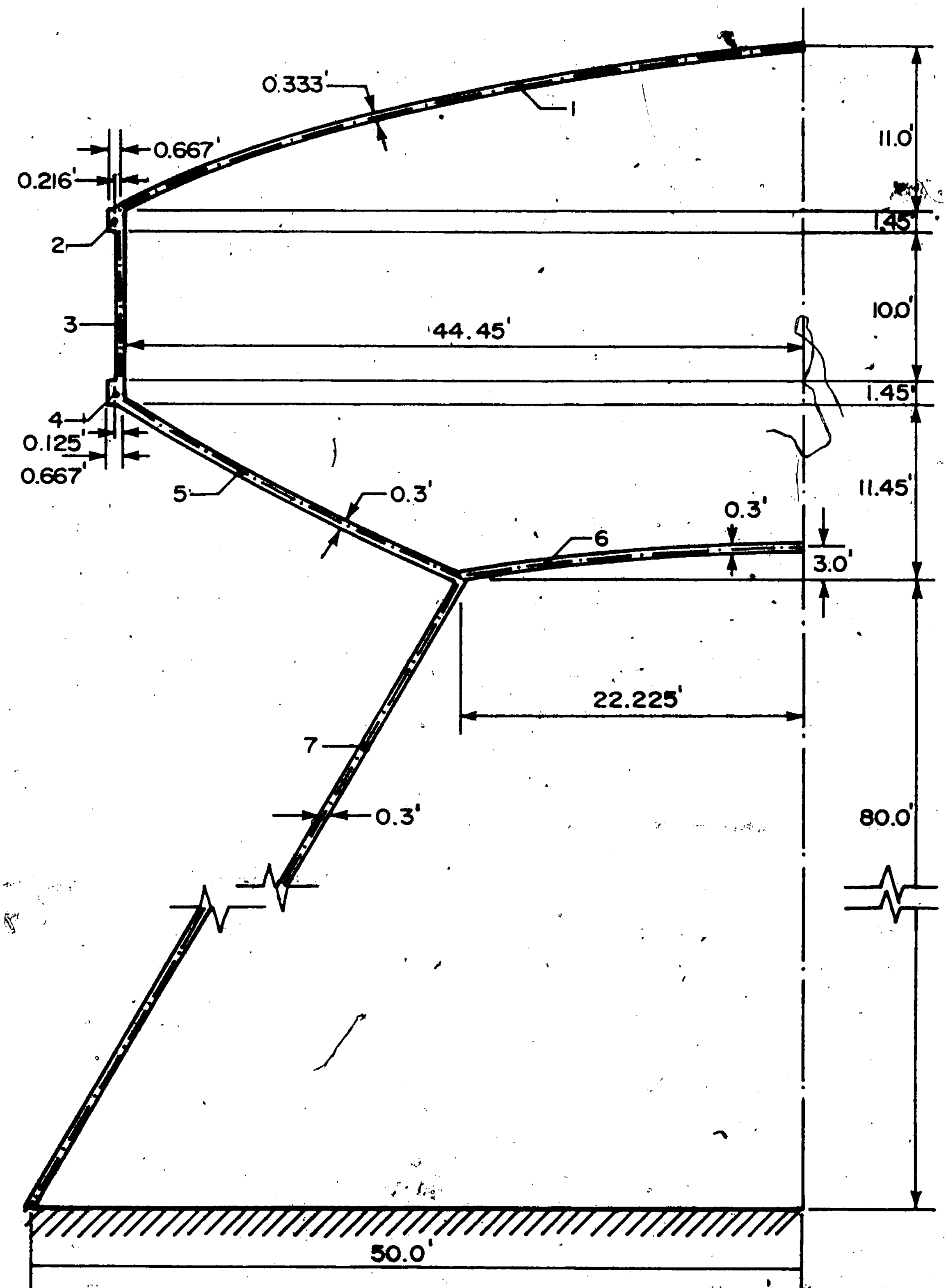


Figure 5.9 INTZE TANK MODEL FOR FLEXSHELL

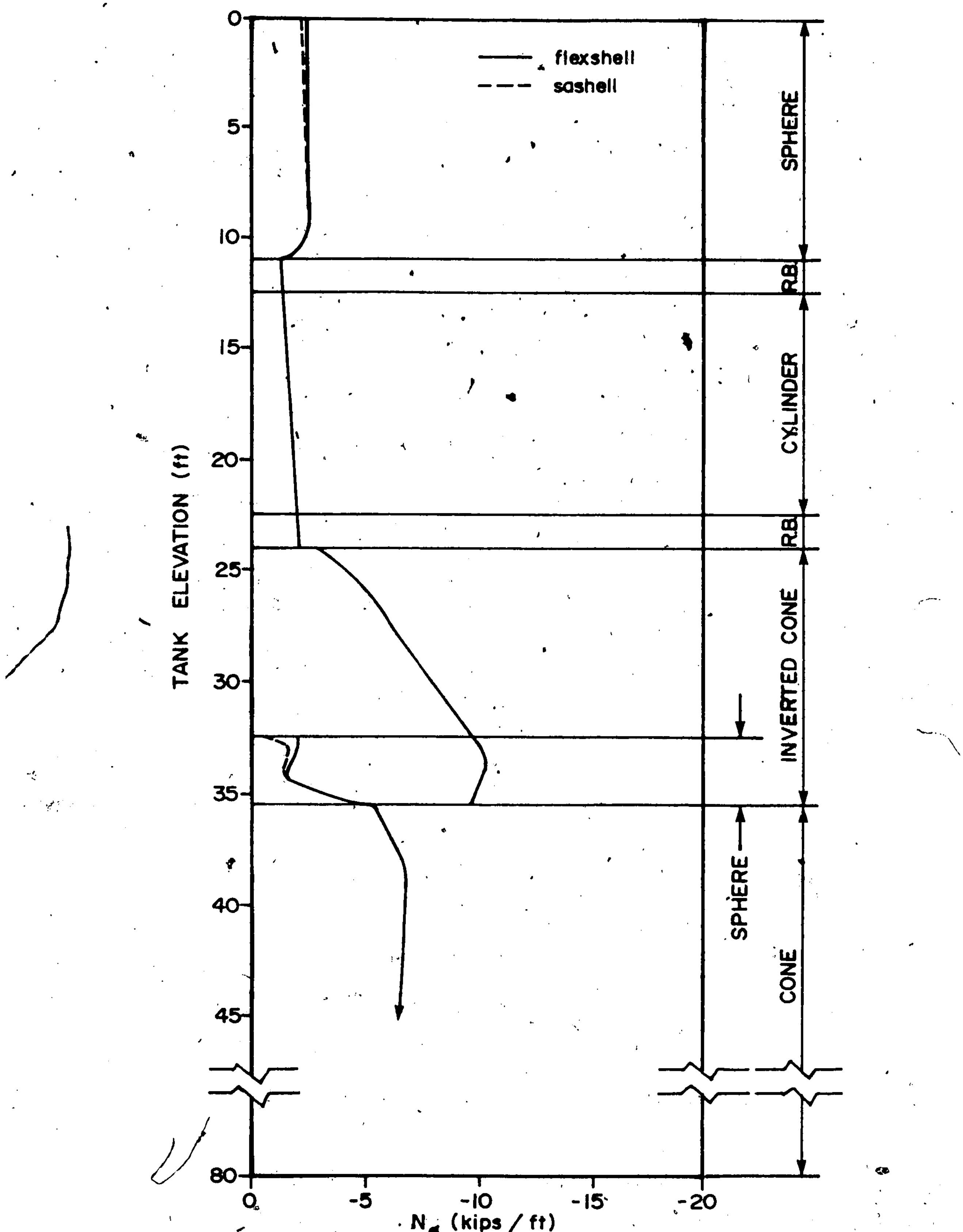


Figure 5.10 INTZE TANK PROBLEM (N_ϕ COMPARISON)

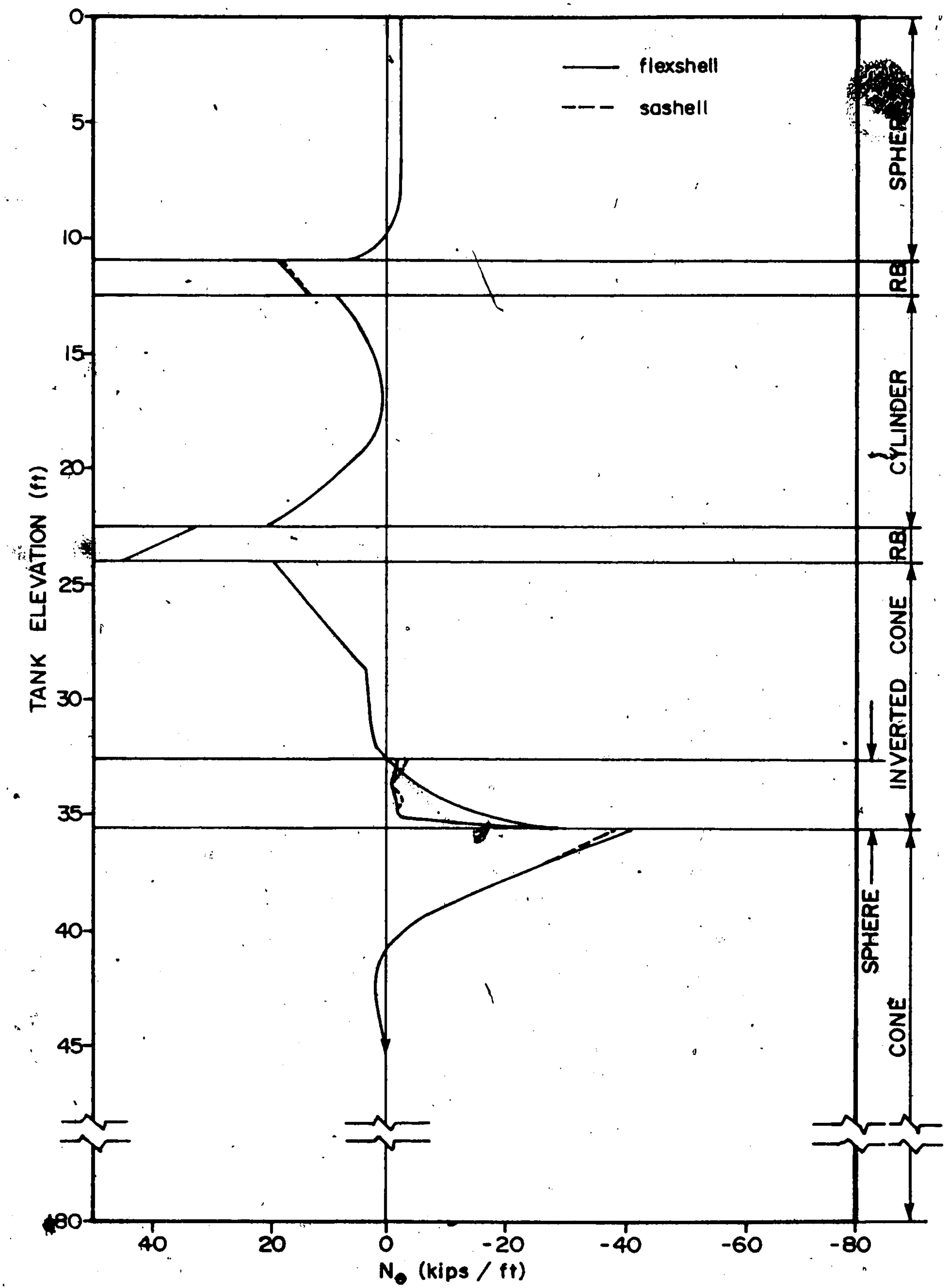


Figure 5.11 INTZE TANK PROBLEM (N_θ COMPARISON)

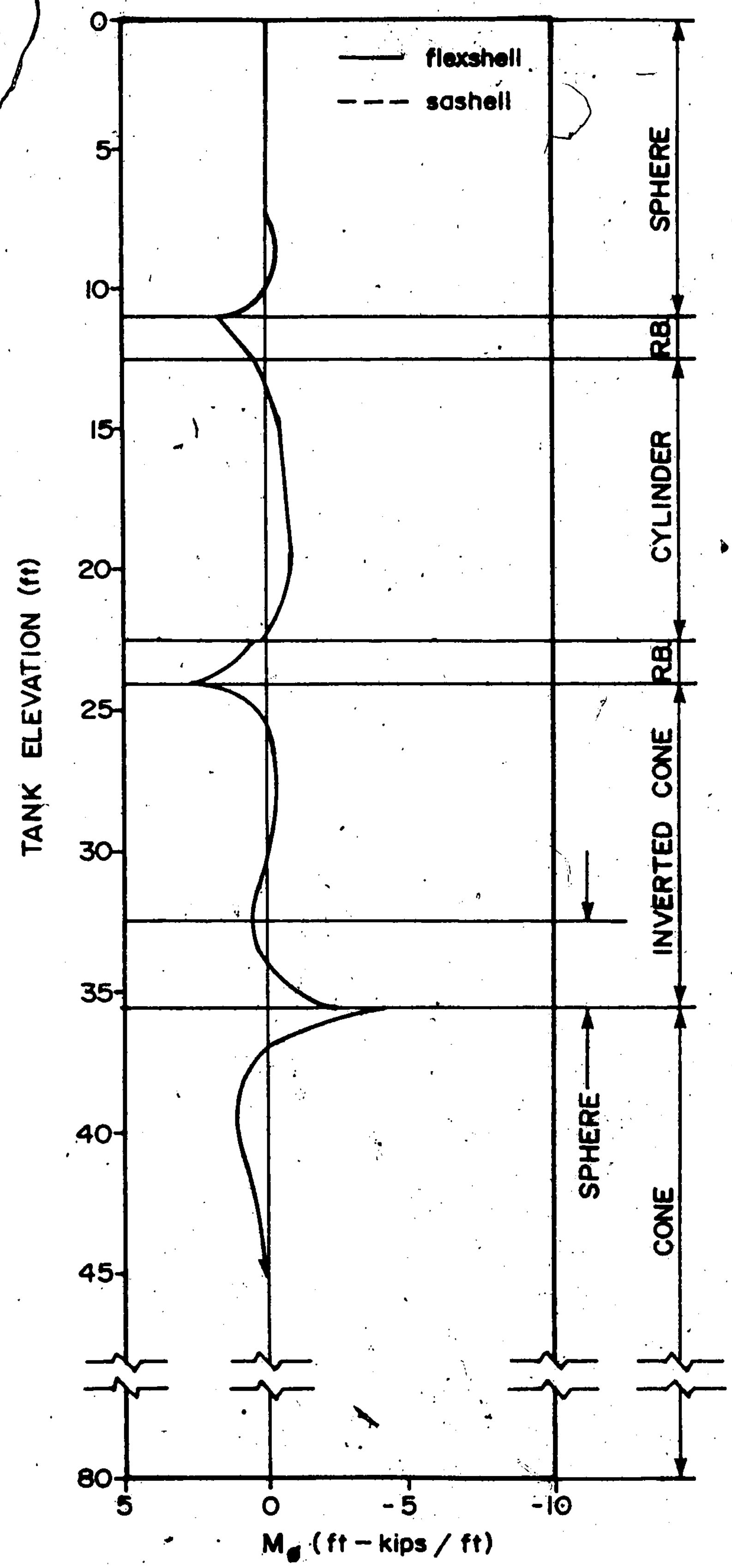


Figure 5.12 INTZE TANK PROBLEM (M_ϕ COMPARISONS)

6. SUMMARY AND CONCLUSIONS

The two basic approaches in the elastic analysis of multi-shell structures were discussed. With the stiffness method, the stiffness matrix relating the forces and deformations at the edge of each shell segment are formed by numerically integrating the basic shell equations which were expanded into a Fourier series. These element stiffness matrices are then superposed to form the structural stiffness matrix from which the segment edge deformations are computed. With the flexibility method, the flexibility matrix is formed from the closed form solutions derived in Chapter 3. The particular solution is approximated by the membrane solution also derived in Chapter 3 and shown on Tables 3.4 to 3.6. A computer program, FLEXSHELL, was developed based on the flexibility approach. The results are then compared with the results from program SASHELL developed by Shazly (5) based on the stiffness approach.

Individual segments under various load cases were investigated. It was found that both programs yield identical solutions for the cylindrical and conical segments. Since Geckeler's assumption was used in the formulation of the solution for the spherical segment in FLEXSHELL, a discrepancy between the solutions was anticipated and observed to be a function of $\lambda(\alpha-\alpha_0)$ and the angle α, where λ is the dimensionless parameter in terms of a/h.

The Intze tank problem discussed in Chapter 5 was selected to demonstrate the capabilities of program FLEXSHELL to analyze an axisymmetric segmented shell structure. Overall, program FLEXSHELL showed excellent agreement with SASHELL. The main advantage to using FLEXSHELL is that input is simple.

Therefore, it may be concluded that FLEXSHELL is a simple, effective tool for the analysis of a wide variety of axisymmetric multi-shell structures.

Further development is possible, with the addition of more load cases and more shell types.

REFERENCES

1. Flügge, W., "Stresses in Shells", Springer-Verlag OHG., Berlin, 1960.

2. Billington, David P., "Thin Shell Concrete Structures", McGraw-Hill Book Company, Inc., New York, 1965.

3. Billington, David P., "Thin Shell Concrete Structures", 2nd Edition, McGraw-Hill Book Company, Inc., New York, 1982.

4. Murray, D.W., Rohardt, A.M., Simmonds S.H., "A Classical Flexibility Analysis for Gentilly-type Containment Structures", Structural Engineering Report No. 63, The University of Alberta, Edmonton, June 1977.

5. Shazly, A.M., Simmonds, S.H., Murray, D.W., "Numerical Analysis of General Shells of Revolution Subjected to Arbitrary Loading", Structures Report No. 69, The University of Alberta, Edmonton, September 1978.

6. Timoshenko, S., Woinowsky-Krieger, S., "Theory of Plates and Shells", 2nd Edition, McGraw-Hill Book Company, Inc., New York, 1959.

7. Tokarski, E.W., "Analysis of Symmetrically-loaded Circular Conical Shells", Concrete and Constructional Engineering, February 1965, March 1965, April 1965.

8. Baltrukonis, J.H., "Influence Coefficients for Edge-loaded Short, Thin, Conical Frustums", Journal of Applied Mechanics, ASME Transactions, June 1959, 241-245.

9. Kraus, H., "Thin Elastic Shells", John Wiley & Sons, Inc., New York, 1967.

10. Abramowitz, M., Stegun, I., "AMS 55 Handbook of Mathematical Functions", Government Printing Office, Washington, D.C., 1964, 379.

11. Pflüger, A., "Elementare Schalenstatik" (Elementary Shell Analysis), 4th Edition, Springer-Verlag, Berlin-New York, 1967.

12. Pflüger, A., "Elementary Statics of Shells", 2nd Edition, F.W. Dodge Corporation, New York, 1961.

13. White, R.N., Gergely, P., Sexsmith, R.G., "Structural Engineering", John Wiley and Sons, Inc., New York, 1972.

14. Girkmann, K., "Flächentragwerke", Springer-Verlag,

Berlin, 1963.

APPENDIX A-DETAILED DERIVATION OF THE HOMOGENEOUS SOLUTION

For a cylindrical segment with geometrical properties as follows,

$r_0 = r_2 = r \qquad N_\phi = N_s \qquad \frac{d}{d\phi} = r_1 \frac{d}{ds}$
$r_1 = \infty \qquad p\phi = p_s$
$\phi = \pi/2$

Eqns. 3.64 and 3.67 derived earlier become

$$r_2 \frac{d^2U}{ds^2} = EtV \qquad \text{A.1}$$

$$r_2 \frac{d^2V}{ds^2} = -\frac{U}{D} \qquad \text{A.2}$$

Combining these equations and using eqn. 3.57 to eliminate U,

$$\frac{d^4\Delta_H}{ds^4} + 4\beta^4\Delta_H = 0 \qquad \text{A.3}$$

where

$$\beta^4 = \frac{3(1-\nu^2)}{r^2h^2} \qquad \text{A.4}$$

for which the final solution is

$$\Delta_H = e^{\beta s}(C_1\cos\beta s + C_2\sin\beta s) + e^{-\beta s}(C_3\cos\beta s + C_4\sin\beta s) \qquad \text{A.5}$$

The geometrical properties of a conical segment is

$r_0 = s\ \sin\alpha \qquad \phi = \pi/2 - \alpha \qquad \frac{d}{d\phi} = r_1 \frac{d}{ds}$
$r_1 = \infty \qquad N_\phi = N_s$
$r_2 = s\ \tan\alpha \qquad p\phi = p_s$

Substituting these relations into Eqns. 3.64 and 3.67 yield

$$r_2 \frac{d^2U}{ds^2} + \frac{dU}{ds}\tan\alpha - \frac{U}{r_2}\tan^2\alpha = EhV \qquad \text{A.6}$$

$$r_2 \frac{d^2V}{ds^2} + \frac{dV}{ds}\tan\alpha - \frac{V}{r_2}\tan^2\alpha = -\frac{U}{D} \qquad \text{A.7}$$

and Eqns. 3.56 become

$$U = s\ Q_s \tan\alpha \qquad \text{A.8}$$

Substituting this into A.6 and A.7,

$$s\,\frac{d^2(sQ_s)}{ds^2} + \frac{d(sQ_s)}{ds} - \frac{(sQ_s)}{s} = EhV\cot^2\alpha \qquad \text{A.9}$$

$$s\,\frac{d^2V}{ds^2} + \frac{dV}{ds} - \frac{V}{s} = -\frac{(sQ_s)}{D} \qquad \text{A.10}$$

These equations can be solved to form a fourth order equation in terms of a single variable(7). However, an alternative approach is possible by introducing a linear differential operator as follows(1):

$$L(\;) = s\,\frac{d^2(\;)}{ds^2} + \frac{d(\;)}{ds} - \frac{(\;)}{s} \qquad \text{A.11}$$

thus, eqns. A.9 and A.10 become,

$$L(sQ_s) = EhV\cot^2\alpha \qquad \text{A.12}$$

$$L(V) = -\frac{(sQ_s)}{D} \qquad \text{A.13}$$

Operating on Eqn. A.12, and substituting back into Eqn. A.13 yields,

$$LL(sQ_s) + \lambda^4(sQ_s) = 0 \qquad \text{A.14}$$

where

$$\lambda^4 = \frac{12(1-\nu^2)}{h^2\tan^2\alpha}$$

This may be written in either of the following forms,

$$L[L(sQ_s) + i\lambda^2(sQ_s)] - i\lambda^2[L(sQ_s) + i\lambda^2(sQ_s)] = 0 \qquad \text{A.15}$$

$$L[L(sQ_s) - i\lambda^2(sQ_s)] + i\lambda^2[L(sQ_s) + i\lambda^2(sQ_s)] = 0 \qquad \text{A.16}$$

which show that the solutions of the two second-order equations are

$$L(sQ_s) \pm i\lambda^2(sQ_s) = 0 \qquad \text{A.17}$$

Expanding this equation yields,

$$s\,\frac{d^2(sQ_s)}{ds^2} + \frac{d(sQ_s)}{ds} - \frac{(sQ_s)}{s} \pm i\lambda^2(sQ_s) = 0 \qquad \text{A.18(a,b)}$$

The solution to Eqns. A.18 is complex, and it will be enough to solve one of the equations, and then use the real and imaginary parts of this solution separately as the solution of a fourth-order equation mentioned earlier. Introducing a new variable,

$$\eta = 2\lambda\sqrt{is}, \qquad \text{A.19}$$

Eqn. A.18(a) become,

$$\frac{d^2(sQ_s)}{d\eta^2} + \frac{1}{\eta}\frac{d(sQ_s)}{d\eta} + \left(1 - \frac{4}{\eta^2}\right)(sQ_s) = 0 \qquad \text{A.20}$$

The solution of this equation consists of Bessel functions of the second kind.

$$J_2(\eta) = \frac{2}{\eta} J_1(\eta) - J_0(\eta) \qquad \text{A.21(a)}$$

$$H_2^{(1)}(\eta) = \frac{2}{\eta} H_1^{(1)}(\eta) - H_0^{(1)}(\eta) \qquad \text{A.22(b)}$$

Let $\xi = 2\lambda\sqrt{s}$, then rewriting Eqns. A.21 in terms of the Kelvin functions of order zero yield

$$J_2(\eta) = \frac{2}{\xi}\text{bei}'\xi - \text{ber}\,\xi + i\left(\frac{2}{\xi}\text{ber}'\xi + \text{bei}\,\xi\right) \qquad \text{A.23(a)}$$

$$H_2^{(1)}(\eta) = \frac{2}{\eta}\,\frac{2}{\xi}\text{ker}'\xi + \text{kei}\,\xi - i\left(\frac{2}{\eta}\,\frac{2}{\xi}\text{kei}'\xi - \text{ker}\,\xi\right) \qquad \text{A.23(b)}$$

These two functions are independent solutions of Eqn. A.18, and their real and imaginary parts separately will satisfy the fourth-order equation formed by combining Eqns. A.9 and A.10. The general solution for a conical shell is

$$Q_s = \frac{1}{s}\left[A_1\left(\text{ber}\,\xi - \frac{2}{\xi}\text{bei}'\xi\right) + A_2\left(\text{bei}\,\xi + \frac{2}{\xi}\text{ber}'\xi\right)\right.$$

$$\left. + B_1\left(\text{ker}\,\xi - \frac{2}{\xi}\text{kei}'\xi\right) + B_2\left(\text{kei}\,\xi + \frac{2}{\xi}\text{ker}'\xi\right)\right] \qquad \text{A.24}$$

Using the recurrence formulas(10) for the Kelvin functions Eqn. A.24 can be rewritten as follows:

$$Q_1 = \frac{1}{s}(C_1 \mathrm{ber}_2 \xi + C_2 \mathrm{bei}_2 \xi + C_3 \mathrm{ker}_2 \xi + C_4 \mathrm{kei}_2 \xi) \qquad \text{A.25}$$

APPENDIX B-CONSTRUCTION OF THE SEGMENT FLEXIBILITY MATRIX

In general, the flexibility matrix of a shell segment is of the form

$$[F] = [TA][TT]^{-1} \qquad B.1$$

The [TA] and [TT] matrices are a function of the geometrical and material properties of the shell segment.

Based on the geometrical properties, the [TA] and [TT] matrices for the cylindrical segment, derived in a similar manner as for the spherical segment in Chapter 3, is as follows

Let

$$\beta^4 = \frac{3(1-\nu^2)}{r^2h^2}$$

$$D = \frac{Eh^3}{12(1-\nu^2)}$$

and

$$\phi_1 = e^{\beta s}\cos\beta s \qquad \theta_1 = e^{\beta s}(\cos\beta s + \sin\beta s)$$

$$\phi_2 = e^{\beta s}\sin\beta s \qquad \theta_2 = e^{\beta s}(\cos\beta s - \sin\beta s)$$

$$\phi_3 = e^{-\beta s}\cos\beta s \qquad \theta_3 = e^{-\beta s}(\cos\beta s + \sin\beta s)$$

$$\phi_4 = e^{-\beta s}\sin\beta s \qquad \theta_4 = e^{-\beta s}(\cos\beta s - \sin\beta s)$$

$$\begin{Bmatrix} H^i \\ M^i_\phi \\ H^j \\ M^j_\phi \end{Bmatrix} = \begin{bmatrix} 2D\beta^3 & 0 & 0 & 0 \\ 0 & 2D\beta^3 & 0 & 0 \\ 0 & 0 & 2D\beta^3 & 0 \\ 0 & 0 & 0 & 2D\beta^3 \end{bmatrix} \begin{bmatrix} -1 & 1 & 1 & 1 \\ 0 & -1 & 0 & 1 \\ \theta^j_1 & -\theta^j_2 & -\theta^j_4 & -\theta^j_3 \\ -\phi^j_2 & \phi^j_1 & \phi^j_4 & \phi^j_3 \end{bmatrix} \begin{Bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{Bmatrix}$$

or

$$\{V\} = [T_5][T_6]\{C\}$$

$$\{V\} = [TT]\{C\} \qquad B.2$$

and

$$\begin{Bmatrix} \Delta_H^i \\ \Delta_\phi^i \\ \Delta_H^j \\ \Delta_\phi^j \end{Bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \beta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & -1 & 1 \\ \phi_1^i & \phi_2^i & \phi_3^i & \phi_4^i \\ \theta_2^i & \theta_1^i & -\theta_3^i & \theta_4^i \end{bmatrix} \begin{Bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{Bmatrix}$$

or

$$\{\Delta\} = [T_7][T_8]\{C\}$$

$$\{\Delta\} = [TA]\{C\} \qquad B.3$$

The [TA] and [TT] matrices for the conical segment is as follows (8).

Let

$$m^4 = 12(1-\nu^2)$$

$$\lambda^4 = \frac{12(1-\nu^2)}{h^2\tan^2\alpha}$$

$$\xi = 2\lambda\sqrt{s}$$

$$(\ )' = \frac{d(\ )}{d\xi}$$

and

$$\phi_1 = ber_2\xi \qquad \theta_1 = \xi ber_2'\xi + 2\nu ber_2\xi$$

$$\phi_2 = bei_2\xi \qquad \theta_2 = \xi bei_2'\xi + 2\nu bei_2\xi$$

$$\phi_3 = ker_2\xi \qquad \theta_3 = \xi ker_2'\xi + 2\nu ker_2\xi$$

$$\phi_4 = kei_2\xi \qquad \theta_4 = \xi kei_2'\xi + 2\nu kei_2\xi$$

$$\gamma_1 = \xi ber_2'\xi - 2\nu ber_2\xi$$

$$\gamma_2 = \xi bei_2'\xi - 2\nu bei_2\xi$$

$$\gamma_3 = \xi ker_2'\xi - 2\nu ker_2\xi$$

$$\gamma_4 = \xi kei_2'\xi - 2\nu kei_2\xi$$

then

$$\begin{Bmatrix} H^i \\ M_\phi^i \\ H^j \\ M_\phi^j \end{Bmatrix} = \begin{bmatrix} \frac{-1}{s_i \sin\alpha} & 0 & 0 & 0 \\ 0 & \frac{h}{2m^2 s_i} & 0 & 0 \\ 0 & 0 & \frac{1}{s_j \sin\alpha} & 0 \\ 0 & 0 & 0 & \frac{-h}{2m^2 s_j} \end{bmatrix} \begin{bmatrix} \phi_1^i & \phi_2^i & \phi_3^i & \phi_4^i \\ \theta_2^i & -\theta_1^i & \theta_4^i & -\phi_3^i \\ \phi_1^j & \phi_2^j & \phi_3^j & \phi_4^j \\ \theta_2^j & -\theta_1^j & \theta_4^j & -\phi_3^j \end{bmatrix} \begin{Bmatrix} C_1 \\ C_2 \\ C_3 \\ C_2 \end{Bmatrix}$$

or

$$\{V\} = [T_9][T_{10}]\{C\}$$

$$\{V\} = [TT]\{C\} \qquad \text{B.4}$$

and

$$\begin{Bmatrix} \Delta_H^i \\ \Delta_\phi^i \\ \Delta_H^j \\ \Delta_\phi^j \end{Bmatrix} = \frac{1}{2Eh} \begin{bmatrix} \sin\alpha & 0 & 0 & 0 \\ 0 & -2m^2/h & 0 & 0 \\ 0 & 0 & \sin\alpha & 0 \\ 0 & 0 & 0 & -2m^2/h \end{bmatrix} \begin{bmatrix} \gamma_1^i & \gamma_2^i & \gamma_3^i & \gamma_4^i \\ \phi_2^i & -\phi_1^i & \phi_4^i & -\phi_3^i \\ \gamma_1^j & \gamma_1^j & \gamma_3^j & \gamma_4^j \\ \phi_2^j & -\phi_1^j & \phi_4^j & -\phi_3^j \end{bmatrix} \begin{Bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{Bmatrix}$$

or

$$\{\Delta\} = [T_{11}][T_{12}\{C\}$$

$$\{\Delta\} = [TA]\{C\} \qquad \text{B.5}$$

Inverted shell

An inverted cone or sphere is shown on Fig. (b) of Tables 3.5 and 3.6 respectively. Note that in the above derivations, i and j, relates to the 'top' and 'bottom' of a shell segment. So, for an inverted shell, the top becomes the bottom and vice versa. Thus, to find the flexibility matrix, it is a simple matter of interchanging rows one with three, and rows two with four, of matrices [TA] and [TT].

APPENDIX C-FLEXSHELL USER'S MANUAL

Using the flexibility method of analysis, program FLEXSHELL computes the in-plane forces, bending moments, and horizontal displacements for an axisymmetrically loaded shell structures due to various loads.

The program is capable of analyzing six types of shells of revolution of uniform thickness. These are cylinders, spheres, inverted spheres, cones, inverted cones, and base slabs on an elastic foundation. Seven axisymmetric loading cases are available. These are self weight, uniform pressure, prestressing, snow load (a uniform vertical load over a horizontal projection), hydrostatic load, uniform temperature change, and temperature gradient through the shell thickness.

The input to FLEXSHELL consists of multiple lines which may be lines in a datafile or a set of punched data cards. There are six input card types. Certain card types may be repeated as necessary.

A typical explanation of a card type consists of the card type number, a descriptive name indicating the nature of the data, the format used, and the number of cards of that type required. This is followed by the variable names, in bold type, followed by the definitions of these variable names, and the options available ,if any, for the input variables. Throughout the input all units have to be consistent. The input and output files for the Intze tank problem discussed in Chapter 5 and the program listing are given in the latter part of the Appendix.

1. **TITLE** card Format 10A8

Any identifier string up to 80 characters.

2. **CONTROL** card Format 2I4

NSEG IPRINT

NSEG = number of shell segments in the structure.

IPRINT = print control character

0 - echos input data and prints final results only;
1 - prints full output including the connectivity matrix, PSF and PBF arrays, and the element and structure flexibility matrices. (used for checking purposes only)

3. **SEGMENT DATA** card Format 5I4,2F10.4

One card per segment required. Note that the segments must be numbered sequentially in such manner that any segment always has a higher number than any of the segments which it supports.

I IT IR(1) IR(2) **NDIV EC(I,1) EC(I,2)**

I = segment number

IT = segment type

1 - cylinder
2 - sphere
3 - base on elastic foundation
4 - cone
5 - inverted sphere
6 - inverted cone

IR(1) = top connectivity flag for segment I

0 - top is not connected to another segment;
1 - top is connected to another segment.

IR(2) = bottom connectivity flag for segment I

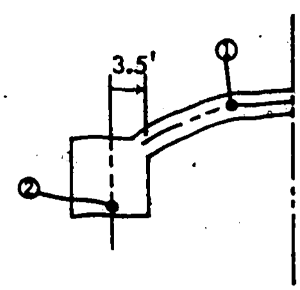
0 - bottom is not connected to another segment;
1 - bottom is connected to another segment;
-1 - bottom is connected to another segment with a pure hinge.

NDIV = number of divisions for segment I at which stress resultants are to be computed and printed. (max = 100)

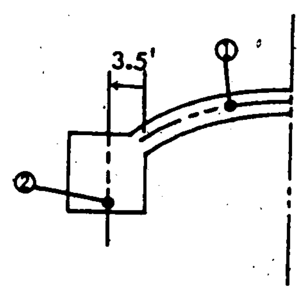
EC(I,1) = eccentricity of joint connection at the top of the segment in feet.

EC(I,2) = eccentricity of joint connection at the bottom of the segment in feet.

NOTE:
When two shell segments are connected at a given elevation but have different midsurface radii, a horizontal eccentricity equal to the differences in horizontal radii to the midsurfaces will result. This eccentricity can be applied to either shell segment and is positive when directed inwards. For the eccentricity between a spherical and cylindrical segment, EITHER of the following entries is permissible.



EC(1,2) = 0
EC(2,1) = 3.5



EC(1,2) = -3.5
EC(2,1) = 0

4. CONNECTIVITY specification card Format 2I4

Specifies the connection between segments. Requires (NSEG - 1) cards.

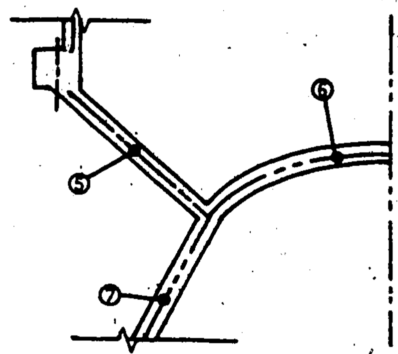
IDCO(1) IDCO(2)

IDCO(1) = number of top segment.

IDCO(2) = number of segment to which the top segment is connected.

NOTE:
Where three shell segments intersect at the same elevation, two connectivity specification cards are required. For the Intze tank shown in Fig. 5.9, the entries are 5 7, on the first card, and 6 7, on the following card. Each segment number appears precisely once in IDCO(1) and these numbers must be arranged consecutively in increasing order, starting with segment

1 and ending with segment NSEG.



5. **SEGMENT PROPERTIES** card — Format I4,F6.0,F12.0, F8.0,5F10.0

I T R H HO E PR ALPHA UW

I = segment number

T = segment thickness, feet

R = radius of the parallel circles for cylinders and spheres, feet; or,
= subgrade coefficient for base segments; or,
= semi-vertex angle of cone in degrees.

H = length in feet for cylinder; or,
= total angle in degrees from the axis of revolution to the outer edge for a sphere; or,
= outer radius of a circular base ring slab in feet; or,
= distance from the apex of cone to the large end in feet.

HO = 0.0 or blank for cylinder; or,
= angle in degrees from the axis of revolution to the inner edge for a sphere; or,
= inner radius of a circular base ring slab; or,
= distance from the apex of cone to inner edge in feet; or,

E = Young's modulus for segment, psf

PR = Poisson's ratio for segment

ALPHA = coefficient of thermal expansion

UW = unit weight of material for segment, pcf

6. **LOAD TYPE** card — Format 2I4,7F10.0

One card per segment is required.

I IP PV WHT PSF(1) PSF(2) PSF(3) PSF(4) PSF(5) PSF(6)

I = segment number

IP = load type parameter

1 - uniform pressure
2 - self weight
3 - prestress loading
4 - uniform temperature change across section
5 - temperature gradient across section
6 - uniformly distributed load over a horizontal projection, or snow load
7 - liquid pressure

NOTE:
A hydrostatic load applied to the base segment is simulated by using a uniform pressure equal to the product of the liquid weight density and the height of the water above the base.

PV = value of the applied load, depending on the type of load.

If IP=1, PV is the magnitude of uniform pressure in psf.
Positive for internally directed pressure and negative for externally directed pressure. For a base segment, this value is positive when pressure is directed downward and negative when directed upward.

If IP=2, value of PV is disregarded and a dead load analysis is carried out for the unit weights specified on the SEGMENT PROPERTIES cards.

If IP=3, PV is the magnitude of the uniformly distributed prestress pressure on the midsurface. Same sign convention as IP=1.

If IP=4, PV is the uniform temperature change in degree Celsius or Fahrenheit, depending on the units of ALPHA. (positive if the temperature rises above the reference temperature)

If IP=5, PV is the gradient of temperature across section in degrees per unit of thickness. (positive if the temperature rises above the reference temperature)

If IP=6, PV is the magnitude of uniform pressure distributed over a horizontal projection, or snow load, in psf

If IP=7, PV is the magnitude of the liquid weight density in pcf.

WHT = height of liquid above the vertex of a cone or height of liquid above the inner edge of a sphere in feet. Value is ignored for load types other than liquid pressure.

PSF(1) = magnitude of externally applied horizontal force at the top of the segment, lbs/ft.

PSF(2) = magnitude of externally applied moment at the top of the segment, ft-lbs/ft.

PSF(3) = magnitude of externally applied horizontal force at the bottom of the segment, lbs/ft.

PSF(4) = magnitude of externally applied moment at the bottom of the segment, ft-lbs/ft.

PSF(5) = magnitude of externally applied vertical force at the top of the segment, lbs/ft.

PSF(6) = magnitude of externally applied vertical force at the bottom of the segment, lbs/ft.

NOTE:

The PSF forces are forces and moments which, if necessary, are to be applied IN ADDITION TO the distributed loading effects identified by the PV values. Prestressing effects are generally simulated as distributed loads but cable anchorages give rise to concentrated loads which are treated as PSF forces.

FLEXSHELL Input and Output Files

for the Intze Tank Problem

```
INTZE TANK MODEL (DEADLOAD)
8,
1,2,0,1,5,
2,1,1,1,5,.216,.125,
3,1,1,1,5,
4,1,1,1,5,.125,.125,
5,6,1,1,5,
6,2,0,1,5,
7,4,1,1,40,
8,3,1,0,1,
1,2,
2,3,
3,4,
4,5,
5,7,
6,7,
7,8,
1,.333,94.5,28.,0.,.5804E09,.167,.6E-5,150.,
2,.667,44.581,1.45,0.,.5804E9,.167,.6E-5,150.,
3,.417,44.456,10.,0.,.5804E9,.167,.6E-5,150.,
4,.667,44.581,1.45,0.,.5804E9,.167,.6E-5,150.,
5,.3,62.75,50.,25.,.5804E9,.167,.6E-5,150.,
6,.3,82.5,15.5,0.,.5804E9,.167,.6E-5,150.,
7,.5,19.146,152.447,67.763,.5804E9,.167,.6E-5,150.,
8,2.,450000.,50.,0.,1.E20,.167,.6E-5,150.,
1,2,
2,2,
3,2,
4,2,
5,2,
6,2,
7,2,
8,2,
```

1INTZE TANK MODEL (DEADLOAD)

1 **** OUTPUT FOR FLEXIBILITY ANALYSIS OF SEGMENTED SHELL ****

NUMBER OF SEGMENTS = 8

IPRINT = 0

SEG	TYPE	IR	JR	NDIV	EC1	EC2
1	2	0	1	5	0.0	0.0
2	1	1	1	5	0.2160	0.1250
3	1	1	1	5	0.0	0.0
4	1	1	1	5	0.1250	0.1250
5	6	1	1	5	0.0	0.0
6	2	0	1	5	0.0	0.0
7	4	1	1	40	0.0	0.0
8	3	1	0	1	0.0	0.0

CONNECTIVITY MATRIX

1	2
2	3
3	4
4	5
5	7
6	7
7	8

GEOMETRIC PARAMETERS

SEG	THICK	RADIUS	L OR ANG	ANGO	MODULUS	P RATIO	THERMCOEF	WEIGHT
1	0.333	94.500	28.000	0.0	0.5804E+09	0.167	0.6000E-05	150.000
2	0.667	44.581	1.450	0.0	0.5804E+09	0.167	0.6000E-05	150.000
3	0.417	44.456	10.000	0.0	0.5804E+09	0.167	0.6000E-05	150.000
4	0.667	44.581	1.450	0.0	0.5804E+09	0.167	0.6000E-05	150.000
5	0.300	62.750	50.000	25.000	0.5804E+09	0.167	0.6000E-05	150.000
6	0.300	82.500	15.500	0.0	0.5804E+09	0.167	0.6000E-05	150.000
7	0.500	19.146	152.447	67.763	0.5804E+09	0.167	0.6000E-05	150.000
8	2.000	450000.000	50.000	0.0	0.1000E+21	0.167	0.6000E-05	150.000

PARTICULAR SOLUTION INPUT INFORMATION

SEG	TYPE	VALUE	LIQ HT	TOP SHEAR	TOP MOM	BOT SHEAR	BOT MOM	TOP FORCE	VERT FORCE

1	2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

FORCES ON ENDS OF SEGMENTS

SEG	J	IF	FORCE
1	3	1	-0.11739E+04
1	4	2	0.13300E+04
2	1	3	0.11739E+04
2	2	4	-0.13300E+04
2	3	5	0.50108E+03
2	4	6	0.32811E+03
3	1	7	-0.50108E+03
3	2	8	-0.32811E+03
3	3	9	-0.83740E+03
3	4	10	0.13911E+03
4	1	11	0.83740E+03
4	2	12	-0.13911E+03
4	3	13	-0.21104E+04
4	4	14	0.22062E+04
5	1	15	0.21104E+04
5	2	16	-0.22062E+04
5	3	17	0.35246E+04
5	4	18	-0.25061E+04
6	3	19	0.40603E+04
6	4	20	-0.21434E+04
7	1	21	-0.75849E+04
7	2	22	0.46496E+04
7	3	23	-0.34904E+02
7	4	24	0.57239E+02
8	1	25	0.34904E+02
8	2	26	-0.57239E+02
8	5	27	0.0

INDIVIDUAL END FORCES FOR SEGMENT 1

0.0	-0.0	-0.11739E+04	0.13300E+04
0.0	0.11769E+04		

*** OUTPUT FOR DOME SEGMENT 1 ***

POINT	ANGLE	N1	N2	M1	M2
1	0.0	-0.23601E+04	-0.23605E+04	0.14759E-01	0.24648E-02
2	5.6000	-0.23664E+04	-0.23315E+04	-0.30456E+00	-0.50862E-01
3	11.2000	-0.23784E+04	-0.22274E+04	0.17828E+01	0.29772E+00
4	16.8000	-0.24242E+04	-0.23336E+04	0.56958E+01	0.95120E+00
5	22.4000	-0.25218E+04	-0.12631E+04	-0.18557E+03	-0.30990E+02
6	28.0000	-0.14704E+04	0.89622E+04	0.13300E+04	0.22212E+03

*** HORIZONTAL DISPLACEMENT ***

POINT	COORD	W
1	0.0	0.0
2	5.6000	0.92390E-04
3	11.2000	0.17375E-03
4	16.8000	0.27288E-03
5	22.4000	0.15902E-03
6	28.0000	-0.21533E-02

INDIVIDUAL END FORCES FOR SEGMENT 2

-0.10288E+04 -0.15830E+04 0.50108E+03 0.49264E+03
0.11712E+04 0.14507E+03

*** OUTPUT FOR CYLINDRICAL SEGMENT 2 ***

POINT	COORD	N1	N2	M1	M2
1	0.0	-0.11712E+04	0.18503E+05	0.15830E+04	0.26436E+03
2	0.2900	-0.12002E+04	0.17658E+05	0.13019E+04	0.21741E+03
3	0.5800	-0.12292E+04	0.16749E+05	0.10540E+04	0.17602E+03
4	0.8700	-0.12582E+04	0.15787E+05	0.83775E+03	0.13990E+03
5	1.1600	-0.12873E+04	0.14784E+05	0.65125E+03	0.10876E+03
6	1.4500	-0.13163E+04	0.13748E+05	0.49264E+03	0.82272E+02

*** HORIZONTAL DISPLACEMENT ***

POINT	COORD	W
1	0.0	-0.21533E-02
2	0.2900	-0.20566E-02

3	0.5800	-0.19524E-02
4	0.8700	-0.18422E-02
5	1.1600	-0.17272E-02
6	1.4500	-0.16085E-02

INDIVIDUAL END FORCES FOR SEGMENT 3
-0.50108E+03 -0.32811E+03 -0.83740E+03 0.13911E+03
0.13200E+04 0.62550E+03

*** OUTPUT FOR CYLINDRICAL SEGMENT 3 ***

POINT	COORD	N1	N2	M1	M2
1	0.0	-0.13200E+04	0.85368E+04	0.32811E+03	0.54795E+02
2	2.0000	-0.14451E+04	0.38472E+04	-0.36143E+03	-0.60359E+02
3	4.0000	-0.15702E+04	0.11619E+04	-0.68899E+03	-0.11506E+03
4	6.0000	-0.16953E+04	0.25766E+04	-0.88098E+03	-0.14712E+03
5	8.0000	-0.18204E+04	0.91843E+04	-0.80163E+03	-0.13387E+03
6	10.0000	-0.19455E+04	0.20208E+05	0.13911E+03	0.23232E+02

*** HORIZONTAL DISPLACEMENT ***

POINT	COORD	W
1	0.0	-0.16085E-02
2	2.0000	-0.75099E-03
3	4.0000	-0.26159E-03
4	6.0000	-0.52527E-03
5	8.0000	-0.17428E-02
6	10.0000	-0.37715E-02

INDIVIDUAL END FORCES FOR SEGMENT 4
0.83740E+03 -0.38162E+03 -0.21104E+04 0.24669E+04
0.19400E+04 0.14507E+03

*** OUTPUT FOR CYLINDRICAL SEGMENT 4 ***

POINT	COORD	N1	N2	M1	M2
1	0.0	-0.19400E+04	0.32426E+05	0.38162E+03	0.63730E+02
2	0.2900	-0.19690E+04	0.35164E+05	0.65591E+03	0.10954E+03
3	0.5800	-0.19980E+04	0.37868E+05	0.99654E+03	0.16642E+03

4	0.8700	-0.20271E+04	0.40523E+05	0.14086E+04	0.23523E+03
5	1.1600	-0.20561E+04	0.43108E+05	0.18971E+04	0.31681E+03
6	1.4500	-0.20851E+04	0.45598E+05	0.24669E+04	0.41197E+03

*** HORIZONTAL DISPLACEMENT ***

POINT	COORD	W
1	0.0	-0.37715E-02
2	0.2900	-0.40873E-02
3	0.5800	-0.43992E-02
4	0.8700	-0.47055E-02
5	1.1600	-0.50038E-02
6	1.4500	-0.52911E-02

INDIVIDUAL END FORCES FOR SEGMENT 5

-0.19499E+04 -0.22062E+04 0.35246E+04 -0.25061E+04
0.20912E+04 0.16875E+04

*** OUTPUT FOR CONE SEGMENT 5 ***

POINT	COORD	N1	N2	M1	M2
1	0.0	-0.28337E+04	0.20253E+05	0.22062E+04	0.39798E+03
2	5.0000	-0.52972E+04	0.12518E+05	-0.21467E+03	-0.78497E+02
3	10.0000	-0.69466E+04	0.33836E+04	-0.25444E+03	-0.51877E+02
4	15.0000	-0.84461E+04	0.28379E+04	0.73113E+02	0.14169E+02
5	20.0000	-0.10222E+05	-0.22919E+04	0.41684E+03	0.29230E+02
6	25.0000	-0.96864E+04	-0.23897E+05	-0.25061E+04	-0.40078E+03

*** HORIZONTAL DISPLACEMENT ***

POINT	COORD	W
1	0.0	-0.41049E-02
2	5.0000	-0.20616E-02
3	10.0000	-0.60989E-04
4	15.0000	-0.25228E-04
5	20.0000	0.70828E-03
6	25.0000	0.33650E-02

INDIVIDUAL END FORCES FOR SEGMENT 6

```
0.0          -0.0              0.40603E+04 -0.21434E+04
0.0           0.50525E+03
```

*** OUTPUT FOR DOME SEGMENT 6 ***

POINT	ANGLE	N1	N2	M1	M2
1	0.0	-0.18563E+04	-0.18866E+04	-0.78728E+01	-0.13148E+01
2	3.1000	-0.18465E+04	-0.16207E+04	-0.17812E+02	-0.29747E+01
3	6.2000	-0.15819E+04	-0.93874E+03	0.37511E+02	0.62643E+01
4	9.3000	-0.14417E+04	-0.19512E+04	0.27983E+03	0.46731E+02
5	12.4000	-0.25239E+04	-0.11428E+05	0.30975E+03	0.51729E+02
6	15.5000	-0.58033E+04	-0.24313E+05	-0.21434E+04	-0.35796E+03

*** HORIZONTAL DISPLACEMENT ***

POINT	COORD	W
1	0.0	0.0
2	3.1000	0.33578E-04
3	6.2000	0.32127E-04
4	9.3000	0.12551E-03
5	12.4000	0.11308E-02
6	15.5000	0.30386E-02

INDIVIDUAL END FORCES FOR SEGMENT 7

```
0.42172E+04  0.46496E+04 -0.34904E+02  0.57239E+02
0.63713E+04  0.45872E+04
```

*** OUTPUT FOR CONE SEGMENT 7 ***

POINT	COORD	N1	N2	M1	M2
1	0.0	-0.53612E+04	-0.40572E+05	-0.46496E+04	-0.79811E+03
2	2.1171	-0.63364E+04	-0.22809E+05	0.41127E+03	0.11532E+01
3	4.2342	-0.67073E+04	-0.66651E+04	0.11506E+04	0.15499E+03
4	6.3513	-0.67407E+04	-0.82821E+02	0.64864E+03	0.98048E+02
5	8.4684	-0.66822E+04	0.93047E+03	0.18812E+03	0.32149E+02
6	10.5855	-0.66824E+04	0.26586E+03	-0.95729E+01	0.95069E+00
7	12.7026	-0.66082E+04	-0.38905E+03	-0.46312E+02	-0.61822E+01
8	14.8197	-0.66016E+04	-0.70163E+03	-0.29901E+02	-0.44729E+01
9	16.9368	-0.66036E+04	-0.78658E+03	-0.10808E+02	-0.17773E+01
10	19.0539	-0.66100E+04	-0.78751E+03	-0.10076E+01	-0.26302E+00
11	21.1710	-0.66195E+04	-0.77856E+03	0.17451E+01	0.21710E+00

12	23.2881	-0.66319E+04	-0.78092E+03	0.15268E+01	0.22221E+00
13	25.4052	-0.66474E+04	-0.79364E+03	0.73281E+00	0.11530E+00
14	27.5223	-0.66658E+04	-0.81139E+03	0.18998E+00	0.34128E-01
15	29.6394	-0.66872E+04	-0.83050E+03	-0.33045E-01	-0.19797E-02
16	31.7565	-0.67112E+04	-0.84945E+03	-0.73377E-01	-0.10094E-01
17	33.8736	-0.67377E+04	-0.86796E+03	-0.49997E-01	-0.75298E-02
18	35.9907	-0.67666E+04	-0.88616E+03	-0.21385E-01	-0.34640E-02
19	38.1078	-0.67977E+04	-0.90422E+03	-0.42646E-02	-0.83598E-03
20	40.2249	-0.68309E+04	-0.92226E+03	0.25118E-02	0.29426E-03
21	42.3420	-0.68660E+04	-0.94031E+03	0.39232E-02	0.59777E-03
22	44.4591	-0.69029E+04	-0.95839E+03	0.33923E-02	0.57897E-03
23	46.5762	-0.69416E+04	-0.97648E+03	0.17878E-02	0.35932E-03
24	48.6933	-0.69820E+04	-0.99458E+03	-0.19939E-02	-0.27078E-03
25	50.8104	-0.70239E+04	-0.10127E+04	-0.97452E-02	-0.16616E-02
26	52.9275	-0.70673E+04	-0.10307E+04	-0.21633E-01	-0.39092E-02
27	55.0446	-0.71121E+04	-0.10486E+04	-0.32338E-01	-0.61519E-02
28	57.1617	-0.71582E+04	-0.10664E+04	-0.26213E-01	-0.56353E-02
29	59.2788	-0.72056E+04	-0.10841E+04	0.25310E-01	0.29012E-02
30	61.3959	-0.72542E+04	-0.11020E+04	0.15686E+00	0.26221E-01
31	63.5130	-0.73039E+04	-0.11206E+04	0.38208E+00	0.67942E-01
32	65.6301	-0.73547E+04	-0.11409E+04	0.63675E+00	0.11830E+00
33	67.7472	-0.74067E+04	-0.11639E+04	0.69623E+00	0.13861E+00
34	69.8643	-0.74597E+04	-0.11896E+04	0.10455E+00	0.46561E-01
35	71.9814	-0.75137E+04	-0.12150E+04	-0.17909E+01	-0.28089E+00
36	74.0985	-0.75688E+04	-0.12319E+04	-0.55410E+01	-0.95822E+00
37	76.2156	-0.76245E+04	-0.12251E+04	-0.10849E+02	-0.19601E+01
38	78.3327	-0.76804E+04	-0.11747E+04	-0.15302E+02	-0.28888E+01
39	80.4498	-0.77357E+04	-0.10664E+04	-0.12846E+02	-0.26814E+01
40	82.5669	-0.77896E+04	-0.91636E+03	0.70031E+01	0.57510E+00
41	84.6840	-0.78423E+04	-0.80901E+03	0.57239E+02	0.93665E+01

*** HORIZONTAL DISPLACEMENT ***

POINT	COORD	W
1	0.0	0.30805E-02
2	2.1171	0.17591E-02
3	4.2342	0.49185E-03
4	6.3513	-0.47361E-04
5	8.4684	-0.13700E-03
6	10.5855	-0.83019E-04
7	12.7026	-0.27107E-04
8	14.8197	-0.28402E-06
9	16.9368	0.60918E-05
10	19.0539	0.45373E-05
11	21.1710	0.19175E-05

12	23.2881	0.34160E-06
13	25.4052	-0.21467E-06
14	27.5223	-0.25388E-06
15	29.6394	-0.14569E-06
16	31.7565	-0.50613E-07
17	33.8736	-0.26017E-08
18	35.9907	0.11343E-07
19	38.1078	0.10000E-07
20	40.2249	0.51889E-08
21	42.3420	0.18195E-08
22	44.4591	0.10329E-08
23	46.5762	0.24987E-08
24	48.6933	0.50482E-08
25	50.8104	0.60002E-08
26	52.9275	0.80871E-10
27	55.0446	-0.20457E-07
28	57.1617	-0.61925E-07
29	59.2788	-0.11842E-06
30	61.3959	-0.15286E-06
31	63.5130	-0.75072E-07
32	65.6301	0.26391E-06
33	67.7472	0.10232E-05
34	69.8643	0.22154E-05
35	71.9814	0.33988E-05
36	74.0985	0.32529E-05
37	76.2156	-0.75002E-06
38	78.3327	-0.12066E-04
39	80.4498	-0.33345E-04
40	82.5669	-0.62278E-04
41	84.6840	-0.84591E-04

INDIVIDUAL END FORCES FOR SEGMENT 8

-0.25408E+04 -0.57239E+02 0.0 0.0
0.74191E+04 0.0

*** OUTPUT FOR BASE ELEMENT 8 ***

POINT	COORD	M1	M2	V
1	50.0000	-0.57239E+02	0.26938E+06	0.74191E+04
2	7027.0831	-0.87313E+02	0.13066E+04	0.90469E+00

```
*** VERTICAL DISPLACEMENT ***
POINT   COORD       W
    1   50.0000   0.66667E-03
    2 7027.0831   0.66667E-03
```

FLEXSHELL Program Listing

```
C           ***   FLEXSHELL   ***
C
C           THIS PROGRAM WAS ORIGINALLY DEVELOPED BY D.W. MURRAY
C     AND A.M. ROHARDT IN DECEMBER, 1976 (REVISED IN MAY, 1977)
C     FOR THE FLEXIBILITY ANALYSIS OF SEGMENTED AXISYMMETRIC
C     SHELLS SUCH AS THE 'SHORT' CYLINDER, 'LONG' SPHERE, AND
C     BASE SEGMENTS, SUBJECTED TO DEAD LOAD, UNIFORM PRESSURE
C     OR PRESTRESSING, AND THERMAL STRESSES.  REDUNDANT FORCES
C     ARE APPLIED TO THE INDIVIDUAL SEGMENTS TO ESTABLISH THE
C     REQUIRED GEOMETRIC COMPATIBILITY.
C
C           THE CAPABILITY OF THE PROGRAM TO  HANDLE A WIDER
C     VARIETY OF PROBLEMS WAS INCREASED IN SPRING, 1983 BY
C     N. HERNANDEZ.  IN ADDITION TO THE EXISTING SEGMENTS, THE
C     'SHORT' SPHERE, CONE, INVERTED SPHERE, AND INVERTED CONE
C     WERE ADDED.  THE 'LONG' SPHERE WAS REPLACED.  TWO LOAD CASES
C     WERE ALSO INCLUDED:  THE SNOW LOAD WHICH IS A UNIFORM
C     PRESSURE OVER THE HORIZONTAL PROJECTION OF THE SEGMENT;
C     AND THE LIQUID PRESSURE LOADING.  CONSEQUENTLY, THE
C     FOLLOWING OPERATIONS WERE COMPLETELY RECODED:
C
C           1.  CALCULATION OF THE MEMBRANE STRESSES.
C           2.  CALCULATION OF THE EFFECTS OF A VERTICAL EDGE LOAD.
C           3.  CONSTRUCTION OF THE FLEXIBILITY MATRIX.
C           4.  CALCULATION OF THE MEMBRANE DISPLACEMENTS.
C           5.  CALCULATION OF THE FINAL STRESS RESULTANTS AND
C               DISPLACEMENTS.
C
C           *** NOTATION ***
C
C      IT           SEGMENT TYPE
C       1        CYLINDER
C       2        SPHERICAL DOME
C       3        CONICAL DOME
C       4        BASE ON ELASTIC FOUNDATION
C       5        INVERTED SPHERE
C       6        INVERTED CONE
C
C      IP          TYPE OF LOADING
C       1        INTERNAL PRESSURE
C       2        DEAD LOAD
C       3        PRESTRESS
C       4        UNIFORM THERMAL STRAIN
C       5        GRADIENT THERMAL STRAIN
C       6        UNIFORM LOAD OVER A HORIZONTAL PROJECTION
C       7        LIQUID PRESSURE
C
C   GEOMETRIC VARIABLES
C   SEG. TYPE  CYLINDER    SPHERE        CONE                BASE
C
C       T      THICKNESS   THICKNESS     THICKNESS           THICKNESS
C       R      RADIUS      RADIUS        SEMI-VERTEX ANGLE   BASE STIFFN
C       H      LENGTH      OUTER ANGLE   DIST. FR. VERTEX    OUTER RADIUS
C                                        TO LARGE END
C      HO        --        INNER ANGLE   DIST. FR. VERTEX    INNER RADIUS
C                                        TO SMALL END
C
C   INDECES
C       NF=NUMBER OF SEGMENT EDGE FORCES            NR=NUMBER OF REDUNDANTS
C       NSEG=NUMBER OF SEGMENTS
C
C   MAIN ARRAYS
C        IR(*,1)=REDUNDANT FLAG TOP OF ELEMENT
C        IR(*,2)=REDUNDANT FLAG BOTTOM OF ELEMENT
C        IDF=IDENTITY OF UNKNOWN FORCES AT TOP AND BOTTOM OF ELEMENTS
C        PBF=PARTICULAR SOLUTION BASE FORCES
C        PSD=PARTICULAR SOLUTION EDGE DISPLACEMENTS
C        PARD=PARTICULAR SOLUTION INCOMPATIBLE DISPLACEMENTS
C        PSF=PARTICULAR SOLUTION FORCES WHICH PRODUCE ADDITIONAL INCOMPAT
C                                       -IBLE DISPLACEMENTS
C        A=MATRIX ESTABLISHING GEOMETRIC COMPATIBILITY BETWEEN DEGREES OF
C                                      FREEDOM
C   EXTERNAL FUNCTIONS AND SUBROUTINES FOLLOW THE MAIN PROGRAM
C   IN THE FOLLOWING ORDER:
C       FUNCTIONS:      SUBROUTINES:
C       1. FN1          4. PCYLIN       9. BASE        14. PCONE
C       2. FN2          5. CYLIN       10. BSHAPE      15. CONE
C       3. FN3          6. PDOME       11. JINVER      16. MMKEL2
C       4. FN4          7. DOME        12. SOL         17. TTINV
C                       8. PBASE       13. PFOR        18. ROWEX
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION T(20),R(20),H(20),HO(20),E(20),PR(20),ALPHA(20),PV(20),
     * S(8,8),PSD(8),F(80,80),TT(80,80),PARD(80),PART(80), PBF(8,20),
     * FO(80),SF(8),CVEC(4),XH(101),A(80,80),TS(4,4),PSF(20,8),BB(4,4),
     * RM1(101),RM2(101),RN1(101),RN2(101),EC(20,2),UW(20),TITLE(10)
      DIMENSION IT(20),IR(20,2),IP(20),IDCO(19,2),IDF(19,6),NDIV(20)
     * ,IBASE(6),IVECT(4),SR(10),M1(100),M2(100),V(100),HR(101),W(100)
     *,WHT(20)
      DATA PI/3.1415926536/ ,RAD/57.295779513/ ,IBASE/1,2,5,3,4,6/
C-----------------------------------------------------------------------
C        READ AND ECHOCHECK DATA
C-----------------------------------------------------------------------
      READ(5,1001) TITLE
      WRITE(6,2001) TITLE
      READ(5,1000)  NSEG,IPRINT
      WRITE(6,2000)  NSEG,IPRINT
      IF(NSEG.GT.20)  GO TO 999
      READ(5,1000) (I,IT(I),IR(I,1),IR(I,2),NDIV(I),EC(I,1),EC(I,2),
     *   I=1,NSEG)
      WRITE(6,2100) (I,IT(I),IR(I,1),IR(I,2),NDIV(I),EC(I,1),EC(I,2),
     *   I=1,NSEG)
      NSEG1=NSEG-1
      READ(5,1200) ((IDCO(I,J),J=1,2),I=1,NSEG1)
      WRITE(6,2200) ((IDCO(I,J),J=1,2),I=1,NSEG1)
      READ(5,1300) (I,T(I),R(I),H(I),HO(I),E(I),PR(I),ALPHA(I),UW(I),
     *   I=1,NSEG)
      WRITE(6,2300) (I,T(I),R(I),H(I),HO(I),E(I),PR(I),ALPHA(I),UW(I),
     *   I=1,NSEG)
      READ(5,1400) (I,IP(I),PV(I),WHT(I),(PSF(I,J),J=1,8),I=1,NSEG)
      WRITE(6,2400) (I,IP(I),PV(I),WHT(I),(PSF(I,J),J=1,8),I=1,NSEG)
      P4=PI/4
      R2=2.0
      R2=DSQRT(R2)
```

```
      IFLAG=0
C-------------------------------------------------------------------------
C IDENTIFY DEGREES OF FREEDOM AND FORM A-MATRIX
C-------------------------------------------------------------------------
C     FORM IDF ARRAY
      DO 30 J=1,6
      DO 30 I=1,NSEG
   30 IDF(I,J)=0
C
      KV=0
      KOUNT=0
      NRH=0
      DO 50 I=1,NSEG
      DO 50 J=1,2
      J2=2*J-1
      IF(IR(I,J).EQ.0) GO TO 50
      IDF(I,J2)=KOUNT+1
      IF(IR(I,J).GT.0) GOTO 40
      KOUNT=KOUNT+1
      NRH=-1
      GOTO 41
   40 IDF(I,J2+1)=KOUNT+2
      KOUNT=KOUNT+2
   41 IF(IT(I).NE.3)  GO TO 50
      IF(J.EQ.1)  J2=5
      IF(J.EQ.2)  J2=6
      KOUNT=KOUNT+1
      IDF(I,J2)=KOUNT
      KV=KV+1
50     CONTINUE
      NF=KOUNT
      NR=2*NSEG1+KV/2+NRH
      IF(NF.GT.80) GO TO 999
C
      IF(IPRINT.EQ.0) GOTO 45
      WRITE(6,2435)
      DO 35 I=1,NSEG
      WRITE(6,2440) (IDF(I,J), J=1,6)
   35 CONTINUE
 2440 FORMAT(20I5)
 2435 FORMAT(///'  THE IDF MATRIX IS'/)
   45 CONTINUE
C
C     FORM A MATRIX
      IO=1
      DO 150 I=1,NR
      DO 150 J=1,NF
  150 A(I,J)=0.0
      DO 300 I=1,NSEG1
      K=IDCO(I,1)
      L=IDCO(I,2)
  250 J1=IDF(K,3)
      J2=IDF(L,1)
      A(IO,J1)=1.
      A(IO,J2)=-1.
      IF(IDF(K,4).EQ.0.OR.IDF(L,2).EQ.0) GOTO 280
      A(IO+1,J1+1)=1.
      A(IO+1,J2+1)=-1.
C
  280 IF(IT(L).EQ.3) A(IO,J2+1)=+EC(L,1)
      IF(IT(K).EQ.3.AND.IT(L).EQ.3) A(IO,J2+1)=-EC(K,2)
      IO=IO+2
      IF(IDF(K,4).EQ.0.OR.IDF(L,2).EQ.0) IO=IO-1
      IF(IT(1).NE.3.OR.IT(L).NE.3) GOTO 300
      J1=IDF(K,6)
      J2=IDF(L,5)
      A(IO,J1)=1
      A(IO,J2)=-1
      IO=IO+1
  300 CONTINUE
C
      IF(IPRINT.EQ.0) GO TO 351
      WRITE(6,2450)
      DO 350 I=1,NR
      WRITE(6,2500) (A(I,J),J=1,NF)
350   CONTINUE
2450  FORMAT(///'  THE A CONNECTIVITY MATRIX IS')
2500  FORMAT(20F4.1)
C-------------------------------------------------------------------------
C  CONSTRUCT BASIC FORCES FOR PARTICULAR SOLUTIONS (PBF AND PSF ARRAYS)
C-------------------------------------------------------------------------
351   DO 355 N=1,NSEG
      DO 355 J=1,6
355   PBF(J,N)=0.0
      DO 356 I=1,NSEG
356   PBF(5,I)=PSF(1,5)
C
      DO 395 N=1,NSEG
C
      IDV = 0
      CALL PFOR(IT(N),T(N),R(N),H(N),HO(N),WHT(N),E(N),PR(N),UW(N)
     *  ,ALPHA(N),IP(N),PV(N),PBF(5,N),PSF,IDV,N)
C
      ITN = IT(N)
      GOTO (360,361,362,363,364,365),ITN
C
C     CYLINDER
C
  360 RN = R(N)
      RO = 1.
      GO TO 370
C
C     SPHERE
C
  361 RN = R(N)*DSIN(H(N)/RAD)
      RO = DSIN(HO(N)/RAD)/DSIN(H(N)/RAD)
      Z1 = 0.
      IF(HO(N).NE.0.)  Z1 = DCOS(HO(N)/RAD)/DSIN(HO(N)/RAD)
      Z2 = DCOS(H(N)/RAD)/DSIN(H(N)/RAD)
      GO TO 363
C
C     BASE ON ELASTIC FOUNDATION
C
```

```
      362  IF(HO(N).LT.1.0E-02) GO TO 361
           PSF(N,4)=PSF(N,4)+PBF(4,N)
      361  PSF(N,2)=PSF(N,2)+PBF(2,N)
           PSF(N,5)=PBF(5,N)
           GO TO 395
     C
     C     CONE
     C
      363  RN = H(N)*DSIN(R(N)/RAD)
           RO = HO(N)/H(N)
           Z1 = DTAN(R(N)/RAD)
           Z2 = Z1
           GO TO 369
     C
     C     INVERTED SPHERE
     C
      364  RN = R(N)*DSIN(HO(N)/RAD)
           Z1 = -DCOS(H(N)/RAD)/DSIN(H(N)/RAD)
           RO = 0.
           Z2 = 0.
           IF (HO(N).EQ.0.)  GO TO 369
           RO = DSIN(H(N)/RAD)/DSIN(HO(N)/RAD)
           Z2 = -DCOS(HO(N)/RAD)/DSIN(HO(N)/RAD)
           GO TO 369
     C
     C.    INVERTED CONE
     C
      365  RN = HO(N)*DSIN(R(N)/RAD)
           Z1 = -DTAN(R(N)/RAD)
           Z2 = Z1
           RO = 0.
           IF (HO(N).EQ.0.)  GO TO 369
           RO = H(N)/HO(N)
      369  PSF(N,1)=PSF(N,1)+PBF(5,N)*Z1
      370  PSF(N,2)=-PBF(5,N)*EC(N,1)+PBF(2,N)+PSF(N,2)
           PSF(N,4)=PSF(N,4)+(PBF(5,N)*RO+PBF(6,N))*EC(N,2)+PBF(4,N)
     C
           DO 385  I=1,NSEC1
           IF(IDCO(I,1).NE.N) GO TO 385
           L=IDCO(I,2)
           RL = R(L)
           IF(IT(L).EQ.2)  RL = RL*DSIN(HO(L)/RAD)
           IF(IT(L).EQ.3)  RL = H(L)
           IF(IT(L).EQ.4)  RL = HO(L)*DSIN(RL/RAD)
           IF(IT(L).EQ.5)  RL = RL*DSIN(H(L)/RAD)
           IF(IT(L).EQ.6)  RL = H(L)*DSIN(RL/RAD)
           IF(RL.LT.1.0D-06) GO TO 959
           IF(ITN.NE.1) PSF(L,1)=PSF(L,1)+(PBF(3,N)-PBF(5,N)*RO*Z2)*RN/RL
           PSF(6,L)=PSF(6,L)+(PBF(5,N)*RO+PBF(6,N))*RN/RL
           IF(IT(L).EQ.3) PSF(L,2)=PSF(L,2)-PBF(3,N)*EC(L,1)
      385  CONTINUE
      395  CONTINUE
           IDV = 1
           IF(IPRINT.EQ.0)GOTO 398
           WRITE(6,2550)
           DO 396 I=1,NSEG
      396  WRITE(6,2551) (PBF(J,I),J=1,6)
           WRITE(6,2552)
           DO 397 I=1,NSEG
      397  WRITE(6,2553) (PSF(I,J),J=1,6)
     2550  FORMAT(///,'  ***  PBF  ***  '/)
     2551  FORMAT(6E13.4)
     2552  FORMAT(///,'  ***  PSF  ***  '/)
     2553  FORMAT(6E13.4)
     C-----------------------------------------------------------------
     C  CONSTRUCT AND ASSEMBLE ELEMENT FLEXIBILITY MATRICES
     C  AND INITIAL DISPLACEMENT VECTOR
     C-----------------------------------------------------------------
      398  DO 400  I=1,NF
           PARD(I)=0.0
           DO 400 J=1,NF
      400  F(I,J)=0.0
     C
           DO 500 N=1,NSEG
           ITN = IT(N)
           IFLAG=0
           GOTO (401,405,410,470,405,470),ITN
      401  CALL CYLIN(T(N),R(N),H(N),HO(N),E(N),PR(N),UW(N),S,TS,D,BETA,
          *    IFLAG)
           PST=PBF(5,N)
           CALL PCYLIN(T(N),R(N),H(N),HO(N),E(N),PR(N),UW(N),ALPHA(N),S,PSD,
          * IP(N),PV(N),N,PSF,PST)
     C
           IF(IPRINT.EQ.0) GO TO 420
           WRITE(6,2600) N,((S(I,J),J=1,4),I=1,4)
     2600  FORMAT(///'  FLEXIBILITY MATRIX FOR CYLINDRICAL SEGMENT',I4/
          *  (4E16.5))
     C
           GOTO 420
     C
     C SPHERICAL SEGMENT
      405  CALL DOME(IT(N),T(N),R(N),H(N),HO(N),E(N),PR(N),UW(N),S,TS
          *,ANG,ANGO,RLAM,IFLAG)
           PST=PBF(5,N)
           CALL PDOME(IT(N),T(N),R(N),H(N),HO(N),E(N),PR(N),UW(N),ALPHA(N),
          *  S,PSD,IP(N),PV(N),IDV,PSF,ANG,ANGO,WHT(N),PST,IFLAG,N)
     C
           IF (IPRINT.EQ.0) GO TO 420
           WRITE(6,2700) N,((S(I,J),J=1,4),I=1,4)
     2700  FORMAT(///'  FLEXIBILITY MATRIX FOR DOME SEGMENT',I4/
          *  (4E16.5))
           GOTO 420
     C
     C  ELASTIC FOUNDATION SEGMENTS
      410  CALL BASE(IFLAG,T(N),R(N),H(N),HO(N),E(N),PR(N),UW(N),BB,S,TS,D)
           CALL PBASE(T(N),R(N),H(N),HO(N),E(N),PR(N),ALPHA(N),UW(N),S,PSD,
          *   IP(N),PV(N),N,PSF,PBF)
     C
     C  SPECIAL ASSEMBLY FOR BASE ELEMENTS
     C
           DO 415  I=1,6
           L=IDF(N,I)
           IF(L.EQ.0) GO TO 415
```

```
      PARD(L)=PSD(I)
      DO 412 J=1,6
      K=IDF(N,J)
      IF(K.EQ.0) GO TO 412
      F(L,K)=S(I,J)
412   CONTINUE
  415 CONTINUE
      IF (IPRINT.EQ.0) GO TO 500
      WRITE(6,2720) N,S
2720  FORMAT(///'  FLEXIBILITY MATRIX FOR BASE SEGMENT',I4/
     * (6E15.5))
      GO TO 500
C
C  CONICAL SEGMENT
 470  CALL CONE(IT(N),T(N),R(N),H(N),HO(N),E(N),PR(N),UW(N),S,TS
     *,ANG,XM,RLAM,IFLAG)
      PST=PSF(S,N)
      CALL PCONE(IT(N),T(N),R(N),H(N),HO(N),E(N),PR(N),UW(N),ALPHA(N),
     *  S,PSD,IP(N),PV(N),IDV,PSP,ANG,WHT(N),PST,IFLAG,N)
C
      IF (IPRINT.EQ.0) GO TO 420
      WRITE(6,2750) N,((S(I,J),J=1,4),I=1,4)
2750  FORMAT(///'  FLEXIBILITY MATRIX FOR CONE SEGMENT',I4/
     * (4E15.5))
C
C  ASSEMBLY OF FLEXIBILITY MATRIX AND DISPLACEMENT VECTOR (PARD)
  420 DO 460  I=1,4
      L=IDF(N,I)
      IF(L.EQ.0) GO TO 460
      PARD(L)=PSD(I)
      DO 450 J=1,4
      K=IDF(N,J)
      IF(K.EQ.0) GO TO 450
      F(L,K)=S(I,J)
450   CONTINUE
  460 CONTINUE
500   CONTINUE
C
      IF (IPRINT.EQ.0) GO TO 508
      WRITE(6,2800)
2800  FORMAT(///'  ELEMENT FLEXIBILITIES AFTER ASSEMBLY')
      DO 505  I=1,NF
      WRITE(6,2850) (F(I,J),J=1,NF)
505   CONTINUE
2850  FORMAT(6E12.4)
C
C  CONDENSE TO REDUNDANT FLEXIBILITY MATRIX AND DISPLACEMENT VECTOR
508   DO 520 I=1,NF
      DO 520 J=1,NR
      C=0.0
      DO 510 K=1,NF
510   C=C+F(I,K)*A(J,K)
520   TT(I,J)=C
C
      DO 650 I=1,NR
      C=0.0
      DO 610 K=1,NF
610   C=C+A(I,K)*PARD(K)
      PART(I)=-C
C
      DO 630 J=1,NR
      C=0.0
      DO 620 K=1,NF
620   C=C+A(I,K)*TT(K,J)
630   F(I,J)=C
650   CONTINUE
C
      IF (IPRINT.EQ.0) GO TO 570
      WRITE(6,2900)
2900  FORMAT(///'  CONDENSED FLEXIBILITY MATRIX')
      DO 660  I=1,NR
      WRITE(6,2850) (F(I,J),J=1,NR)
660   CONTINUE
      WRITE(6,3001) (I,PART(I),I=1,NR)
3001  FORMAT(///' INCOMPATIBLE DISPLACEMENTS'/' IDISP      VALUE'/
     * (I5,E13.5))
C-------------------------------------------------------------------
C  SOLVE FOR REDUNDANTS AND FIND SEGMENT END FORCES
C-------------------------------------------------------------------
570   CALL SOL(F,PART,SO,NR)
C
C  FIND SEGMENT FORCES
      DO 700 I=1,NF
      C=0.0
      DO 690 J=1,NR
690   C=C+A(J,I)*PART(J)
700   FO(I)=C
C
C  WRITE TOTAL VECTOR OF SEGMENT END FORCES
      WRITE(6,3100)
3100  FORMAT(///'FORCES ON ENDS OF SEGMENTS'/' SEG   J  IF',8X,
     * 'FORCE')
      DO 706 N=1,NSEG
      DO 705 J=1,6
      L=IDF(N,J)
      IF(L.EQ.0) GO TO 705
      WRITE(6,3200) N,J,L,FO(L)
705   CONTINUE
  706 CONTINUE
3200  FORMAT(3I4,E13.5)
C-------------------------------------------------------------------
C  FIND AND OUTPUT SEGMENT STRESS RESULTANTS
C-------------------------------------------------------------------
C
      DO 800 N=1,NSEG
      IFLAG=1
      ITN = IT(N)
      IPN=IP(N)
      IF(IT(N).EQ.3) GO TO 800
C
C   FORM SEGMENT END FORCE CVECTOR
      IF(NDIV(N).GT.100)  GO TO 999
      DH=(H(N)-HO(N))/DFLOAT(NDIV(N))
```

```
      NDIV1=NDIV(N)+1
      DO 710 I=1,4
      SF(I)=PBF(N,I)
      L=IBF(N,I)
      IF(L.EQ.0) GO TO 710
      SF(I)=FO(L)+SF(I)
  710 CONTINUE
      SF(5)=PBF(5,N)
      SF(6)=PBF(6,N)
C
C WRITE  INDIVIDUAL SEGMENT END FORCES
      WRITE(6,3300) N,(SF(I),I=1,6)
 3300 FORMAT (///' INDIVIDUAL END FORCES FOR SEGMENT ',I6,/(4E13.6))
C
      GOTO (705,750,800,870,750,870),ITN
C---------------------------------------------------------------------
C   STRESS RESULTANTS FOR CYLINDRICAL SEGMENTS
C---------------------------------------------------------------------
  705 CALL CYLIN(T(N),R(N),H(N),ND(N),E(N),PR(N),UW(N),S,TS,D,BETA,
     *    IFLAG)
      DN1=0.0
      CM1=0.0
      CM2=0.0
      CN1=-PBF(5,N)
      WP=CN1*R(N)*PR(N)/(E(N)*T(N))
      CN2=0.0
      WW=0.0
      WL=0.0
      RN=E(N)*T(N)/R(N)
      GOTO (711,712,711,713,714,714,715),IPN
  711 CN2=-PV(N)*R(N)
      WP=PV(N)*R(N)**2/(E(N)*T(N))+WP
      GOTO 721
  712 DN1=-T(N)*UW(N)*DH
      WW=DN1*R(N)*PR(N)/(E(N)*T(N))
      GOTO 721
  713 WP=-R(N)*ALPHA(N)*PV(N)+WP
      GOTO 721
  714 CM1=(1.+PR(N))*D*ALPHA(N)*PV(N)
      CM2=CM1
      GO TO 721
  715 WL=PV(N)*R(N)**2/(E(N)*T(N))
  721 D=2.*D*BETA**2
      DO 730 I=1,4
      C=0.0
      DO 720 J=1,4
  720 C=C+TS(I,J)*SF(J)
  730 CVEC(I)=C
C
      X=0.0
      DO 740 L=1,NDIV1
      BX=BETA*X
      DC=DCOS(BX)
      DS=DSIN(BX)
C
      W(L)=WP+DEXP(BX)*(CVEC(1)*DC+CVEC(2)*DS)+DEXP(-BX)*
     *    (CVEC(3)*DC+CVEC(4)*DS)+WW*DFLOAT(L-1)+WL*X
C
      RN1(L)=CN1+DN1*DFLOAT(L-1)
      RN2(L)=CN2-(DEXP(BX)*(CVEC(1)*DC+CVEC(2)*DS)+
     *DEXP(-BX)*(CVEC(3)*DC+CVEC(4)*DS)+WL*X)*RN
      RM1(L)=DEXP(BX)*(D*CVEC(2)*DC-D*CVEC(1)*DS)+DEXP(-BX)*(D*
     *  CVEC(3)*DS-D*CVEC(4)*DC)
      RM2(L)=PR(N)*RM1(L)+CM2
      RM1(L)=RM1(L)+CM1
      XH(L)=X
  740 X=X+DH
C
      WRITE(6,4000) N,(L,XH(L),RN1(L),RN2(L),RM1(L),RM2(L),L=1,NDIV1)
      WRITE(6,4005) (L,XH(L),W(L),L=1,NDIV1)
      GOTO 900
C---------------------------------------------------------------------
C   STRESS RESULTANTS FOR DOME SEGMENTS
C---------------------------------------------------------------------
  750 CALL DOME(IT(N),T(N),R(N),H(N),ND(N),E(N),PR(N),UW(N),S,TS
     *,ANG,ANGO,RLAM,IFLAG)
      X=0.
      DX=DH/RAD
      D=E(N)*T(N)**3/(12.*(1.-PR(N)**2))
C
      DO 787 I=1,4
      C=0.0
      DO 785 J=1,4
  785 C=C+TS(I,J)*SF(J)
  787 CVEC(I)=C
C
      DO 780 L=1,NDIV1
C
      IF(IT(N).EQ.2)  PHI = X+ANGO
      IF(IT(N).EQ.5)  PHI = ANG-X
      CADX=DCOS(PHI)
      SADX=DSIN(PHI)
C
      DCO = DCOS(RLAM*PHI)
      DSI = DSIN(RLAM*PHI)
      EP  = DEXP(RLAM*PHI)
      EM  = DEXP(-RLAM*PHI)
      TH1 = EP*(DCO+DSI)
      TH2 = EP*(DCO-DSI)
      TH3 = EM*(DCO+DSI)
      TH4 = EM*(DCO-DSI)
      PHI1= EP*DCO
      PHI2= EP*DSI
      PHI3= EM*DCO
      PHI4= EM*DSI
C
      WP=0.
      RM1(L) = 0
      RM2(L) = 0
      RN2(L) = FN2(PV(N),SF(5),UW(N),R(N),T(N),IP(N),
     *              IT(N),PHI,ANGO,ANG,WHT(N),IDV)
C
      GOTO (753,753,753,754,760,753,753),IPN
```

```
      753  RM1(L) = FN1(PV(N),SF(5),UW(N),R(N),T(N),IP(N),
          *            IT(N),PHI,ANGO,ANG,WHT(N),IDV)
           WP = -R(N)*SAOX*(RM2(L)-PR(N)*RM1(L))/(E(N)*T(N))
           GO TO 755
      754  WP=-ALPHA(N)*R(N)*SAOX*PV(N)
           GO TO 756
      780  RM1(L) = PV(N)*ALPHA(N)*D*(1.+PR(N))/T(N)
           RM2(L) = RM1(L)
     C
      755  W(L) = WP + R(N)*SAOX*RLAM*(CVEC(1)*TH2 + CVEC(2)
          *         *TH1 - CVEC(3)*TH3 + CVEC(4)*TH4)/(E(N)*T(N))
     C
           RM2(L) = RM2(L) - RLAM*(CVEC(1)*TH2 +
          *                      CVEC(2)*TH1 - CVEC(3)*TH3 + CVEC(4)*TH4)
     C
      758  IF(PHI.EQ.0.)  GO TO 761
           RM1(L) = RM1(L) - (CAOX/SAOX)*(CVEC(1)*PHI1
          *                 + CVEC(2)*PHI2 + CVEC(3)*PHI3 +CVEC(4)*PHI4)
     C
      761  RM1(L) =  RM1(L) + 0.5*R(N)*(-CVEC(1)*TH1 + CVEC(2)*TH2
          *                 + CVEC(3)*TH4 + CVEC(4)*TH3)/RLAM
     C
           RM2(L) =  RM2(L) + PR(N)*RM1(L)
     C
       771 X=X+DX
           XN(L)=DFLOAT(L-1)*DH
     7800  CONTINUE
     C
           WRITE(6,4500) N,(L,XN(L),RN1(L),RN2(L),RM1(L),RM2(L),L=1,NDIV1)
           WRITE(6,4005) (L,XN(L),W(L),L=1,NDIV1)
           GOTO 900
     C
     C----------------------------------------------------------
     C  STRESS RESULTANTS FOR BASE-SLAB ELEMENT
     C----------------------------------------------------------
     C  FORM SEGMENT END FORCE VECTOR
      800  IF (NDIV(N).GT.100) GO TO 898
           NDIV1=NDIV(N)+1
           DO 805 I=1,6
           SF(I)=FSF(N,I)
           L=IDF(N,I)
           IF(L.EQ.0) GOTO 805
           SF(I)=FD(L)+SF(I)
      805  CONTINUE
     C
     C   WRITE INDIVIDUAL SEGMENT END FORCES
           WRITE(6,5000) N,(SF(I),I=1,6)
     5000  FORMAT(///'    INDIVIDUAL END FORCES FOR',
          *  ' SEGMENT',I5,/(4E13.5))
     C
     C  COMPUTE STRESS RESULTANTS
           IFLAG=1
           CALL BASE(IFLAG,T(N),R(N),H(N),HO(N),E(N),PR(N),UW(N),SB,S,TS,D)
           DATA IVECT/2,5,4,6/
           DO 820 I=1,4
           C=0.0
           DO 815 J=1,4
           JJ=IVECT(J)
      815  C=C+TS(I,J)*SF(JJ)
      820  CVEC(I)=C
           HT=H(N)
     C
           IFLAG=2
           DO 825 I=1,4
           DO 825 J=1,4
      825  SB(I,J)=0.0
           DO 850 L=1,NDIV1
           CALL BASE(IFLAG,T(N),R(N),HT,HO(N),E(N),PR(N),UW(N),SB,S,TS,D)
           DO 830,I=1,4
           CC=0.0
           DO 835 J=1,4
      835  CC=CC+SB(I,J)*CVEC(J)
      830  SR(I)=CC
           HR(L)=HT
           DH=(H(N)-HO(N))/DFLOAT(NDIV(N))
           HT=HT-DH
     C
           RM1(L)=SR(2)
           RM2(L)=SR(1)
           V(L)=SR(3)
           W(L)=SR(4)
           GOTO (840,841,850,850,850,850,840),IPN
      840  W(L)=W(L)+PV(N)/R(N)
           GOTO 850
      841  W(L)=W(L)+UW(N)*T(N)/R(N)
           GOTO 850
      850  CONTINUE
           WRITE(6,5500) N,(L,HR(L),RM1(L),RM2(L),V(L),L=1,NDIV1)
           WRITE(6,5505) (L,HR(L),W(L),L=1,NDIV1)
           GO TO 900
     C
     C---------------------------------------------------------------------
     C   STRESS RESULTANTS FOR CONE SEGMENTS
     C---------------------------------------------------------------------
      870  CALL CONE(IT(N),T(N),R(N),H(N),HO(N),E(N),PR(N),UW(N),S,TS
          *,ANG,XM,RLAM,IFLAG)
     C
           DO 872 I=1,4
           C = 0.
           DO 871 J=1,4
      871  C = C + TS(I,J)*SF(J)
      872  CVEC(I) = C
     C
           X=0.
           D = E(N)*T(N)**3/(12.*(1.-PR(N)**2))
     C
           DO 890 L=1,NDIV1
     C
           WP = 0.
           W(L) = 0.
           RN1(L) = 0.
           RN2(L) = 0.
           RM1(L) = 0.
```

```
            RM2(L) = 0.
      C
            Y = X + HO(N)
            IF(ITN.EQ.6)  Y = H(N) - X
            IF(Y.EQ.0.)  GO TO 884
      C
            XI = 2.*RLAM*Y**0.5
            CALL MMKEL2(XI,BER,BEI,XKER,XKEI,
           *            DBER,DBEI,DKER,DKEI)
            G1 = XI*DBER - 2.*PR(N)*BER
            G2 = XI*DBEI - 2.*PR(N)*BEI
            G3 = XI*DKER + 2.*PR(N)*XKER
            G4 = XI*DKEI - 2.*PR(N)*XKEI
      C
            U1 = G1 + 4.*PR(N)*BER
            U2 = G2 + 4.*PR(N)*BEI
            U3 = G3 + 4.*PR(N)*XKER
            U4 = G4 + 4.*PR(N)*XKEI
      C
            V1 = 2.*BER + PR(N)*XI*DBER
            V2 = 2.*BEI + PR(N)*XI*DBEI
            V3 = 2.*XKER + PR(N)*XI*DKER
            V4 = 2.*XKEI + PR(N)*XI*DKEI
      C
        880 GOTO (874,874,874,875,876,874,874),IPN
        874 RN1(L) = FN3(IP(N),IT(N),Y,ANG,H(N),HO(N),
           *            WHT(N),PV(N),PBF(5,N),UW(N),T(N),IDV)
            RN2(L) = FN4(IP(N),IT(N),Y,ANG,H(N),HO(N),
           *            WHT(N),PV(N),PBF(5,N),UW(N),T(N))
            WP = -Y*DSIN(ANG)*(RN2(L)-PR(N)*RN1(L))/(E(N)*T(N))
            GO TO 879
        875 WP = PV(N)*ALPHA(N)*Y*DSIN(ANG)*E(N)*T(N)
            GO TO 879
        876 RM1(L) = PV(N)*ALPHA(N)*D*(1.+PR(N))/T(N)
            RM2(L) = RM1(L)
      C
        879 W(L) = (WP + 0.5*DSIN(ANG)*(CVEC(1)*G1 + CVEC(2)*G2 +
           *        CVEC(3)*G3 + CVEC(4)*G4))/(E(N)*T(N))
      C
            RN1(L) = RN1(L) - (CVEC(1)*BER + CVEC(2)*BEI +
           *CVEC(3)*XKER + CVEC(4)*XKEI)/Y
      C
            RN2(L) = RN2(L) - 0.5*XI*(CVEC(1)*DBER + CVEC(2)
           **DBEI + CVEC(3)*DKER + CVEC(4)*DKEI)/Y
      C
            RM1(L) = RM1(L) - T(N)*(CVEC(1)*U2 - CVEC(2)*U1 +
           *CVEC(3)*U4 - CVEC(4)*U3)/(2.*XM**2*Y)
      C
            RM2(L) = RM2(L) - T(N)*(CVEC(1)*V2 - CVEC(2)*V1 +
           *CVEC(3)*V4 - CVEC(4)*V3)/(2.*XM**2*Y)
      C
        884 X = X + DH
            XH(L) = DFLOAT(L-1)*DH
        890 CONTINUE
      C
            WRITE(6,4510) N,(L,XH(L),RN1(L),RN2(L),RM1(L),RM2(L),L=1,NDIV1)
            WRITE(6,4005) (L,XH(L),W(L),L=1,NDIV1)
        900 CONTINUE
            STOP
        999 WRITE(6,3000)
       3000 FORMAT(' STOP FOR PROGRAM DIAGNOSED INPUT ERROR ')
            STOP
      C-----------------------------------------------------------------
      C     FORMAT STATEMENTS
      C-----------------------------------------------------------------
       1001 FORMAT(10A8)
       2004 FORMAT('1',10A8//)
       1000 FORMAT(5I4,2F10.4)
       2000 FORMAT('1','  ****  OUTPUT FOR FLEXIBILITY ANALYSIS OF SEGMENTED',
           *' SHELL  ****'/,
           *'          NUMBER OF SEGMENTS          =',I4/
           *'          IPRINT                      =',I4///
           *' SEG  TYPE  IR   JR  NDIV',4X,'EC1',7X,'EC2')
       2100 FORMAT(I4,4I5,2F10.4)
       1200 FORMAT(2I4)
       2200 FORMAT(///,'   CONNECTIVITY MATRIX'/(8X,2I4))
       1300 FORMAT(I4,F8.0,F12.0,F8.0,5F10.0)
       2300 FORMAT(///'   GEOMETRIC PARAMETERS'/'  SEG',4X,'THICK',3X,'RADIUS'
           *,3X,'L OR ANG',4X,'ANGO',7X,'MODULUS',6X,'P RATIO',2X,'THERMCOEF',
           * 3X,'WEIGHT'/ (I4,F8.3,F12.3,2F10.3,E13.4,F10.3,E13.4,F10.3))
       1400 FORMAT(2I4,8F10.0)
       2400 FORMAT(///'  PARTICULAR SOLUTION INPUT INFORMATION'/
           *' SEG  TYPE   VALUE',4X,'LIQ HT',2X,'TOP SHEAR',4X,'TOP MOM',
           * 6X,'BOT SHEAR',4X,'BOT MOM',4X,'TOP FORCE',4X,'VERT FORCE'/
           * (I4,I5,F10.3,F10.3,6E13.5))
       4000 FORMAT(///'  ***  OUTPUT FOR CYLINDRICAL SEGMENT',I4,' ***'/
           *' POINT    COORD          N1',12X,'N2',12X,'M1',12X,'M2',/
           *    (I6,F10.4,4E14.5))
       4005 FORMAT(///' *** HORIZONTAL DISPLACEMENT ***'/
           *  ' POINT    COORD          W '/(I6,F10.4,E14.5))
       4500 FORMAT(///'  ***  OUTPUT FOR DOME SEGMENT',I4,' ***'/
           *' POINT    ANGLE          N1',12X,'N2',12X,'M1',12X,'M2',/
           *    (I6,F10.4,4E14.5))
       4510 FORMAT(///'  ***  OUTPUT FOR CONE SEGMENT',I4,' ***'/
           *' POINT    COORD          N1',12X,'N2',12X,'M1',12X,'M2',/
           *    (I6,F10.4,4E14.5))
       5500 FORMAT(///'  *** OUTPUT FOR BASE ELEMENT',I4,'    ***'/
           *  '  POINT      COORD          M1',12X,'M2',12X,'V',12X,
           * /(I6,F10.4,3E14.5))
       5505 FORMAT(///' *** VERTICAL DISPLACEMENT ***'/
           * 'POINT    COORD          W'/(I6,F10.4,E14.5))
            END
      C*****************************************************************
            FUNCTION FN1(PV,PBT,UW,R,T,IP,IT,PHI,ANGO,ANG,WHT,IDV)
      C  THIS FUNCTION IS USED FOR N1 DOME STRESS RESULTANTS
            IMPLICIT REAL*8(A-H,O-Z)
            FN1=0.0
            C1 = 1.
            GAMMA = ANGO
            IF(IT.NE.5)  GO TO 5
            C1 = -1.
            GAMMA = ANG
          5 GOTO(10,20,10,100,100,15,30),IP
```

```
   10 FN1=-0.5*PV*R
      IF(ANGO.LE.1.0E-03) RETURN
      FN1 = FN1*(1.-(DSIN(GAMMA)/DSIN(PHI))**2)
      GO TO 100
   15 FN1=-0.5*PV*R
      IF(ANGO.LE.1.0E-3)  RETURN
      FN1 = C1*FN1*(1.-(DSIN(GAMMA)/DSIN(PHI))**2)
      GO TO 100
   20 FN1 = -0.5*UW*T*R
      IF (PHI.EQ.0.)  RETURN
      FN1 = C1*2.*FN1*(DCOS(GAMMA)-DCOS(PHI))/DSIN(PHI)**2
      GO TO 100
   30 IF(PHI.EQ.0.)  RETURN
      CONST1 = (DCOS(PHI)**3-(DCOS(GAMMA)**3))/DSIN(PHI)**2
      CONST2 = (DSIN(GAMMA)/DSIN(PHI))**2
      FN1 = -C1*PV*R*(0.5*(WHT+C1*R*DCOS(GAMMA))*(1.-CONST2)
     *        +C1*R*CONST1/3.)
  100 IF(PHI.EQ.0.)  RETURN
      IF(IDV.EQ.0)  RETURN
      FN1 = FN1 - PBT*DSIN(GAMMA)/DSIN(PHI)**2
      RETURN
      END
C*************************************************************************
      FUNCTION FN2(PV,PBT,UW,R,T,IP,IT,PHI,ANGO,ANG,WHT,IDV)
C  THIS FUNCTION IS USED FOR N2 DOME STRESS RESULTANTS
      IMPLICIT REAL*8(A-H,O-Z)
      FN2=0.0
      C = 1.
      GAMMA = ANGO
      IF(IT.EQ.2)  GO TO 5
      C = -1.
      GAMMA = ANG
    5 GOTO(10,20,10,100,100,15,30),IP
   10 FN2=-0.5*PV*R
      IF(ANGO.LE.1.0E-03) RETURN
      FN2 = FN2*(1.+(DSIN(GAMMA)/DSIN(PHI))**2)
      GO TO 100
   15 FN2 = -C*.5*PV*R
      IF(PHI.EQ.0.) RETURN
      FN2 = C*0.5*PV*R*(1.-(DSIN(GAMMA)/DSIN(PHI))**2
     *     - C*2.*DCOS(PHI)**2)
      GO TO 100
   20 FN2 = -C*0.5*UW*T*R
      IF (PHI.EQ.0.) RETURN
      FN2 = UW*T*R*((DCOS(GAMMA)-DCOS(PHI))/DSIN(PHI)**2
     *     - C*DCOS(PHI))
      GO TO 100
   30 IF(PHI.EQ.0.) RETURN
      CONST1 = (DCOS(PHI)**3-DCOS(GAMMA)**3)/DSIN(PHI)**2
      CONST2 = (DSIN(GAMMA)/DSIN(PHI))**2
      FN2 = -C*PV*R*(0.5*(WHT+C*R*DCOS(GAMMA))*(1.+CONST2)
     *       -C*R*(CONST1/3.+DCOS(PHI)))
  100 IF(PHI.EQ.0.)  RETURN
      IF (IDV.EQ.0) RETURN
      FN2 = FN2 + PBT*DSIN(GAMMA)/DSIN(PHI)**2
      RETURN
      END
C*************************************************************************
      FUNCTION FN3(IP,IT,Y,ANG,H,HO,WHT,PV,PBT,UW,T,IDV)
C  THIS FUNCTION IS USED FOR N1 CONE STRESS RESULTANTS
      IMPLICIT REAL*8(A-H,O-Z)
      FN3 = 0.
      IF(Y.EQ.0.)  RETURN
      Y1 = HO
      C1 = 1.
      IF(IT.NE.6)  GO TO 5
      C1 = -1.
      Y1 = H
    5 C = -C1*0.5*(Y**2-Y1**2)/Y
      GOTO(10,20,10,100,100,10,40),IP
   10 FN3 = PV*C*DTAN(ANG)
      GO TO 100
   20 FN3 = UW*T*C/DCOS(ANG)
      GO TO 100
   40 FN3 = -PV*Y*DTAN(ANG)*(3.*WHT*(1.-(Y1/Y)**2)
     *       +C1*2.*Y*DCOS(ANG)*(1.-(Y1/Y)**3))/6.
  100 IF(IDV.EQ.0) RETURN
      FN3 = FN3 - PBT*Y1/(Y*DCOS(ANG))
      RETURN
      END
C*************************************************************************
      FUNCTION FN4(IP,IT,Y,ANG,H,HO,WHT,PV,PBT,UW,T)
C  THIS FUNCTION IS USED FOR N2 CONE STRESS RESULTANTS
      IMPLICIT REAL*8(A-H,O-Z)
      FN4 = 0.
      C1 = 1.
      C = -Y*DTAN(ANG)
      IF(IT.EQ.6) C1 = -1
      GOTO(10,20,10,100,100,30,40),IP
   10 FN4 = -C*PV
      RETURN
   20 FN4 = UW*T*C*C1*DSIN(ANG)
      RETURN
   30 FN4 = PV*C*C1*DSIN(ANG)**2
      RETURN
   40 FN4 = -PV*Y*DTAN(ANG)*(WHT+C1*Y*DCOS(ANG))
  100 RETURN
      END
C*************************************************************************
      SUBROUTINE PFOR(IT,T,R,H,HO,WHT,E,PR,UW,ALPHA,IP,PV,PBT,PBF,IDV,N)
C  THIS SUBROUTINE COMPUTES PARTICULAR SOLUTION EDGE FORCES (PBF)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION PBF(6,20)
C
C  SELECT SEGMENT TYPE
      GO TO (50,500,600,700,500,700),IT
C
C  CYLINDRICAL SEGMENTS
   50 GO TO (900,100,900,900,650,900,900),IP
C  DEAD LOAD
  100 PBF(5,N)=PBF(5,N)+T*UW*H
      RETURN
C
```

```
C  SPHERICAL SEGMENTS
600   ANG=H/57.295779513
      ANGO=HO/57.295779513
      PHI=ANG
      C1 = 1.
      IF(IT.EQ.2)  GO TO 605
      PHI=ANGO
      C1 = -1.
605   RN1 = FN1(PV,PST,UW,R,T,IP,IT,PHI,ANGO,ANG,WHT,IDV)
      GOTO (610,610,610,800,850,610,610),IP
610   PBF(6,N) = PBF(6,N) - RN1*DSIN(PHI)
      PBF(3,N) = PBF(3,N) + C1*RN1*DCOS(PHI)
      RETURN
C
C  BASE ON ELASTIC FOUNDATION
800   GOTO (900,900,900,900,850,900,900),IP
C
C  THERMAL GRADIENT
850   C = 1.
      IF(IT.GT.4)  C=-1.
      PBF(4,N)=PBF(4,N)-C*E*PV*ALPHA*T**2/(12.*(1.-PR))
      PBF(2,N)=-PBF(4,N)
      RETURN
C
C  CONE SEGMENTS
700   ANG=R/57.295779513
      Y = H
      C1 = 1.
      IF (IT.NE.6)  GO TO 705
      Y = HO
      C1 = -1.
705   RN1 = FN3(IP,IT,Y,ANG,H,HO,WHT,PV,PST,UW,T,IDV)
C
      GOTO (710,710,710,800,850,710,710),IP
710   PBF(6,N) = PBF(6,N) - RN1*DCOS(ANG)
      PBF(3,N) = PBF(2,N) + C1*RN1*DSIN(ANG)
900   RETURN
      END
C*****************************************************************
      SUBROUTINE CYLIN(T,R,H,HO,E,PR,UW,F,TT,D,BETA,IFLAG)
C  THIS SUBROUTINE COMPUTES THE CYLINDER FLEXIBILITY (F)
C  AND  MATRICES (TT).
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION F(6,6),TT(4,4),TA(4,4)
C
      D=E*T**3/(12.*(1.-PR**2))
      BETA=(3.*(1.-PR**2)/(R*T)**2)**.25
      PHI1=DEXP(BETA*H)*DCOS(BETA*H)
      PHI2=DEXP(BETA*H)*DSIN(BETA*H)
      PHI3=DEXP(-BETA*H)*DCOS(BETA*H)
      PHI4=DEXP(-BETA*H)*DSIN(BETA*H)
      TH1=PHI1+PHI2
      TH2=PHI1-PHI2
      TH3=PHI3+PHI4
      TH4=PHI3-PHI4
C
      TT(1,1) =  -2.*D*BETA**3
      TT(1,2) =   2.*D*BETA**3
      TT(1,3) =   2.*D*BETA**3
      TT(1,4) =   2.*D*BETA**3
C
      TT(2,1) =   0.
      TT(2,2) =  -2.*D*BETA**2
      TT(2,3) =   0.
      TT(2,4) =   2.*D*BETA**2
C
      TT(3,1) = TH1*2.*D*BETA**3
      TT(3,2) =-TH2*2.*D*BETA**3
      TT(3,3) =-TH4*2.*D*BETA**3
      TT(3,4) =-TH3*2.*D*BETA**3
C
      TT(4,1) =-PHI2*2.*D*BETA**2
      TT(4,2) = PHI1*2.*D*BETA**2
      TT(4,3) = PHI4*2.*D*BETA**2
      TT(4,4) =-PHI3*2.*D*BETA**2
      CALL TTINV(TT,H)
C
      IF(IFLAG.NE.0)  RETURN
C
      TA(1,1)=1.0
      TA(1,2)=0.0
      TA(1,3)=1.0
      TA(1,4)=0.0
      TA(2,1)=BETA
      TA(2,2)=BETA
      TA(2,3)=-BETA
      TA(2,4)=BETA
      TA(3,1)=PHI1
      TA(3,2)=PHI2
      TA(3,3)=PHI3
      TA(3,4)=PHI4
      TA(4,1)=TH2*BETA
      TA(4,2)=TH1*BETA
      TA(4,3)=-TH3*BETA
      TA(4,4)=TH4*BETA
C
      DO 100 I=1,4
      DO 100 J=1,4
      C=0.0
      DO 80 K=1,4
80    C=C+TA(I,K)*TT(K,J)
100   F(I,J)=C
C
      RETURN
      END
C*****************************************************************
      SUBROUTINE DOMEFIT(T,R,H,HO,E,PR,UW,F,TT,ANG,ANGO,RLAM,IFLAG)
C  THIS SUBROUTINE COMPUTES FLEXIBILITY MATRICES (F) FOR
C  A COMPLETE OR TRUNCATED SPHERE. THE FLEXIBILITY MATRIX
C  IS REDUCED TO A TWO BY TWO MATRIX FOR A COMPLETE SPHERE.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION F(6,6),TT(4,4),TA(4,4)
```

```
      C
            ANG=H/57.295779513
            ANGO=HO/57.295779513
            RLAM=(3.*(1.-PR**2)*(R/T)**2)**0.25
            PHI10 = DEXP(RLAM*ANGO)*DCOS(RLAM*ANGO)
            PHI20 = DEXP(RLAM*ANGO)*DSIN(RLAM*ANGO)
            PHI30 = DEXP(-RLAM*ANGO)*DCOS(RLAM*ANGO)
            PHI40 = DEXP(-RLAM*ANGO)*DSIN(RLAM*ANGO)
            PHI11 = DEXP(RLAM*ANG)*DCOS(RLAM*ANG)
            PHI21 = DEXP(RLAM*ANG)*DSIN(RLAM*ANG)
            PHI31 = DEXP(-RLAM*ANG)*DCOS(RLAM*ANG)
            PHI41 = DEXP(-RLAM*ANG)*DSIN(RLAM*ANG)
            TH10  = PHI10 + PHI20
            TH20  = PHI10 - PHI20
            TH30  = PHI30 + PHI40
            TH40  = PHI30 - PHI40
            TH11  = PHI11 + PHI21
            TH21  = PHI11 - PHI21
            TH31  = PHI31 + PHI41
            TH41  = PHI31 - PHI41
            IF (HO.NE.0.)  GO TO 10
      C
      C  INITIALIZE THE MATRICES TO ZERO.
            DO 5 I=1,4
            DO 5 J=1,4
            TA(I,J) = 0.
          5 TT(I,J) = 0.
            GO TO 15
         10 TT(1,1) = PHI10/(-DSIN(ANGO))
            TT(1,2) = PHI20/(-DSIN(ANGO))
            TT(1,3) = PHI30/(-DSIN(ANGO))
            TT(1,4) = PHI40/(-DSIN(ANGO))
      C
            TT(2,1) = -TH10*(-.5*R/RLAM)
            TT(2,2) = TH20*(-.5*R/RLAM)
            TT(2,3) = TH40*(-.5*R/RLAM)
            TT(2,4) = TH30*(-.5*R/RLAM)
      C
            TT(3,3) = PHI31/DSIN(ANG)
            TT(3,4) = PHI41/DSIN(ANG)
            TT(4,3) = TH41*.5*R/RLAM
            TT(4,4) = TH31*.5*R/RLAM
      C
         15 TT(3,1) = PHI11/DSIN(ANG)
            TT(3,2) = PHI21/DSIN(ANG)
            TT(4,1) =-TH11*.5*R/RLAM
            TT(4,2) = TH21*.5*R/RLAM
            IF(IT.EQ.5) CALL ROWEX(TT)
         20 CALL TTINV(TT,HO)
      C
            IF(IFLAG.NE.0) RETURN
            IF(HO.EQ.0.) GO TO 30
      C
            TA(1,1) = TH20*(R*RLAM*DSIN(ANGO)/(E*T))
            TA(1,2) = TH10*(R*RLAM*DSIN(ANGO)/(E*T))
            TA(1,3) =-TH30*(R*RLAM*DSIN(ANGO)/(E*T))
            TA(1,4) = TH40*(R*RLAM*DSIN(ANGO)/(E*T))
            TA(2,1) =-PHI20*(2.*RLAM**2/(E*T))
            TA(2,2) = PHI10*(2.*RLAM**2/(E*T))
            TA(2,3) = PHI40*(2.*RLAM**2/(E*T))
            TA(2,4) =-PHI30*(2.*RLAM**2/(E*T))
            TA(3,3) =-TH31*(R*RLAM*DSIN(ANG)/(E*T))
            TA(3,4) = TH41*(R*RLAM*DSIN(ANG)/(E*T))
            TA(4,3) = PHI41*(2.*RLAM**2/(E*T))
            TA(4,4) =-PHI31*(2.*RLAM**2/(E*T))
         30 TA(3,1) = TH21*(R*RLAM*DSIN(ANG)/(E*T))
            TA(3,2) = TH11*(R*RLAM*DSIN(ANG)/(E*T))
            TA(4,1) =-PHI21*(2.*RLAM**2/(E*T))
            TA(4,2) = PHI11*(2.*RLAM**2/(E*T))
            IF(IT.EQ.5) CALL ROWEX(TA)
      C
            DO 100 I=1,4
            DO 100 J=1,4
            C=0.0
            DO 80 K=1,4
      80    C=C+TA(I,K)*TT(K,J)
      100   F(I,J)=C
      C
            RETURN
            END
      C=====================================================================
            SUBROUTINE CONE(IT,T,R,H,HO,E,PR,UW,F,TT,ANG,XM,RLAM,IFLAG)
      C  THIS SUBROUTINE CALCULATES THE FLEXIBILITY MATRIX (F) FOR
      C  A COMPLETE OR TRUNCATED CONE.  THE FLEXIBILITY MATRIX IS
      C  REDUCED TO A TWO BY TWO MATRIX FOR A COMPLETE CONE.
      C
            IMPLICIT REAL*8(A-H,O-Z)
            DIMENSION F(6,6),TA(4,4),TT(4,4)
      C
            ANG = R/57.295779513
            XM  = (12.*(1.-PR**2))**0.25
            RLAM= (XM**4/(T*DTAN(ANG))**2)**0.25
      C
            X1  = 2.*RLAM*(H**0.5)
            IF(X1.GT.119.)  GO TO 999
            CALL MMKEL2(X1,BER21,BEI21,XKER21,XKEI21,
           *DBER21,DBEI21,DKER21,DKEI21)
            IF (HO.NE.0.)  GO TO 10
      C
      C  INITIALIZE THE MATRICES TO ZERO.
            DO 5 I=1,4
            DO 5 J=1,4
            TA(I,J) = 0.
          5 TT(I,J) = 0.
            GO TO 15
      C
         10 XO  = 2.*RLAM*(HO**0.5)
            IF(XO.GT.119.)  GO TO 999
            CALL MMKEL2(XO,BER20,BEI20,XKER20,XKEI20,
           *DBER20,DBEI20,DKER20,DKEI20)
            TT(1,1) = BER20/(-HO*DSIN(ANG))
            TT(1,2) = BEI20/(-HO*DSIN(ANG))
            TT(1,3) = XKER20/(-HO*DSIN(ANG))
```

```
      TT(1,4) = XKEI20/(-HO*DSIN(ANG))
C
      TT(2,1) = (XO*DBEI20+2.*PR*BEI20)*T/(2.*XM**2*HO)
      TT(2,2) =-(XO*DBER20+2.*PR*BER20)*T/(2.*XM**2*HO)
      TT(2,3) = (XO*DKEI20+2.*PR*XKEI20)*T/(2.*XM**2*HO)
      TT(2,4) =-(XO*DKER20+2.*PR*XKER20)*T/(2.*XM**2*HO)
C
      TT(3,3) = XKER21/(H*DSIN(ANG))
      TT(3,4) = XKEI21/(H*DSIN(ANG))
      TT(4,3) = (X1*DKEI21+2.*PR*XKEI21)*T/(-2.*XM**2*H)
      TT(4,4) =-(X1*DKER21+2.*PR*XKER21)*T/(-2.*XM**2*H)
C
   15 TT(3,1) = BER21/(DSIN(ANG)*H)
      TT(3,2) = BEI21/(DSIN(ANG)*H)
      TT(4,1) = (X1*DBEI21+2.*PR*BEI21)*T/(-2.*XM**2*H)
      TT(4,2) =-(X1*DBER21+2.*PR*BER21)*T/(-2.*XM**2*H)
      IF(IT.EQ.6)  CALL ROWEX(TT)
   20 CALL TTINV(TT,HO)
C
      IF(IFLAG.NE.0) RETURN
      IF(HO.EQ.0.)  GO TO 30
      TA(1,1) = (XO*DBER20 - 2.*PR*BER20)/(2.*E*T/DSIN(ANG))
      TA(1,2) = (XO*DBEI20 - 2.*PR*BEI20)/(2.*E*T/DSIN(ANG))
      TA(1,3) = (XO*DKER20 - 2.*PR*XKER20)/(2.*E*T/DSIN(ANG))
      TA(1,4) = (XO*DKEI20 - 2.*PR*XKEI20)/(2.*E*T/DSIN(ANG))
      TA(2,1) = BEI20/(-E*T**2/XM**2)
      TA(2,2) =-BER20/(-E*T**2/XM**2)
      TA(2,3) = XKEI20/(-E*T**2/XM**2)
      TA(2,4) =-XKER20/(-E*T**2/XM**2)
      TA(4,3) = XKEI21/(-E*T**2/XM**2)
      TA(4,4) =-XKER21/(-E*T**2/XM**2)
      TA(3,3) = (X1*DKER21 - 2.*PR*XKER21)/(2.*E*T/DSIN(ANG))
      TA(3,4) = (X1*DKEI21 - 2.*PR*XKEI21)/(2.*E*T/DSIN(ANG))
   30 TA(3,1) = (X1*DBER21 - 2.*PR*BER21)/(2.*E*T/DSIN(ANG))
      TA(3,2) = (X1*DBEI21 - 2.*PR*BEI21)/(2.*E*T/DSIN(ANG))
      TA(4,1) = BEI21/(-E*T**2/XM**2)
      TA(4,2) =-BER21/(-E*T**2/XM**2)
      IF(IT.EQ.6)  CALL ROWEX(TA)
C
C  ASSEMBLE THE ELEMENT FLEXIBILITY MATRIX F.
C
      DO 100 I=1,4
      DO 100 J=1,4
      C=0.0
      DO 80 K=1,4
   80 C=C+TA(I,K)*TT(K,J)
  100 F(I,J)=C
      RETURN
  999 WRITE(6,1000)
 1000 FORMAT('  PROGRAM STOPPED FOR CONE',/,
     *       '  CHECK INPUT ARGUMENT FOR KELVIN FUNCTIONS')
      STOP
      END
C*********************************************************************
      SUBROUTINE MMKEL2(X,BER2,BEI2,XKER2,XKEI2,
     *DBER2,DBEI2,DKER2,DKEI2)
C
C  THIS SUBROUTINE CALCULATES KELVIN FUNCTIONS AND ITS
C  DERIVATIVES OF THE FIRST AND SECOND KIND, UTILIZING
C  IMSL ROUTINES MMKELO AND MMKEL1.
C
      IMPLICIT REAL*8(A-H,O-Z)
      CALL MMKELO (X,BERO,BEIO,XKERO,XKEIO,IERO)
      CALL MMKEL1 (X,BER1,BEI1,XKER1,XKEI1,IER1)
      R2 = 2.**0.5
C
      BER2 = -R2/X * (BER1-BEI1) -BERO
      BEI2 = -R2/X * (BEI1+BER1) -BEIO
      XKER2 = -R2/X * (XKER1-XKEI1) -XKERO
      XKEI2 = -R2/X * (XKEI1+XKER1) -XKEIO
C
      DBER2 = -(BER1+BEI1)/R2 - 2.*BER2/X
      DBEI2 = -(BEI1-BER1)/R2 - 2.*BEI2/X
      DKER2 = -(XKER1+XKEI1)/R2 - 2.*XKER2/X
      DKEI2 = -(XKEI1-XKER1)/R2 - 2.*XKEI2/X
C
      RETURN
      END
C*********************************************************************
      SUBROUTINE TTINV(A,HO)
C  THIS SUBROUTINE INVERTS THE (TT) MATRIX FOR
C  ALL TYPES OF SHELLS
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(4,4),B(4,4)
      I1 = 1
      I2 = 4
      J1 = 1
      J2 = 4
      DET = 0.
C
      DO 5 I=1,4
      DO 5 J=1,4
    5 B(I,J) = 0
      IF(HO.NE.0.) GO TO 20
      DO 15 M=1,3,2
      DO 10 N=1,3,2
      IF(A(M,N).EQ.0.)  GO TO 10
      I1 = M
      I2 = M + 1
      J1 = N
      J2 = N + 1
      I = 0
      J = 0
      K = 0
      L = 0
      GO TO 25
   10 CONTINUE
   15 CONTINUE
C
   20 DO 80 I=1,4
      DO 75 J=1,4
C
```

```
          N1 = 1
          B(I,J) = 0.
          DO 55 K=1,4
          IF(K.EQ.I) GO TO 55
          DO 50 L=1,4
          IF(L.EQ.J) GO TO 50
C
   25       N2 = 1
            C2 = 0.
            DO 50 M=I1,I2
            IF(M.EQ.K.OR.M.EQ.I) GO TO 50
            DO 45 N=J1,J2
            IF(N.EQ.L.OR.N.EQ.J) GO TO 45
C
              DO 35 MM=I1,I2
              IF(MM.EQ.M.OR.MM.EQ.K.OR.MM.EQ.I) GO TO 35
              DO 30 NN=J1,J2
              IF(NN.EQ.N.OR.NN.EQ.L.OR.NN.EQ.J) GO TO 30
              C3 = A(MM,NN)
              IF(HO.NE.0.) GO TO 40
              B(M,N) = A(MM,NN)
              B(MM,NN) = A(M,N)
              GO TO 40
   30         CONTINUE
   35         CONTINUE
C
   40       C2 = C2 + (-1.)**(N2+1)*A(M,N)*C3
            N2 = N2 + 1
            IF(HO.EQ.0..AND.N2.EQ.3)  GO TO 55
            IF(N2.EQ.3) GO TO 55
   45       CONTINUE
   50       CONTINUE
C
   55     B(I,J) = B(I,J) + (-1.)**(N1+1)*A(K,L)*C2
          N1 = N1 + 1
          IF(N1.EQ.4.AND.I.EQ.1) GO TO 70
          IF(N1.EQ.4.AND.I.GT.1)  GO TO 75
   50     CONTINUE
   55     CONTINUE
C
   70   DET = DET + (-1.)**(I+J)*A(I,J)*B(I,J)
   75   CONTINUE
   80   CONTINUE
C
        GO TO 90
   85   DET = C2
C
   90   DO 100 I=1,4
        DO 95 J=1,4
   95   A(I,J) = (-1.)**(I+J)*B(J,I)/DET
  100   CONTINUE
        RETURN
        END
C*********************************************************************
        SUBROUTINE ROWEX(A)
C   THIS SUBROUTINE PERFORMS ROW INTERCHANGES FOR
C   MATRICES (TA) AND AND (TT) WHEN THE SHELL IS
C   INVERTED.
C
        IMPLICIT REAL*8(A-H,O-Z)
        DIMENSION A(4,4)
        C = 1.
        DO 20 I=1,2
        DO 10 J=1,4
        TEMP = C*A(I,J)
        A(I,J) = C*A(I+2,J)
   10   A(I+2,J) = TEMP
        C = -1.
   20   CONTINUE
        RETURN
        END
C*********************************************************************
         SUBROUTINE PCYLIN(T,R,H,HO,E,PR,UW,ALPHA,F,PSD,IP,PV,N,PSF,PST)
C   THIS SUBROUTINE COMPUTES CYLINDER PARTICULAR SOLUTION DISPLACEMENTS (PSD)
        IMPLICIT REAL*8 (A-H,O-Z)
        DIMENSION F(6,6),PSD(4),PSF(20,6)
C
        DO 10 I=1,4
 10     PSD(I)=0.0
        IF(IP.LT.1.OR.IP.GT.7)  GO TO 999
        PSD(1)=-PST*PR*R/(E*T)
        PSD(3)=PSD(1)
C
        GO TO (20,30,20,40,70,70,50),IP
 20     PSD(1)=PV*R**2/(E*T) +PSD(1)
        PSD(3)=PSD(1)
        GO TO 70
C
 30     PSD(3)=PSD(3)-UW*T*PR*R*H/(E*T)
        C=UW*T*PR*R/(E*T)
        PSD(2)=PSD(2)-C
        PSD(4)=PSD(4)-C
        GO TO 70
C
 40     C=-ALPHA*R*PV
        PSD(1)=PSD(1)+C
        PSD(3)=PSD(3)+C
        GO TO 70
C
   50   C = PV*R**2/(E*T)
        PSD(3) = PSD(3) + C*H
        PSD(2) = PSD(2) - C
        PSD(4) = PSD(4) + C
C
 70     DO 100   I=1,4
        C=PSD(I)
        DO 80 J=1,4
 80     C=C+F(I,J)*PSF(N,J)
 100    PSD(I)=C
C
        RETURN
C
 999    WRITE(6,1000) IP
```

```
 1000 FORMAT('  PROGRAM STOPPED FOR CYLINDER IP =',I4)
      STOP
      END
C**********************************************************************
      SUBROUTINE PDOME(IT,T,R,H,HO,E,PR,UW,ALPHA,F,PSD,IP,PV,IDV,PSF,
     *  ANG,ANGO,WHT,PST,IFLAG,N)
C  THIS SUBROUTINE COMPUTES DOME PARTICULAR DISPLACEMENTS (PSD)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION F(6,6),PSD(6),PSF(20,6)
C
      IF(IP.LT.1.OR.IP.GT.7)  GO TO 999
      DO 10 I=1,6
   10 PSD(I)=0.0
      RLAM=(3.*(1.-PR**2)*(R/T)**2)**.25
C
      PHI1 = ANGO
      PHI2 = ANG
      C1 = 1.
      IF(IT.EQ.6)  GO TO 15
      PHI1 = ANG
      PHI2 = ANGO
      C1 = -1.
C
C  INITIALIZE STRESS RESULTANTS AT THE TOP AND
C  BOTTOM OF THE SPHERICAL SEGMENT.
C
   15 IF(PHI1.EQ.0.)  GO TO 20
      RN10 = FN1(PV,PST,UW,R,T,IP,IT,PHI1,ANGO,ANG,WHT,IDV)
      RN20 = FN2(PV,PST,UW,R,T,IP,IT,PHI1,ANGO,ANG,WHT,IDV)
   20 IF(PHI2.EQ.0.)  GO TO 25
      RN11 = FN1(PV,PST,UW,R,T,IP,IT,PHI2,ANGO,ANG,WHT,IDV)
      RN21 = FN2(PV,PST,UW,R,T,IP,IT,PHI2,ANGO,ANG,WHT,IDV)
C
   25 GOTO(30,30,30,40,70,30,30),IP
   30 IF(PHI1.EQ.0.)  GO TO 35
      PSD(1) = PSD(1) -R*DSIN(PHI1)*(RN20-PR*RN10)/(E*T)
      IF(IP.EQ.2) PSD(2) = PSD(2)-UW*R*((1.+PR)*(C1*DCOS(PHI1)**2
     *                     -1.)/DSIN(PHI1)-C1*DSIN(PHI1))/E
C
      IF(IP.EQ.6) PSD(2) = PSD(2)  - PV*R*((1.+PR)*(C1*DCOS(PHI1)**2
     *                              -1.)-2*C1*DSIN(PHI1)**2)/
     *                              (DTAN(PHI1)*E*T)
C
      IF(IP.EQ.7) PSD(2)=PSD(2) - ((RN10-RN20)*(1.+PR)
     *                          +PV*R*((1.+PR)*(WHT+C1*R)
     *                          +C1*R*DSIN(PHI1)
     *                          +C1*(1.+PR)*R*DCOS(PHI1)
     *                          ))/(E*T*DTAN(PHI1))
C
   35 IF(PHI2.EQ.0.)  GO TO 70
      PSD(3) = PSD(3) -R*DSIN(PHI2)*(RN21-PR*RN11)/(E*T)
      IF(IP.EQ.2) PSD(4) = PSD(4)+UW*R*((1.+PR)*(C1*DCOS(PHI2)**2
     *                     -1.)/DSIN(PHI2)-C1*DSIN(PHI2))/E
C
      IF(IP.EQ.6) PSD(4) = PSD(4)  + PV*R*DCOS(PHI2)*((1.+PR)*(C1*
     *                              DCOS(PHI2)**2-1.)-2.*C1*DSIN
     *                              (PHI2)**2)/(DSIN(PHI2)*E*T)
C
      IF(IP.EQ.7) PSD(4)=PSD(4)  + ((RN11-RN21)*(1.+PR)
     *                           -(C1*PV*R*((1.+PR)*(WHT+C1*R*DCOS
     *                           (PHI1))*(DSIN(PHI1)/DSIN(PHI2))**2
     *                           + C1*(1.+PR)*R*(DCOS(PHI2)
     *                           + 2.*(DCOS(PHI2)**3-DCOS(PHI1)**3)/
     *                           (3.*DSIN(PHI2)**2)))))
     *                           /(E*T*DTAN(PHI2))
     *                           + PV*R*R*DSIN(PHI2)/(E*T)
      GO TO 70
C
 40   PSD(3)=PSD(3)-ALPHA*PV*R*DSIN(PHI2)
      PSD(1)=PSD(1)-ALPHA*PV*R*DSIN(PHI1)
C
   70 DO 100   I=1,4
      C=PSD(I)
      DO 80 J=1,4
 80   C=C+F(I,J)*PSF(N,J)
 100  PSD(I)=C
      RETURN
C
 999  WRITE(6,1000)  IP
 1000 FORMAT('  PROGRAM STOPPED FOR DOME IP =',I4)
      RETURN
      END
C**********************************************************************
      SUBROUTINE PCONE(IT,T,R,H,HO,E,PR,UW,ALPHA,F,PSD,IP,
     *  PV,IDV,PSF,ANG,WHT,PST,IFLAG,N)
C  THIS SUBROUTINE COMPUTES CONE PARTICULAR DISPLACEMENTS (PSD)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION F(6,6),PSD(6),PSF(20,6)
C
      IF(IP.LT.1.OR.IP.GT.7)  GO TO 999
      DO 10 I=1,6
   10 PSD(I)=0.0
      Y2 = H
      Y1 = HO
      C1 = 1.
      IF(IT.NE.6)  GO TO 15
      C1 = -1.
      Y1 = H
      Y2 = HO
   15 C = .5*PR*(H**2+HO**2)/Y2**2
C
      GOTO (20,30,20,60,70,40,50),IP
   20 DER0 = PV*(PR-DTAN(ANG))
      DER1 = PV*(C -DTAN(ANG))
      GO TO 55
   30 DER0 = UW*T*(PR/DCOS(ANG)-C1*DSIN(ANG)*DTAN(ANG))
      DER1 = UW*T*(C /DCOS(ANG)-C1*DSIN(ANG)*DTAN(ANG))
      GO TO 55
   40 DER0 = PV*DTAN(ANG)*(PR-C1*DSIN(ANG)**2)
      DER1 = PV*DTAN(ANG)*(C -C1*DSIN(ANG)**2)
      GO TO 55
   50 DER0 = PV*DTAN(ANG)*(-WHT-C1*2.*Y1*DCOS(ANG)
     *          + 0.5*PR*WHT*(1.+(HO/Y1)**2)
```

```
     *          + C1*2.*Y1*DCOS(ANG)/3.
     *          + C1*PR*HD**3*DCOS(ANG)/(3.*Y1**2))
   51 DERO = PV*DTAN(ANG)*(-WHT-C1*2.*Y2*DCOS(ANG)
     *          + 0.5*PR*WHT*(1.+(HD/Y2)**2)
     *          + C1*2.*Y2*DCOS(ANG)/3.
     *          + C1*PR*HD**3*DCOS(ANG)/(3.*Y2**2))
   55 IF(Y1.EQ.0.)  GO TO 57
      RN10 = FN3(IP,IT,Y1,ANG,H,HD,WHT,PV,PST,UW,T,IDV)
      RN20 = FN4(IP,IT,Y1,ANG,H,HD,WHT,PV,PST,UW,T)
      PSD(1) = PSD(1) - Y1*DSIN(ANG)*(RN20-PR*RN10)/(E*T)
      PSD(2) = PSD(2)-DTAN(ANG)*((1.+PR)*(RN10-RN20)
     *                                -Y1*DERO)/(E*T)
   57 PSB = 0.
      RN11 = FN3(IP,IT,Y2,ANG,H,HD,WHT,PV,PSB,UW,T,IDV)
      RN21 = FN4(IP,IT,Y2,ANG,H,HD,WHT,PV,PSB,UW,T)
      PSD(3) = PSD(3) - Y2*DSIN(ANG)*(RN21-PR*RN11)/(E*T)
      PSD(4) = PSD(4)-DTAN(ANG)*((1.+PR)*(RN11-RN21)
     *                                -Y2*DER1)/(E*T)
      GO TO 70
C [illegible]
   60 PSD(3)=PSD(3)-ALPHA*PV*Y2*DSIN(ANG)
      PSD(1)=PSD(1)-ALPHA*PV*Y1*DSIN(ANG)
C
   70 DO 100   I=1,4
      C = PSD(I)
      DO 80 J=1,4
   80 C=C+F(I,J)*PSF(N,J)
  100 PSD(I)=C
      RETURN
C
  [illegible] WRITE(5,1000) IP
 1000 FORMAT(' PROGRAM STOPPED FOR CONE IP =',I4)
      RETURN
      END
C*********************************************************************
      SUBROUTINE SOL(A,B,NN,NEQ)
C   THIS SUBROUTINE SOLVES A SET OF LINEAR ALGEBRAIC EQUATIONS
C   OF THE FORM A*X=B BY GAUSSIAN ELIMINATION, WHERE 'A' IS A
C   SQUARE MATRIX. B IS THE RIGHT-HAND SIDE VECTOR
C   ON ENTRY, BUT IS OVERWRITTEN WITH THE SOLUTION VECTOR 'X'
C   DURING BACK SUBSTITUTION
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(NN,NN),B(NN)
      NL=NEQ-1
      DO 250 N=1,NL
      IF(A(N,N).LE.0.)  GO TO 500
      N1=N+1
      DO 100 J=N1,NEQ
  100 A(N,J)=A(N,J)/A(N,N)
      B(N)=B(N)/A(N,N)
      DO 250 I=N1,NEQ
      IF(A(I,N).EQ.0.)  GO TO 250
      C=A(I,N)
      DO 200 J=N1,NEQ
  200 A(I,J)=A(I,J)-C*A(N,J)
      B(I)=B(I)-C*B(N)
  250 CONTINUE
C    BACK SUBSTITUTION
      M=NEQ
      B(M)=B(M)/A(M,M)
      DO 400  N=1,NL
      M1=M
      M=M-1
      DO 400 J=M1,NEQ
  400 B(M)=B(M)-B(J)*A(M,J)
      GO TO 600
  500 WRITE (6,1000)  N
      CALL EXIT
 1000 FORMAT(' ZERO OR NEGATIVE ELEMENT ON MAIN DIAGONAL OF TRIANGULARIZ
     1ED STIFFNESS MATRIX    '/'  FOR EQUATION NUMBER ',I4)
  600 RETURN
      END
C*********************************************************************
      SUBROUTINE BASE(IFLAG,T,R,H,HO,E,PR,UW,BB,S,TT,D)
C   THIS SUBROUTINE COMPUTES THE FLEXIBILITY MATRIX (S)
C   FOR A BASE SEGMENT ON AN ELASTIC FOUNDATION, OR
C   (IF IFLAG=1) THE MATRIX BB TO DETERMINE INTERNAL
C   DISPLACEMENTS AND STRESS RESULTANTS
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION S(6,6),TT(4,4),B(4,4),G(4,4),BB(4,4)
      DIMENSION  PHI(4),PHIP(4),PHIDP(4),PHITP(4),IVEC(4)
      DATA IVEC/2,5,4,6/
C
C   FUNCTION DEFINITIONS
      FII(RO,RI)=CM*RI/RO**2+CP/RI
      FOI(RO,RI)=2.*C/RO
      FOO(RO,RI)=CM*RO/RI**2+CP/RO
      FIO(RO,RI)=2.*C/RI
C
      LM=2
      IF(HO.EQ.0 ) LM=1
      LN=2*LM
      D=E*T**3/(12.*(1.-PR**2))
      STIFL=(D/R)**0.25
      IF(IFLAG.EQ.2) GOTO 300
C
      SIGN=1.0
      RD=H
      DO 50 J=1,6
      DO 50 I=1,6
   50 S(I,J)=0.0
      DO 51 I=1,4
      DO 51 J=1,4
   51 B(I,J)=0.0
C
      DO 100 I=1,LM
      I1=2*(I-1)+1
      I2=I1+1
      IF(I.EQ.2)  RD=HO
      IF(I.EQ.2)  SIGN=-1.0
      RDI=1./RD
      RDI2=RDI**2
      CALL  BSHAPE(STIFL,RD,PHI,PHIP,PHIDP,PHITP)
```

```
      DO 80 J=1,LN
      B(I2,J)=-D*(PHITP(J)+RDI*PHIDP(J)-RDI2*PHIP(J))*SIGN
      B(I1,J)=D*(PHIDP(J)+PR*RDI*PHIP(J))*SIGN
      B(I2,J)=PHI(J)
      B(I1,J)=PHIP(J)
   80 CONTINUE
100   CONTINUE
C
      CALL JINVER(B,TT,4,LN)
      IF(IFLAG.EQ.1) GOTO 210
C
      DO 200 I=1,LN
      II=IVEC(I)
      DO 200 J=1,LN
      JJ=IVEC(J)
      C=0.0
      DO 150 K=1,LN
150   C=C+B(I,K)*TT(K,J)
200   S(II,JJ)=C
C
C  ADD FLEXIBILITIES FOR IN PLANE STIFFNESSES
      IF(LM.EQ.1) GOTO 205
      C=(H*HO)**2/(T*(H**2-HO**2)*E)
      CM=(1.-PR)*C
      CP=(1.+PR)*C
      S(1,1)=F00(H,HO)
      S(1,3)=F0I(H,HO)
      S(3,1)=FI0(H,HO)
      S(3,3)=FII(H,HO)
      GOTO 210
205   S(1,1)=H*(1.-PR)/(E*T)
C
  210 RETURN
C
300   CALL BSHAPE(STIFL,H,PHI,PHIP,PHIDP,PHITP)
      DO 320 J=1,LN
      BB(1,J)=D*(PR*PHIDP(J)+PHIP(J)/H)
      BB(2,J)=D*(PHIDP(J)+PR*PHIP(J)/H)
      BB(3,J)=-D*(PHITP(J)+PHIDP(J)/H-PHIP(J)/H**2)
      BB(4,J)=PHI(J)
      X=HO/STIFL
      IF(X.LT.2.) X=2.0
      HO=X*STIFL
  320 CONTINUE
      GOTO 210
      END
C
C
C***********************************************************************
      SUBROUTINE BSHAPE(STIFL,RD,PHI,PHIP,PHIDP,PHITP)
C  THIS SUBROUTINE EVALUATES THE PHI VECTOR AND ITS DERIVATIVES
C  FOR A BASE ON ELASTIC FOUNDATION SEGMENT
      IMPLICIT REAL*8(A-H,O-Z)
      DATA RT,PI/2.0,3.1415926536/
      DIMENSION PHI(4),PHIP(4),PHIDP(4),PHITP(4)
C
      P8=PI/8.
      CD1=1./(DSQRT(RT)*STIFL)
      SIG=RD*CD1
      RSIG=DSQRT(SIG)
      COSP=DCOS(SIG+P8)
      SINP=DSIN(SIG+P8)
      COSM=DCOS(SIG-P8)
      SINM=DSIN(SIG-P8)
      ETA=RT**0.75*DSQRT(PI)
      CPHIP=DEXP(SIG)/(ETA*RSIG)
      CPHIM=PI*DEXP(-SIG)/(ETA*RSIG)
      CD2=CD1**2
      CD3=CD2*CD1
C
C  FORM PHI VECTOR
      PHI(1)=CPHIP*COSM
      PHI(2)=CPHIP*SINM
      PHI(3)=CPHIM*COSP
      PHI(4)=CPHIM*SINP
C
C  FORM PHIP VECTOR
      SIG2I=1./(2.*SIG)
      SIGP=1.+SIG2I
      SIGM=1.-SIG2I
      PHIP(1)=CD1*(SIGM*PHI(1)-PHI(2))
      PHIP(2)=CD1*(SIGM*PHI(2)+PHI(1))
      PHIP(3)=-CD1*(SIGP*PHI(3)+PHI(4))
      PHIP(4)=CD1*(-SIGP*PHI(4)+PHI(3))
C
C  FORM PHIDP VECTOR
      C2=1./(2.*SIG**2)
      PHIDP(1)=CD2*C2*PHI(1)+(SIGM*PHIP(1)-PHIP(2))*CD1
      PHIDP(2)=CD2*C2*PHI(2)+(SIGM*PHIP(2)+PHIP(1))*CD1
      PHIDP(3)=CD2*C2*PHI(3)-(SIGP*PHIP(3)+PHIP(4))*CD1
      PHIDP(4)=CD2*C2*PHI(4)-(SIGP*PHIP(4)-PHIP(3))*CD1
C
C  FORM PHITP VECTOR
      B2=2.*C2
      C3=-1./(SIG**3)
      PHITP(1)=CD3*C3*PHI(1)+CD2*C2*PHIP(1)+(SIGM*PHIDP(1)-
     *    PHIDP(2))*CD1
      PHITP(2)=CD3*C3*PHI(2)+CD2*C2*PHIP(2)+(SIGM*PHIDP(2)+
     *    PHIDP(1))*CD1
      PHITP(3)=CD3*C3*PHI(3)+CD2*C2*PHIP(3)-(SIGP*PHIDP(3)+
     *    PHIDP(4))*CD1
      PHITP(4)=CD3*C3*PHI(4)+CD2*C2*PHIP(4)-(SIGP*PHIDP(4)-
     *    PHIDP(3))*CD1
C
      RETURN
      END
C
C***********************************************************************
      SUBROUTINE JINVER(A,B,NDIM,NEQ)
C  THIS SUBROUTINE INVERTS THE MATRIX A BY THE JACOBI METHOD
C  AND STORES THE RESULT IN B
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(NDIM,1),B(NDIM,1)
```

```
C
C  INITIALIZE THE B MATRIX
      DO 100 J=1,NDIM
      DO 100 I=1,NDIM
100   B(I,J)=0.0
      DO 110 J=1,NEQ
110   B(J,J)=1.0
C
C  BEGIN JACOBI REDUCTION OF MATRIX A AND ALSO OPERATE ON B
      DO 800 N=1,NEQ
      IF(DABS(A(N,N)).LT.1.0D-06) GO TO 999
      C=1./A(N,N)
      N1=N+1
      IF(N.EQ.NEQ) GO TO 410
      DO 400 J=N1,NEQ
      AJ=A(N,J)*C
      BJ=B(N,J)*C
      DO 300 I=1,NEQ
      A(I,J)=A(I,J)-AJ*A(I,N)
300   B(I,J)=B(I,J)-BJ*A(I,N)
      A(N,J)=AJ
400   B(N,J)=BJ
C
410   DO 500 J=1,N
      BJ=B(N,J)*C
      DO 450 I=1,NEQ
450   B(I,J)=B(I,J)-BJ*A(I,N)
500   B(N,J)=BJ
C
800   CONTINUE
      RETURN
C
999   WRITE(6,1000) N
1000  FORMAT('0',' ** ZERO ELEMENT ON MAIN DIAGONAL FOR EQUATION',
     *  I4,' INDICATES MATRIX IS SINGULAR')
      STOP
      END
C
C
C*************************************************************************
      SUBROUTINE PBASE(T,R,H,HO,E,PR,ALPHA,UW,F,PSD,IP,PV,N,PSF,PBF)
C  THIS SUBROUTINE EVALUATES THE PARTICULAR SOLUTION
C  DISPLACEMENTS FOR A BASE SEGMENT
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION  F(6,6),PSD(6),PSF(20,6),PBF(6,20)
C
      DO 10 I=1,6
10    PSD(I)=0.0
C
C  SELECT LOAD TYPE
      IF(IP.LT.1.OR.IP.GT.7)  GO TO 999
      GO TO (20,30,40,50,70,70,60),IP
C
C  INTERNAL PRESSURE
20    PSD(5)=PV/R
      PSD(6)=PSD(5)
      GO TO 70
C
C  DEAD LOAD
30    PSD(5)=UW*T/R
      PSD(6)=PSD(5)
      GO TO 70
C
C  IN-PLANE PRESTRESS
40    PSD(1)=PV*H/T
      PSD(3)=PSD(1)*HO/H
      GO TO 70
C
C  UNIFORM THERMAL
50    PSD(1)=-PV*ALPHA*H
      PSD(3)=-PV*ALPHA*HO
      GO TO 70
C
C  LIQUID PRESSURE
60    PSD(5)=PV*H/R
      PSD(6)=PSD(5)
C
70    DO 100 I=1,6
      C=PSD(I)
      DO 80 J=1,6
80    C=C+F(I,J)*PSF(N,J)
  100 PSD(I)=C
      RETURN
C
999   WRITE(6,1000)
1000  FORMAT('0',' **  PROGRAM STOPPED IN SUBROUTINE PBASE FOR DIAGNOSE'
     *  ,'D ERROR')
C
      STOP
      END
```