NASA TECHNICAL NOTE

NASA TN D-6158

c.1

NASA TN D-6158

COMPUTER PROGRAM FOR STATIC AND DYNAMIC AXISYMMETRIC NONLINEAR RESPONSE OF SYMMETRICALLY LOADED ORTHOTROPIC SHELLS OF REVOLUTION

by Wendell B. Stephens

Langley Research Center
Hampton, Va. 23365

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • DECEMBER 1970

1. Report No.
NASA TN D-6158

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
COMPUTER PROGRAM FOR STATIC AND DYNAMIC AXISYMMETRIC NONLINEAR RESPONSE OF SYMMETRICALLY LOADED ORTHOTROPIC SHELLS OF REVOLUTION

5. Report Date
December 1970

6. Performing Organization Code

7. Author(s)
Wendell B. Stephens

8. Performing Organization Report No.
L-6299

9. Performing Organization Name and Address
NASA Langley Research Center
Hampton, Va. 23365

10. Work Unit No.
124-08-15-02

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, D.C. 20546

13. Type of Report and Period Covered
Technical Note

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract
A computer program has been developed which determines the nonlinear behavior of symmetrically loaded elastic orthotropic shells of revolution. The loading can be due to either mechanical or thermal forces and can be applied statically or dynamically. The analysis is based on Sanders' equations for shells with small strains and moderately small rotations and allows for variable stiffness properties of the shell along the meridian. Spatial derivatives are approximated by finite differences, and integration with respect to time is carried out by the Houbolt method. For static behavior, or dynamic response at each point in time, a Newton-Raphson method is applied for convergence to the nonlinear solution. The boundary conditions are presented in a general form which allows either classical or elastic constraints to be used. The program, which is written in FORTRAN IV language, is described in detail and sample calculations are included.

17. Key Words (Suggested by Author(s))
Elastic orthotropic shells of revolution
Nonlinear analysis
Symmetric static or dynamic loads
FORTRAN IV computer program

18. Distribution Statement
Unclassified – Unlimited

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of Pages
98

22. Price*
$3.00

CONTENTS

COMPUTER PROGRAM FOR STATIC AND DYNAMIC AXISYMMETRIC NONLINEAR RESPONSE OF SYMMETRICALLY LOADED ORTHOTROPIC SHELLS OF REVOLUTION

By Wendell B. Stephens
Langley Research Center

SUMMARY

A computer program has been developed which determines the nonlinear behavior of symmetrically loaded elastic orthotropic shells of revolution. The loading can be due to either mechanical or thermal forces and can be applied statically or dynamically. The analysis is based on Sanders' equations for shells with small strains and moderately small rotations and allows for variable stiffness properties of the shell along the meridian. Spatial derivatives are approximated by finite differences, and integration with respect to time is carried out by the Houbolt method. For static behavior, or dynamic response at each point in time, a Newton-Raphson method is applied for convergence to the nonlinear solution. The boundary conditions are presented in a general form which allows either classical or elastic constraints to be used. The program, which is written in FORTRAN IV language, is described in detail and sample calculations are included.

INTRODUCTION

The analysis of shells of revolution subjected to static, thermal, or time-dependent loads is an important problem in the design of missiles and space vehicles. A finite-difference solution for the linear bending behavior of an isotropic shell subjected to an arbitrary static load is contained in reference 1 and is modeled after the analysis procedure found in reference 2. Geometrically nonlinear terms are included for essentially the same problem in reference 3. However, there remains a need for a program which accounts for dynamic loads and material orthotropy. Such a dynamic response analysis is useful for practical aerospace applications such as the study of launch, staging, and water-impact loadings of aeroshells. In addition, such an analysis would provide a means of determining the nonlinear prebuckling stress distributions required for accurate stability analyses. In this report a computer program is described which has been developed to determine the axisymmetric nonlinear static and dynamic response including axisymmetric static and dynamic buckling of an arbitrary elastic orthotropic shell of revolution subjected to axisymmetric loads. The analysis, programing techniques, and the computer program documentation are presented as well as representative sample problems.

The analysis is based on Sanders' nonlinear equations (ref. 4) with material orthotropy added as in reference 5. The governing partial differential equations are written in terms of first-order spatial derivatives and solved numerically by using central differences for derivatives along the meridian and backward differences for time derivatives. Integration with respect to time is started by using the Houbolt technique (refs. 6 and 7). For the boundary conditions, either classical or elastic constraints may be used. The nonlinear difference equations are solved for each time step or static load increment by the Newton-Raphson method (ref. 8). "Top-of-the-knee" static buckling is determined from the lack of convergence of the Newton-Raphson procedure.

The program is divided into nine subroutines and seven user-supplied function subprograms. A maximum of 101 equal stations is provided requiring an octal storage of 70 000 memory words. The program is written in CDC version of FORTRAN IV language for operation in the CDC 6600/6400 digital computer at the Langley Research Center. The output consists of a problem description together with displacements, rotations, and moment and force resultants in tabular form.

In order to present both the analysis and the computer program, appendixes are frequently used to simplify the text. Appendixes A, B, C, D, and E are used to clarify the presentation of the analysis, and appendixes F and G contain the program listing and a sample of the program output, respectively.

SYMBOLS

The units for physical quantities defined in this paper are given both in the U.S. Customary Units and in the International System of Units (SI). Factors relating the two systems are given in appendix H.

a reference (or characteristic) length

C_{11},C_{12},C_{22} nondimensional orthotropic extensional material constants defined in equations (14); for example, $C_{11} = \frac{\overline{C}_{11}}{E_o H_o}$

$\overline{C}_{11},\overline{C}_{12},\overline{C}_{22}$ orthotropic extensional material constants

D_{11},D_{12},D_{22} nondimensional orthotropic bending material constants defined in equations (14); for example, $D_{11} = \frac{\lambda^2 \overline{D}_{11}}{E_o H_o^3}$

$\overline{D}_{11},\overline{D}_{12},\overline{D}_{22}$ orthotropic bending material constants

E_o reference modulus of elasticity

E_1,E_2 moduli of elasticity in principal directions, meridional and circumferential, respectively

E_{10},E_{20} nondimensional moduli of elasticity in principal directions; for example, $E_{10} = \frac{E_1}{E_o}$

$E_o/\rho g$ specific stiffness where ρg is weight density

e_{11},e_{22} nondimensional extensional strain, meridional and circumferential directions, respectively; for example, $e_{11} = \epsilon_{11}\eta$

g acceleration due to gravity

H shell thickness

H_o reference thickness

$\overline{H}$ maximum shell rise of spherical cap considered in sample problem

h nondimensional shell thickness, H/H_o

j temperature exponent in equation (17)

K_{11},K_{12},K_{22} nondimensional orthotropic material constants associated with coupling between extension and bending and defined in equations (14); for example, $K_{11} = \frac{\lambda\overline{K}_{11}}{E_oH_o^2}$

$\overline{K}_{11},\overline{K}_{12},\overline{K}_{22}$ orthotropic material constants associated with coupling between extension and bending

k_{11},k_{22} principal change in curvatures, meridional and circumferential directions, respectively

M_{11}, M_{22} bending-moment resultants in principal directions, meridional and circumferential, respectively

m_{11}, m_{22} nondimensional bending-moment resultants in principal directions; fòr example, $m_{11} = \frac{aM_{11}}{\sigma H_o^3}$

N_{11}, N_{22} membrane stress resultants, meridional and circumferential directions, respectively

n number of stations along meridian

n_{11}, n_{22} nondimensional membrane stress resultants, meridional and circumferential, respectively; for example, $n_{11} = \frac{N_{11}}{\sigma H_o}$

P, P_s lateral and meridional forces per unit area, respectively

P_{cr} critical symmetric buckling load

$P^* = \frac{p}{p_{cl}}$

p, p_s nondimensional lateral and meridional forces per unit area, respectively; for example, $p = \frac{Pa}{\sigma H_o}$

p_{cl} nondimensional classical buckling pressure of complete spherical shell (see eq. (30))

Q transverse shear resultant

q nondimensional transverse shear resultant, $Q/\sigma H_o$

R radial distance from axis of symmetry to shell reference surface

R_1, R_2 principal radii of curvature, meridional and circumferential directions, respectively

r	nondimensional radial distance from axis of symmetry to shell reference surface, R/a
r_1, r_2	nondimensional principal radii of curvature; for example, $r_1 = \frac{R_1}{a}$
S	distance measured along shell meridian
s	nondimensional distance measured along meridian, S/a
Δs	nondimensional meridional difference increment
T, T_1, T_2	temperature quantities defined with equation (17)
t	real time
Δt	real time increment
t_1^m, t_2^m	nondimensional thermal moment resultant in principal directions, defined in equations (16)
t_1^n, t_2^n	nondimensional thermal force resultant in principal directions, defined in equations (16)
U, W	meridional and normal displacement, respectively
u, w	nondimensional meridional and normal displacement, respectively; for example, $u = \frac{\eta U}{a}$
x	force vector with elements n_{11}, q, and m_{11}
y	displacement vector with elements u, w, and β
z	vector composed of x and y vectors
α_1, α_2	coefficients of linear thermal expansion in principal directions, meridional and circumferential, respectively
$\overline{\alpha}, \overline{\beta}, \overline{\gamma}, \overline{\delta}$	coefficients of the acceleration difference equation (21)

β	nondimensional rotation, $\eta\tilde{\beta}$
$\tilde{\beta}$	meridional rotation
Δ	average deflection of spherical cap in sample problem
δ_{pg}	Kronecker delta
$\epsilon_{11},\epsilon_{22}$	membrane strains
ζ	coordinate normal to reference surface of shell, positive outward, with origin at reference surface and nondimensionalized by H_o
η	ratio of reference elasticity modulus to reference stress, E_o/σ
θ	circumferential coordinate
κ_{11},κ_{22}	nondimensional principal curvatures; for example, $\kappa_{11} = a\eta k_{11}$
λ	ratio of reference thickness to characteristic length, H_o/a
λ_s	shell parameter defined by equation (31)
ν_{12},ν_{21}	Poisson's ratios for meridional and circumferential directions, respectively
ρ	mass density
σ	reference stress
τ	nondimensional time, $\sqrt{\frac{E_o}{\rho a^2}}\, t$
$\Delta\tau$	nondimensional time increment
ϕ	colatitude angle, angle between shell axis and normal to shell middle surface

Subscripts:

i	spatial station number, that is, 1, 2, . . ., n

j	matrix number, that is, 1, 2, . . ., 2n
k	kth equation of set of equations at a point
m	time step, that is, 1, 2, . . .
max	maximum

Matrices:

z,e	6×1
$\left.\begin{array}{l}A,B,C,D,E,\\F_1,F_3,P,Q\end{array}\right\}$	3×3
$\left.\begin{array}{l}XOLD,X,\\R,x,y,q,l\end{array}\right\}$	3×1
$\hat{H},\tilde{H},M$	6×6

A prime indicates a derivative with respect to the nondimensional meridional distance s.

A dot indicates a derivative with respect to nondimensional time τ.

ANALYTICAL FORMULATION

The shell analysis procedure is summarized in this section. Also included are the geometric description, derivation of the nonlinear equilibrium conditions, compatibility equations, and differencing scheme as well as the Newton-Raphson procedure for solution of the governing equations.

Shell Geometry

The shell geometry and coordinate system for the reference surface of a general shell of revolution are shown in figure 1. The geometry of the shell reference surface is defined by ϕ and R. Any point in the shell may be located by specifying the orthogonal coordinates s, θ, and ζ where $s = \frac{S}{a}$ and is the nondimensional meridional coordinate, S is the meridional shell coordinate, a is the reference length of the shell, θ is the circumferential coordinate, and ζ is a coordinate normal to and originating at the shell reference surface, positive outward. The nondimensional principal radii of curvature, r_1 and r_2, are (ref. 1)

$$\left.\begin{aligned} \frac{1}{r_1} &= \phi' \\ \frac{1}{r_2} &= \frac{\sin\phi}{r} \end{aligned}\right\} \tag{1}$$

where the prime indicates a derivative with respect to the nondimensional meridional distance s. The radii are nondimensionalized by use of the reference length a.

Equilibrium Conditions

By utilizing the results in reference 4 and the nondimensional variables described in reference 2, the nondimensional equilibrium equations become

$$n_{11}' + \frac{\cos\phi}{r} n_{11} + \phi' q - \frac{\cos\phi}{r} n_{22} + \frac{1}{\eta} \phi' \beta n_{11} - h\ddot{u} = -p_s \tag{2}$$

$$q' - \phi' n_{11} + \frac{\cos\phi}{r} q - \frac{\sin\phi}{r} n_{22} + \frac{1}{\eta}\left[\left(\beta n_{11}\right)' + \frac{r'}{r} \beta n_{11}\right] - h\ddot{w} = -p \tag{3}$$

$$m_{11}' + \frac{\cos\phi}{r} m_{11} - \frac{\cos\phi}{r} m_{22} - \frac{q}{\lambda^2} = 0 \tag{4}$$

where n_{11} and n_{22} are the stress resultants, m_{11} and m_{22} are the bending-moment resultants, u and w are the displacements, p_s and p are the surface loads, q is the shear resultant, and β is the meridional rotation. These quantities are defined in figure 2. The term λ is a nondimensional constant representing the ratio of the reference thickness H_o to the reference length a. The dots indicate derivatives with respect to nondimensional time τ.

Rotational and Strain-Displacement Relationships

The nondimensional rotation, strain-displacement relationships, and curvatures from reference 4 are

$$\beta = w' - \phi' u \tag{5}$$

$$e_{11} = u' + \phi' w + \frac{1}{2\eta} \beta^2 \tag{6}$$

$$e_{22} = \frac{\cos \phi}{r} u + \frac{\sin \phi}{r} w \tag{7}$$

$$\kappa_{11} = -\beta' \tag{8}$$

$$\kappa_{22} = -\frac{\cos \phi}{r} \beta \tag{9}$$

The terms e_{11} and e_{22} are the nondimensional principal strains and κ_{11} and κ_{22} are the nondimensional principal curvatures.

Constitutive Equations

For symmetrically loaded orthotropic shells of revolution nondimensional elasticity relationships obtained from reference 5 can be written as

$$n_{11} = C_{11}e_{11} + C_{12}e_{22} + K_{11}\kappa_{11} + K_{12}\kappa_{22} - t_1^n \tag{10}$$

$$n_{22} = C_{12}e_{11} + C_{22}e_{22} + K_{12}\kappa_{11} + K_{22}\kappa_{22} - t_2^n \tag{11}$$

$$m_{11} = \lambda^{-2}K_{11}e_{11} + \lambda^{-2}K_{12}e_{22} + \lambda^{-2}D_{11}\kappa_{11} + \lambda^{-2}D_{12}\kappa_{22} - t_1^m \tag{12}$$

$$m_{22} = \lambda^{-2}K_{12}e_{11} + \lambda^{-2}K_{22}e_{22} + \lambda^{-2}D_{12}\kappa_{11} + \lambda^{-2}D_{22}\kappa_{22} - t_2^m \tag{13}$$

Since only axisymmetric behavior is considered, only these four relationships are required. The nondimensional stiffnesses are given by

$$\left.\begin{aligned}
C_{11} &= \frac{E_{10}}{1 - \nu_{12}\nu_{21}} \int_{\zeta_1}^{\zeta_2} d\zeta \\
C_{12} &= \frac{\nu_{12}E_{10}}{1 - \nu_{12}\nu_{21}} \int_{\zeta_1}^{\zeta_2} d\zeta \\
C_{22} &= \frac{E_{20}}{1 - \nu_{12}\nu_{21}} \int_{\zeta_1}^{\zeta_2} d\zeta \\
K_{11} &= \frac{\lambda E_{10}}{1 - \nu_{12}\nu_{21}} \int_{\zeta_1}^{\zeta_2} \zeta \, d\zeta
\end{aligned}\right\} \tag{14}$$

(Equations continued on next page)

$$\left.\begin{aligned}
K_{12} &= \frac{\lambda \nu_{12} E_{10}}{1 - \nu_{12}\nu_{21}} \int_{\zeta_1}^{\zeta_2} \zeta \, d\zeta \\
K_{22} &= \frac{\lambda E_{20}}{1 - \nu_{12}\nu_{21}} \int_{\zeta_1}^{\zeta_2} \zeta \, d\zeta \\
D_{11} &= \frac{\lambda^2 E_{10}}{1 - \nu_{12}\nu_{21}} \int_{\zeta_1}^{\zeta_2} \zeta^2 \, d\zeta \\
D_{12} &= \frac{\lambda^2 \nu_{12} E_{10}}{1 - \nu_{12}\nu_{21}} \int_{\zeta_1}^{\zeta_2} \zeta^2 \, d\zeta \\
D_{22} &= \frac{\lambda^2 E_{20}}{1 - \nu_{12}\nu_{21}} \int_{\zeta_1}^{\zeta_2} \zeta^2 \, d\zeta
\end{aligned}\right\} \tag{14}$$

where ζ is positive outward and ζ_1 and ζ_2 are the distances to the inner and outer shell surfaces, respectively, from the reference surface.

Because of the symmetry of the orthotropic constants, use has been made of the relationship

$$E_{10}\nu_{12} = E_{20}\nu_{21} \tag{15}$$

The nondimensional thermal forces and moments in the meridional and circumferential directions, respectively, due to a temperature $T(s,\zeta)$ are (ref. 1)

$$\left.\begin{aligned}
t_1^n &= \frac{E_{10}\eta}{1 - \nu_{12}\nu_{21}}\left(\alpha_1 + \nu_{12}\alpha_2\right)\int_{\zeta_1}^{\zeta_2} T \, d\zeta \\
t_2^n &= \frac{E_{20}\eta}{1 - \nu_{12}\nu_{21}}\left(\alpha_2 + \nu_{21}\alpha_1\right)\int_{\zeta_1}^{\zeta_2} T \, d\zeta \\
t_1^m &= \frac{E_{10}\eta}{1 - \nu_{12}\nu_{21}}\left(\alpha_1 + \nu_{12}\alpha_2\right)\int_{\zeta_1}^{\zeta_2} T\zeta \, d\zeta \\
t_2^m &= \frac{E_{20}\eta}{1 - \nu_{12}\nu_{21}}\left(\alpha_2 + \nu_{21}\alpha_1\right)\int_{\zeta_1}^{\zeta_2} T\zeta \, d\zeta
\end{aligned}\right\} \tag{16}$$

where α_1 and α_2 are the orthrotropic coefficients of thermal expansion in the principal directions.

Temperature Profile

The temperature is allowed to vary through the thickness of the shell and along the meridian as follows

$$T(s) = T_1(s) + T_2(s)\zeta^j \tag{17}$$

where the T_1 defines the temperature change from a standard temperature at the reference surface and T_2 is the difference between the temperatures of the shell outer and inner surfaces at ζ_2 and ζ_1. The exponent j is used to define the temperature thickness profile as a constant ($j = 0$) or as a linear variation through the thickness ($j = 1$) or as a nonlinear variation through the thickness ($j \geqq 2$).

Finite-Difference Formulation of Governing Equations

It is shown in appendix A that equations (2) to (13) can be written as six partial differential equations. These equations are first order in spatial derivatives and second order in time derivatives. The set of equations in matrix form is

$$Iz' + \left(\hat{H} + \tilde{H}\right)z = e + M\ddot{z} \tag{18}$$

where

$$z = \begin{Bmatrix} n_{11} \\ q \\ m_{11} \\ u \\ w \\ \beta \end{Bmatrix}$$

Here z is the solution vector of six variables, I is the 6×6 identity matrix, $\hat{H}$ and $\tilde{H}$ are the linear and nonlinear 6×6 coefficient matrices of z, respectively, M is the 6×6 mass matrix of z, and e is the six-element load vector. The elements of $\hat{H}$, $\tilde{H}$, e, and M, are listed in appendix A.

The governing equations are converted into difference equations by utilizing central differences for the spatial derivatives and backward differences (refs. 6 and 7) for the time derivatives. As shown in reference 7, this backward-difference scheme is numerically stable. The spatial finite-difference representations are written at a point halfway between stations as shown in figure 3 and are of the form

$$z_{i-1/2} = \frac{z_i + z_{i-1}}{z} \tag{19}$$

$$z'_{i-1/2} = \frac{z_i - z_{i-1}}{\Delta s} \tag{20}$$

The second-order time derivative in equations (18) is approximated at the ith station by

$$\ddot{z}_{i,m} = \frac{1}{(\Delta\tau)^2}\left(\bar{\alpha}_m z_{i,m} + \bar{\beta}_m z_{i,m-1} + \bar{\gamma}_m z_{i,m-2} + \bar{\delta}_m z_{i,m-3}\right) \tag{21}$$

where $i = 1, 2, \ldots, n$ and $m = 1, 2, \ldots$. In equations (19), (20), and (21) the subscripts i and m indicate spatial and time stations, respectively, and Δs and $\Delta\tau$ are the spatial and time increments, respectively. The coefficients $\bar{\alpha}_m$, $\bar{\beta}_m$, $\bar{\gamma}_m$, and $\bar{\delta}_m$ depend on the time step and initial conditions and are given in appendix B. Application of these finite-difference approximations (eqs. (19) to (21)) to the governing equations (18) leads to the following set of nonlinear algebraic equations at the mth time step:

$$F_{i-1/2} z_{i-1,m} + G_{i-1/2} z_{i,m} = L_{i-1/2} \tag{22}$$

where

$$\left.\begin{aligned}
F_{i-1/2} &= \frac{1}{2}\left(\hat{H}_{i-1/2} + \tilde{H}_{i-1/2} - \frac{M_{i-1/2}\bar{\alpha}_m}{(\Delta\tau)^2}\right) - \frac{I}{\Delta s} \\
G_{i-1/2} &= \frac{1}{2}\left(\hat{H}_{i-1/2} + \tilde{H}_{i-1/2} - \frac{M_{i-1/2}\bar{\alpha}_m}{(\Delta\tau)^2}\right) + \frac{I}{\Delta s} \\
L_{i-1/2} &= e_{i-1/2} + \frac{M_{i-1/2}}{(\Delta\tau)^2}\left(\bar{\beta}_m z_{i-1/2,m-1} + \bar{\gamma}_m z_{i-1/2,m-2}\right. \\
&\quad \left. + \bar{\delta}_m z_{i-1/2,m-3}\right)
\end{aligned}\right\} \tag{23}$$

and $i = 2, 3, \ldots, n$ and $m = 1, 2, 3, \ldots$. These equations with three boundary conditions at each edge of the shell define the problem to be solved and must be solved simultaneously to determine z at the mth time step.

Boundary Conditions

As shown in reference 4 the classical shell boundary conditions at either edge, $s = 0$ and $s = \frac{S_{max}}{a}$, are defined by either force resultants n_{11}, q, and m_{11} or displacements u, w, and β, so that

$$\Omega x_i + \Lambda y_i = l \tag{24}$$

Here the subscript i is either 1 or n and the 3×3 Ω and Λ matrices and the 3×1 l vector are required to define the boundary conditions. The vectors x_i and y_i are

$$x_i = \begin{Bmatrix} n_{11} \\ q \\ m_{11} \end{Bmatrix}_i \qquad y_i = \begin{Bmatrix} u \\ w \\ \beta \end{Bmatrix}_i \tag{25}$$

These vectors define the force and displacement subvectors of z, respectively. Typical boundary conditions including general elastic constraints are discussed in appendix C.

Computational Procedure

The nonlinear set of equations (22) and (24) are linearized by use of an iterative Newton-Raphson procedure (ref. 8). This is done by placing the $L_{i-1/2}$ term and the l term on the left-hand side of equations (22) and (24), respectively, and writing the kth equation at the ith station as

$$f_k\left(z_i, z_{i-1}, s\right) = 0 \tag{26}$$

where k is 1, 2, . . ., 6 for equation (22) and $k = 1, 2, 3$ for equations (24). Use of the first two terms of a Taylor's expansion for equation (26) together with an approximate solution vector $\overline{z}$ gives

$$f_k\left(z_i, z_{i-1}, s\right) = f_k\left(\overline{z}_i, \overline{z}_{i-1}, s\right) + \left.\frac{\partial f_k}{\partial z_i}\right|_{z_i=\overline{z}_i} \delta z_i + \left.\frac{\partial f_k}{\partial z_{i-1}}\right|_{z_{i-1}=\overline{z}_{i-1}} \delta z_{i-1} = 0 \tag{27}$$

where $i = 1, 2, . . ., n$ and where δz_i is the correction vector which must be added to the approximate solution vector $\overline{z}_i$ so that equation (26) is satisfied.

The iterative procedure consists of adding the correction vector to the approximate solution vector to obtain an improved approximate solution. Thus

$$\bar{z}_i^{j+1} = \bar{z}_i^j + \delta z_i \tag{28}$$

where the superscript j indicates the jth iteration cycle. When δz_i becomes sufficiently small, convergence has been obtained.

For convenience in the solution of the simultaneous equations, the correction vector δz_i is partitioned into the two three-element ordered subvectors δx_i and δy_i. Thus the set of governing equations (27) including the appropriate boundary conditions take the following form of a five-diagonal-banded matrix where each element is a 3×3 matrix.

$$\begin{bmatrix}
C_1 & D_1 & E_1 & & & & & & & \\
B_2 & C_2 & D_2 & E_2 & & & & & & \\
A_3 & B_3 & C_3 & D_3 & E_3 & & & & & \\
 & & \ddots & & & & & & & \\
 & & A_{2(i-1)} & B_{2(i-1)} & C_{2(i-1)} & D_{2(i-1)} & E_{2(i-1)} & & & \\
 & & & A_{2i-1} & B_{2i-1} & C_{2i-1} & D_{2i-1} & E_{2i-1} & & \\
 & & & & & & \ddots & & & \\
 & & & & & A_{2n-1} & B_{2n-1} & C_{2n-1} & D_{2n-1} \\
 & & & & & & A_{2n} & B_{2n} & C_{2n}
\end{bmatrix}
\begin{Bmatrix} x_1 \\ y_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_i \\ y_i \\ \cdot \\ \cdot \\ \cdot \\ x_n \\ y_n \end{Bmatrix}
=
\begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ \cdot \\ \cdot \\ \cdot \\ q_{2(i-1)} \\ q_{2i-1} \\ \cdot \\ \cdot \\ \cdot \\ q_{2n-1} \\ q_{2n} \end{Bmatrix}
\tag{29}$$

For brevity, δx_i and δy_i are written as x_i and y_i in equation (29). The first and last rows are the boundary conditions at $s = 0$ and $s = \frac{S_{max}}{a}$, respectively, and are obtained from equations (24). The six first-order governing equations from equations (18) and (27) correspond to the pair of rows at $2(i-1)$ and $2i-1$, respectively. Here n is equal to the number of spatial stations. The definitions of the A, B, C, D, E, and q matrices in terms of equations (2) to (13), (22), and (27) are given in appendix D.

The set of equations (29) is solved by using a modified Potters method (refs. 2 or 9) for banded matrices. A presentation of the recurrence equations required for the Potters method is contained in appendix E. For each time step, the elements in the 3×3 matrices A, B, C, D, and E and the three-element vector q are functions of the shell properties, and the new displacement and stress state for the last three time steps.

The vectors $z_{i,0}$ and $\dot{z}_{i,0}$ at the initial time $\tau = 0$ must be given. Both the incorporation of the initial conditions into the problem and the definition of the $\bar{\alpha}_m$, $\bar{\beta}_m$, $\bar{\gamma}_m$, and $\bar{\delta}_m$ coefficients for $\ddot{z}_m$ are contained in reference 7 and appendix B. Appendix B also includes nonhomogeneous initial conditions.

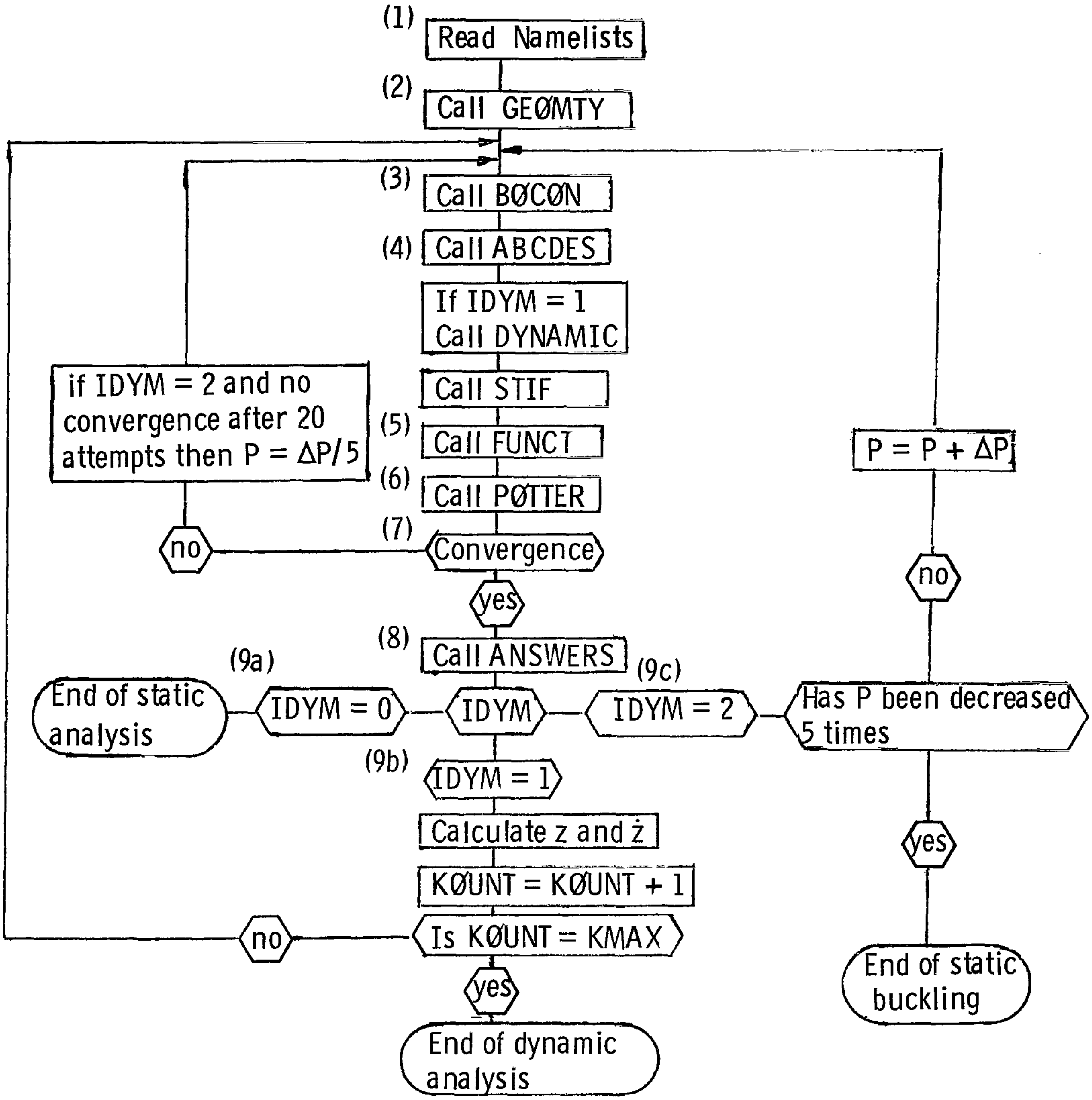
This analysis and numerical solution has been programed in FORTRAN IV and the resulting computer program (SADAØS) includes the input provisions of general shell shape, thermal and mechanical loads, structural orthotropy, and arbitrary boundary conditions at each end of the shell.

COMPUTER PROGRAM

This section contains the description of the computer program SADAØS and is intended to be a user's document. A listing of the program is contained in appendix F and a sample printout in appendix G. In writing this program, various options on types of analyses, geometry, and boundary conditions have been included to eliminate the necessity of having the user develop subroutines. However, should these options be inadequate, the program is subdivided into separate subroutines so that further options can be exercised without a detailed knowledge of the program. Certain function subprograms are required to be programed by the user. These function subprograms define the loading, shell thickness, temperature integrals, and initial conditions. In addition, input data and computer subroutine preparation are explained in detail in later sections.

Program Organization

The flow chart is presented in the following block diagram. As an aid in reading the block diagram, a list of subroutines and their description is presented in table 1. In table 2 the variables and constants are listed with their program names.



Block diagram of SADAØS

A detailed description of the computing in each block of the block diagram is as follows:

BLOCK (1): Namelist GIVEN containing basic input data is read in. If requested, the optional information on boundary conditions will be read in through namelists ELBØL and ELBØR.

BLOCK (2): Shell geometry is defined at the i-1/2 increment midpoints along the shell meridian. The values defined by GEØMTY are r, ϕ, and ϕ'.

BLOCK (3): Boundary conditions at each end are set. Matrices C_1, D_1, E_1, q_1, A_{2n}, B_{2n}, C_{2n}, and q_{2n} in equation (29) are defined.

BLOCK (4): Matrices A, B, C, D, E, and q from equation (29) are calculated. These matrices are further defined in appendix D. These matrices are calculated for each $i = 2, 3, \ldots, n$ and, in turn, call the subroutines DYNAMIC (if IDYM = 1) and STIF. STIF sets the C, D, and K values from equations (14). DYNAMIC sets $\overline{\alpha}$, $\overline{\beta}$, $\overline{\gamma}$, $\overline{\delta}$ of equations (21).

BLOCK (5): From equation (27) the term $f_k\left(\overline{z}_i,\overline{z}_{i-1},s\right)$ is calculated and placed on the right-hand side of that equation. This corresponds to the q vector in equation (29).

BLOCK (6): The Potters method or Gaussian elimination scheme described in appendix E is used to solve equation (29) for the vector δz where $\delta z_i = \begin{Bmatrix} \delta x_i \\ \delta y_i \end{Bmatrix}$. The improved approximate vector is given by equation (28).

BLOCK (7): If the norm of δz is very small compared with the norm of the improved vector $\overline{z}$, then the problem has converged. If δz is not sufficiently small, a return is made to block (3), and blocks (3) to (7) are repeated until convergence is obtained. If a static buckling problem (IDYM = 2) is being attempted, and after 20 iterations there is still no convergence, a return is made to the last converged load solution and a smaller load increment step is attempted.

BLOCK (8): After convergence there is a tabular printout of the following variables at each station i: n_{11}, q, m_{11}, u, w, β, n_{22}, m_{22}, p, and p_s.

BLOCK (9a): If the analysis is a static stress analysis (IDYM = 0), the program terminates.

(9b): If the analysis is a dynamic response (IDYM = 1) problem, the $\dot{z}$ and $\ddot{z}$ vectors are calculated and an increment in time and time step is taken. This procedure is continued until KMAX steps are taken.

(9c): If static buckling load (IDYM = 2) is desired, the load is increased. If from block (7) the load increment has not been decreased five times, the problem returns to block (3). If five load increment reductions have taken place, the problem terminates.

Input Data

The following quantities in namelist GIVEN must be defined: N, NTYPE, CHAR, HØ, SØ, RØ, PHIØ, PP, PPS, EØ, E1, E2, NU12, NU21, SIGØ, NØNLIN, CØNV, IDYM, KMAX, DTAU, ALFA1, ALFA2, T1, T2, ITEMP, LBCL, LBCR, SL, SR, IFREQ, and ISTART. The format for the input data contained in a namelist is given in reference 10. The first column of the data cards cannot be used. The definitions of these quantities are as follows:

Name	Type	Interpretation
N	integer	n, number of stations along the meridian
NTYPE	integer	sets type of shell geometry to be analyzed: NTYPE = 1 denotes a cylindrical shell. RØ is the radius from the shell axis to the shell reference surface. PHIØ, the colatitude angle, is 90^{o}. SØ is the shell length. NTYPE = 2 denotes a conical shell. RØ is the radial distance to the first station at S = 0 from the shell axis. SØ is the length along the shell meridian. PHIØ is the colatitude angle (i.e., 90^{o} – Semivertex angle). NTYPE = 3 denotes a spherical cap. RØ is the shell radius. PHIØ is the colatitude angle at $S = S_{max}$. SØ is calculated internally and is read in as zero. NTYPE = 4 denotes that the user will read in a special geometry by adding statements to GEØMTY as required to define r, ϕ, and ϕ' at each i-1/2 station. Input constants RØ, PHIØ, and SØ can be used as desired by the programer. The statements are placed after the card labeled 50 and before the card labeled 60.
CHAR	real	a, reference shell dimension to be selected by user and used internally for nondimensionalizing the geometry and output quantities
HØ	real	H_o, reference thickness

Name	Type	Interpretation
SØ	real	input quantity defined by NTYPE
RØ	real	input quantity defined by NTYPE
PHIØ	real	input quantity defined by NTYPE and read in degrees
PP	real	constant used to define p, the normal pressure, in FUNCTION PL(I)
PPS	real	constant used to define p_s, the meridional pressure, in FUNCTION PS(I)
EØ	real	E_o, reference elasticity modulus
E1	real	E_1, elasticity modulus in meridional direction
E2	real	E_2, elasticity modulus in circumferential direction
NU12	real	ν_{12}, Poisson's ratio in the meridional direction
NU21	real	ν_{21}, Poisson's ratio in the circumferential direction
SIGØ	real	σ, reference stress level; normally SIGØ = 1.
NØNLIN	integer	If a linear solution is desired, set NØNLIN = 0. If nonlinear terms are to be included, set NØNLIN = 1.
CØNV	real	convergence criteria. Compares the error norm with the norm of the approximate solution vector $\left(\text{i.e., } \frac{\|\delta z\|}{\|\bar{z}\|}\right)$, to insure convergence to proper order of magnitude. Usually CØNV = 1. × 10^{-3}.
IDYM	integer	If static stress analysis is desired, set IDYM = 0. If dynamic response analysis is desired, set IDYM = 1. If static buckling analysis is desired, set IDYM = 2.
KMAX	integer	number of time steps desired when IDYM = 1. Maximum number of static-load solutions when IDYM = 2. (Provides an upper limit on iterations when snap-through buckling (IDYM = 2) does not occur.)
DTAU	real	$\Delta\tau$, size of the nondimensional time increment. $\Delta\tau = \sqrt{\frac{E_o}{\rho a^2}}\,\Delta t$.
ALFA1	real	α_1, coefficient of thermal expansion in the meridional direction
ALFA2	real	α_2, coefficient of thermal expansion in the circumferential direction

Name	Type	Interpretation
T1	real	constant used to define T_1 in equation (17)
T2	real	constant used to define T_2 in equation (17)
ITEMP	integer	j, integer exponent used in equation (17)
LBCL	integer	sets boundary condition at the $i = 1$ $(S = 0)$ edge (see eqs. (C2) to (C5)): LBCL = 1 is a pole point LBCL = 2 is a pinned edge LBCL = 3 is a fixed edge LBCL = 4 is a free edge LBCL = 5 elastic constants in namelist ELBØL must be given
LBCR	integer	sets boundary conditions at the $i = n$ $(S = S_{max})$ edge (see eqs. (C2) to (C5)): LBCR = 1 is a pole point LBCR = 2 is a pinned edge LBCR = 3 is a fixed edge LBCR = 4 is a free edge LBCR = 5 elastic constants in namelist ELBØR must be given
SL	real	three-element array equated to values defined by LBCL (see appendix C), equivalent to l_1 in equations (24) and (C1)
SR	real	three-element array equated to values defined by LBCR (see appendix C), equivalent to l_n in equations (24) and (C1)
IFREQ	integer	frequency of printout at time steps of dynamic-load problems (IDYM = 1) or at load steps in the static buckling problem (IDYM = 2)
ISTART	integer	Normally ISTART = 0. If ISTART = 1, user must supply non-homogeneous initial values to FUNCTION DV for deflections and velocities u, w, $\dot{u}$, and $\dot{w}$.

The first input quantity in namelist GIVEN is preceded by $GIVEN and the last input quantity is followed by $. For example, the first input card could be

```
$GIVEN N = 21, NTYPE = 3, HØ = 1.,
```

and the last namelist card could be

```
SL(1) = 0., 0., 0., SR(1) = 0., 0., 0., IFREQ = 4, ISTART = 0$
```

Finally, one input card may contain a description of the problem. All 80 columns may be used. If no description is desired, a blank card must be included after the last namelist GIVEN data card.

If LBCL = 5, then namelist ELBØL must be included in the input. Two 3 × 3 matrices Ω_L and Λ_L are read in columnwise. Their elements specify the elastic constraints at the first boundary of the meridian (i = 1). For example, a simply supported edge of a shell free to displace in a horizontal plane and with an applied edge moment yields the boundary conditions

$$n_{11} \cos \phi_o + q \sin \phi_o = 0$$

$$u \sin \phi_o - w \cos \phi_o = 0$$

$$m_{11} = m_o$$

where ϕ_o is the colatitude angle at the boundary. Thus, the Ω, Λ, and SL matrices become

$$\Omega_L = \begin{bmatrix} \cos \phi_o & \sin \phi_o & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \Lambda_L = \begin{bmatrix} 0 & 0 & 0 \\ \sin \phi_o & -\cos \phi_o & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad SL = \begin{Bmatrix} 0 \\ 0 \\ m_o \end{Bmatrix}$$

If LBCR = 5 the namelist ELBØR must be included in the input. Two 3 × 3 matrices Ω_R and Λ_R are read in columnwise. Their elements specify the elastic constraint at the last boundary of the meridian (i = n).

User-Prepared Function Subprograms

In addition to the input data, the user must prepare certain function subprograms. These function subprograms must be written and included in the program by the user to calculate the quantity at each half-station, i-1/2. For example, at i = 2 the FUNCTION PL(I) will define the lateral nondimensional pressure at a point halfway between i = 1 and i = 2. The i-1/2 station is shown in figure 3. The following table describes the function subprograms.

FUNCTION	Quantity	Interpretation
PL(I)	$p_{i-1/2}$ (i = 2, 3, . . ., n)	computes the nondimensional lateral pressure

FUNCTION	Quantity	Interpretation
PS(I)	$p_{s,i-1/2}$ (i = 2, 3, . . ., n)	computes the nondimensional meridional pressure
TDZ(I)	$\int_{\zeta_1}^{\zeta_2} T_{i-1/2}\, d\zeta$ (i = 2, 3, . . ., n)	computes the nondimensional thermal force integral
TZDZ	$\int_{\zeta_1}^{\zeta_2} T_{i-1/2}\zeta\, d\zeta$ (i = 2, 3, . . ., n)	computes the nondimensional thermal moment integral
T(I)	$h_{i-1/2}$ (i = 2, 3, . . ., n)	computes the nondimensional shell thickness
IP(KØUNT)	required when IDYM = 1	locates time stations where a load is suddenly applied or removed
DV(M,I)	required when ISTART = 1	prescribes the nonhomogeneous initial conditions: M = 1 denotes the u displacement at station i M = 2 denotes the w displacement at station i M = 3 denotes the $\dot{u}$ velocity at station i M = 4 denotes the $\dot{w}$ velocity at station i

Program Output

The output is divided into two parts. The first part is a printout of the input data and shell geometry. The second part is the printout of n_{11}, q, m_{11}, u, w, β, p, p_s, n_{22}, and m_{22} at all stations for the converged solution. If the problem varies with time, then the second part is repeated KMAX times. At a time station where there is a sudden change in load (i.e., step loads denoted by IP = 1 in FUNCTION IP(KØUNT)) there is an additional printout of the vectors u, w, $\dot{u}$, and $\dot{w}$.

Program Limitations

The program is limited to 101 spatial stations and 70 000 octal storage locations. At present there is no programed mechanism for allowing the orthotropic coefficients of thermal expansion α_1 and α_2 to vary through the thickness. This could be accomplished by the user by writing FUNCTION TDZ and FUNCTION TZDZ to include variable thermal coefficients $\alpha_1(\zeta)$ and $\alpha_2(\zeta)$.

In subroutine STIF these constants are set up for a general orthotropic shell with the reference surface at the middle surface. The stiffnesses, that is, the C_{ij}, D_{ij}, and K_{ij} defined in equations (14), are general. Thus, the user could, with minimal knowledge of the computer program and equations (14), alter subroutine STIF to include a shell stiffened by rings or stringers smeared over appropriate increments (ref. 11) and using any reference surface.

Analytical Limitations

For static buckling problems (IDYM = 2) the buckling is limited to "top-of-the-knee" axisymmetric buckling loads. A detailed discussion of top-of-the-knee buckling is contained in reference 12.

Errors in results will normally be one of two types: (1) inconsistency of input data or (2) numerical error inherent in the analysis. The first type of error can be eliminated by careful scrutiny and checking of the input data and user-prepared function statements. The second type of error can only be minimized by taking the increments in time and space small enough to guarantee that a sufficient number of stations exist for an accurate solution. A comparison test of the results for various increment sizes is an adequate means of determining appropriate increment sizes.

SAMPLE PROBLEMS

Spherical Cap With Dynamic Loading

The first problem to be solved is one considered in references 12 and 13. An isotropic shallow spherical cap with clamped edges is subjected to a step pulse compressive pressure applied at $\tau = 0$ and removed at $\tau = 5$. The compressive nondimensional lateral pressure is taken as 60 percent of the classical buckling pressure p_{cl} applied to a complete spherical shell where

$$p_{cl} = \frac{2\eta\lambda\left(\frac{h}{r_o}\right)^2}{\left[3\left(1 - \nu^2\right)\right]^{1/2}} \tag{30}$$

and $r_o = \frac{R}{a}$ (fig. 4). The remaining shell properties are

$R = 100$ in. (2.54 m)

$H = 1$ in. (0.0254 m)

$E_1 = E_2 = 10 \times 10^6$ psi $(68.95\ \mathrm{GN/m^2})$

$\phi = 15.8094^\circ$

$\nu_{12} = \nu_{21} = 0.3$

These parameters correspond to a shallow-shell parameter of $\lambda_s = 5$ where

$$\lambda_s = 2\left[3\left(1 - \nu^2\right)\right]^{1/4}\left(\frac{\overline{H}}{H}\right)^{1/2} \tag{31}$$

and $\overline{H}$ is the maximum shell rise. The reference length CHAR or a is set at 100 in. (2.54 m) and E_o is taken as 10×10^6 psi $(68.95\ \mathrm{GN/m^2})$ with H_o and σ set at unity. In addition, spatial increments are set at 1/25 of the meridian and the time increment (DTAU) is set at 0.25. The number of spatial stations and size of the time increment for this problem were established by comparing increasingly small spacings until stable solutions were obtained. The time response is desired out to $\tau = 10$ and a printout is requested at every fourth time increment. A complete listing of the program along with the results for this sample problem is contained in appendixes F and G.

The namelist GIVEN quantities become

N = 26
NTYPE = 3
CHAR = 100.
HØ = 1.
SØ = 0.
RØ = 100.
PHIØ = 15.8094
PP = -0.6
PPS = 0.
EØ = 10. × 10^6
E1 = 10. × 10^6
E2 = 10. × 10^6
NU12 = 0.3
NU21 = 0.3
SIGØ = 1.0
NØNLIN = 1
CØNV = 0.001
IDYM = 1
KMAX = 40

DTAU = 0.25
ALFA1 = 0.
ALFA2 = 0.
T1 = 0.
T2 = 0.
ITEMP = 0
LBCL = 1
LBCR = 3
SL = 0., 0., 0.
SR = 0., 0., 0.
IFREQ = 4
ISTART = 0

The descriptive card comment is

SAMPLE PROBLEM FOR A CLAMPED SPHERICAL CAP WITH LAMBDA = 5.

The namelists ELBØL and ELBØR are not needed since neither LBCL or LBCR is set equal to five.

The FUNCTION IP sets the time steps m at which there are step load changes. The input data (DTAU) was selected to have an incremental size, $\Delta\tau = 0.25$. Therefore, the abrupt or sudden load changes occur at stations $m = 0$ and $m = 20$ for $\tau = 0$ and $\tau = 5$, respectively. In the program the subscript m is represented by KØUNT. If there is no sudden change in load, IP is set equal to zero. If there is sudden change in load, IP is set equal to one at that time station. The user-supplied statements for FUNCTION IP(KØUNT) become

```
IP = 0
IF (KØUNT .EQ. 0) IP = 1
IF (KØUNT .EQ. 20) IP = 1
```

The user-supplied information to FUNCTION T(I) is

```
T = 1.0/HØ
```

where T defines h at a station i-1/2.

FUNCTION PL(I) and FUNCTION PS(I) set the lateral and meridional loads. For the sample problem a compressive uniform lateral load is used. Thus, in FUNCTION PS(I) the user-supplied statement is

```
PS = 0.
```

Common statements provide the values for ETA (η), LAM (λ), NU12 $\left(\nu_{12}\right)$, NU21 $\left(\nu_{21}\right)$, ROA (r_o), PP, and KØUNT to be set in the function statement. Therefore, the user-supplied statements for FUNCTION PL(I) become

```
   PCL = 2.*LAM*ETA/(3.*(1 - NU12*NU21))**.5*(T(I)/ROA)**2*E10
   PLL = PP*PCL
   PL = PLL
   IF (KØUNT .EQ. 0) PL = 0.
   IF (KØUNT .GT. 20) PL = 0.
   IF (IACC .NE. 1) GO TO 1
   IF (KØUNT .EQ. 0) PL = PLL
   IF (KØUNT .EQ. 20) PL = 0.
 1 CØNTINUE
```

Here IACC is computed internally at time points where the load changes abruptly (i.e., a step load) leaving the load doubly defined at that point in time. The first definition of a doubly defined load point at the mth time station (represented by KØUNT) is placed before the IACC statement card and the second definition of PL at that time-step point m is placed after the IACC statement. In this sample problem the abrupt load changes occur at KØUNT equal to 0 and 20. Therefore, at KØUNT equals 0 the statement before the IACC statement is

```
IF (KØUNT .EQ. 0) PL = 0.
```

and after the IACC statement is

```
IF (KØUNT .EQ. 0) PL = PLL
```

At time station $m = 20$ the statement before the IACC statement is

```
IF (KØUNT .GT. 20) PL = 0.
```

and after the IACC statement is

```
IF (KØUNT .EQ. 20) PL = 0.
```

At $m = 20$ the load is being suddenly removed so that initially the loading is equal to PLL and finally is equal to zero. The negative sign in the namelist GIVEN quantity PP makes the loading compressive.

Since ISTART = 0, the FUNCTION DV(M,I) will not be called. It is interesting to note that since the initial conditions are zero, ISTART could be either one or zero. Since $z_{i,0} = \dot{z}_{i,0} = 0$, the input to FUNCTION DV for ISTART = 1 would have been DV = 0.

FUNCTION TDZ and FUNCTION TZDZ are completely contained as defined by equations (16) and input constants for use in defining α_1, α_2, T_1, T_2, and j are provided for in the namelist GIVEN through ALFA1, ALFA2, T1, T2, and ITEMP, respectively.

The statements appearing in appendix F for these subroutines assume that the reference surface is located at the mid surface.

A listing of the program with this sample problem is contained in appendix F. A special nondimensional value Δ, the average inward deflection, is computed and printed for ease of comparison with references 12 and 13. The value Δ will be printed whenever NTYPE = 3. The output to KØUNT = 4 is contained in appendix G.

The results for 60 percent of classical buckling load (this sample problem) and other percentages are summarized in figure 5. The agreement with the results in reference 12 is quite good. The discrepancies are attributed to differences in problem formulation and time increment sizes. The agreement with reference 13 is also good for $P^* = 0.4$ but poor for $P^* = 0.6$ and $\tau > 2$. The discrepancy between the present results and those of reference 13 for $P^* = 0.6$ is attributed to the use of a five-degree-of-freedom analysis in that study as compared with 26 finite-difference stations in the present study. Reference 13 reports a dynamic buckling load of $P_{cr} = 0.52$. In reference 12 the P_{cr} is 0.65 which agrees closely with the present result of 0.68 as shown in figure 6.

An extensive study utilizing the program for both static and dynamic buckling has been made in reference 14. Also contained in reference 14 is a thorough discussion of both static and dynamic buckling criteria.

Thermally Loaded Clamped Cylinder

This sample problem demonstrates the use of the program for analyzing thermal loads. The problem chosen is that one contained in reference 15 where an isotropic cylinder clamped at the first station and clamped and on rollers in the longitudinal direction at the final edge undergoes a linear temperature rise of 350° F (194.4 K) from one end of the shell to the other. The analysis is linear and the shell dimensions are

$S_{max} = 48$ in. (1.2192 m)
$R = 12$ in. (0.3048 m)
$H = 2$ in. (0.0508 m)
$\nu_{12} = \nu_{21} = 0.3$
$E_1 = E_2 = 28 \times 10^6$ psi (193 GN/m^2)
$\alpha_1 = \alpha_2 = 9.5 \times 10^{-6}$ in./in./$^{\circ}$F (17.1×10^{-6} m/m/K)

Thus, namelist GIVEN becomes

N = 21
NTYPE = 1
CHAR = 1.
HØ = 1.

SØ = 48.
RØ = 12.
PHIØ = 90.
PP = 0.
PPS = 0.
EØ = 1.
E1 = $28. \times 10^6$
E2 = $28. \times 10^6$
NU12 = 0.3
NU21 = 0.3
SIGØ = 1.0
NØNLIN = 0
CØNV = .001
IDYM = 0
KMAX = 0
DTAU = 0.
ALFA1 = 9.5×10^{-6}
ALFA2 = 9.5×10^{-6}
T1 = 350.
T2 = 0.
ITEMP = 0
LBCL = 3
LBCR = 5
SL = 0., 0., 0.
SR = 0., 0., 0.
IFREQ = 1
ISTART = 0

The FUNCTION T(I) requires the statement

T = 2./HØ

The FUNCTION TDZ(I) requires the statement

TDZ = T(I)*T1*((FLØAT (I) - 1.5)/FLØAT (N-1))**2

Since equation (17) is now simply $T = T_1$ then $T_1 = 350\left(\frac{S}{48}\right)^2$

The description card comment becomes

THERMAL PROBLEM OF MENDELSON PG 186.

Since LBCR = 5 then namelist ELBØR becomes

OMEGAR (1,1) = 1.0, 8*0.

ALAMDAR (1,1) = 4*0., 1., 3*0., 1.

where the elements are read in columnwise. These input data correspond to the boundary conditions at i = n.

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} n_{11} \\ q \\ m_{11} \end{Bmatrix}_n + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} u \\ w \\ \beta \end{Bmatrix}_n = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix}$$

The right-hand side of this equation is supplied by the vector SR in the namelist GIVEN.

Functions PL, PS, IP, DV, and TZDZ require only the following cards, respectively:

PL = 0.
PS = 0.
IP = 0.
DV = 0.
TZDZ = 0.

The results of this analysis are compared with those of reference 15 in figure 7. The comparison is good; the differences are attributed to the fact that in reference 15 the isotropic shell is approximated by a six-layered shell and the u deformations are neglected.

CONCLUDING REMARKS

A computer program has been developed to analyze thin shells of revolution which are both elastically and thermally orthotropic and are subjected to either mechanical or thermal loads. These loads can be applied either statically or dynamically. The program has many options concerning geometry, boundary conditions, and loading built into the subroutines. In addition, the basic subroutines of the program allow stiffness and geometry changes to be made easily without a detailed knowledge of the entire program. The present report describes the numerical analysis procedure and serves as the user's manual for the resulting computer program.

A sample problem of the dynamic response of a spherical cap is included. The sample problem demonstrates the input data preparation for the program as well as the accuracy of the results obtained. A second example of a cylinder loaded thermally is included to show the input data required for that problem. Here again the agreement with existing results is good.

Langley Research Center,
National Aeronautics and Space Administration,
Hampton, Va., September 18, 1970.

APPENDIX A

REDUCTION TO SIX GOVERNING EQUATIONS

The six governing equations comprising equations (18) are derived in this appendix. Equations (2) to (13) are reduced to six equations for the six unknowns n_{11}, q, m_{11}, u, w, and β. Equations (10) and (12) are rewritten to define e_{11} and κ_{11} as

$$e_{11} = \frac{D_{11}}{\lambda^2 G} n_{11} - \frac{K_{11}}{G} m_{11} - N_{12} e_{22} - \lambda^2 N_3 \kappa_{22} + \frac{D_{11}}{\lambda^2 G} t_1^n - \frac{K_{11}}{G} t_1^m \tag{A1}$$

$$\kappa_{11} = -\frac{K_{11}}{\lambda^2 G} n_{11} + \frac{C_{11}}{G} m_{11} - \frac{M_{12} e_{22}}{\lambda^2} - M_3 \kappa_{22} + \frac{C_{11}}{G} t_1^m - \frac{K_{11}}{\lambda^2 G} t_1^n \tag{A2}$$

When these expressions for e_{11} and κ_{11} are substituted into equations (11) and (13) the following equations result:

$$n_{22} = N_{12} n_{11} + M_{12} m_{11} + \left(E_{12} + C_{22}\right) e_{22} + \left(\overline{K}_{21} + K_{22}\right) \kappa_{22} + N_{12} t_1^n + M_{12} t_1^m - t_2^n \tag{A3}$$

$$m_{22} = N_3 n_{11} + M_3 m_{11} + \left(E_3 + \frac{K_{22}}{\lambda^2}\right) e_{22} + \left(K_3 + \frac{D_{22}}{\lambda^2}\right) \kappa_{22} + N_3 t_1^n + M_3 t_1^m - t_2^m \tag{A4}$$

where the coefficients are

$$\left.\begin{aligned}
N_{12} &= \frac{1}{\lambda^2 G}\left(C_{12} D_{11} - K_{11} K_{12}\right) \\
M_{12} &= \frac{1}{G}\left(C_{11} K_{12} - C_{12} K_{11}\right) \\
E_{12} &= \frac{1}{\lambda^2 G}\left(2C_{12} K_{11} K_{12} - C_{12}^2 D_{11} - C_{11} K_{12}^2\right) \\
\overline{K}_{21} &= \frac{1}{\lambda^2 G}\left(C_{12} D_{12} K_{11} - C_{12} D_{11} K_{12} + K_{11} K_{12}^2 - C_{11} D_{12} K_{12}\right) \\
N_3 &= \frac{1}{\lambda^4 G}\left(D_{11} K_{12} - D_{12} K_{11}\right) \\
M_3 &= \frac{1}{\lambda^2 G}\left(C_{11} D_{12} - K_{11} K_{12}\right)
\end{aligned}\right\} \tag{A5}$$

(Equations continued on next page)

$$\left.\begin{aligned}
E_3 &= \frac{\overline{K}_{21}}{\lambda^2} \\
K_3 &= \frac{1}{\lambda^4 G}\left(2D_{12}K_{11}K_{12} - D_{11}K_{12}^2 - C_{11}D_{12}^2\right) \\
G &= \frac{1}{\lambda^2}\left(C_{11}D_{11} - K_{11}^2\right)
\end{aligned}\right\} \tag{A5}$$

The quantities e_{22} and κ_{22} are eliminated from equations (A1) to (A4) by using equations (7) and (9). Then substituting equations (A3) and (A4) into the equilibrium equations (2), (3), and (4) yields the first three equations in equations (18). Substitution of equations (A1) and (A2) into equations (6) and (8) yields the fourth and sixth equations of equations (18). Finally, equation (5) can be utilized as the definition of β for the fifth equation of equations (18). Thus, the elements of the $\hat{H}$, $\tilde{H}$, and M matrices and of the e vector in equations (18) are defined as follows. The elements of the $\hat{H}$ matrix are

$$\left.\begin{aligned}
h_{11} &= \frac{\cos\phi}{r}\left(1 - N_{12}\right) & h_{12} &= \phi' \\
h_{13} &= -\frac{\cos\phi}{r} M_{12} & h_{14} &= -\frac{\cos^2\phi}{r^2}\left(E_{12} + C_{22}\right) \\
h_{15} &= -\frac{\cos\phi\,\sin\phi}{r^2}\left(E_{12} + C_{22}\right) & h_{16} &= \frac{\cos^2\phi}{r^2}\left(\overline{K}_{21} + K_{22}\right) \\
h_{21} &= -\left(\phi' + \frac{\sin\phi}{r} N_{12}\right) & h_{22} &= \frac{\cos\phi}{r} \\
h_{23} &= -\frac{\sin\phi}{r} M_{12} & h_{24} &= -\frac{\cos\phi\,\sin\phi}{r^2}\left(E_{12} + C_{22}\right) \\
h_{25} &= -\frac{\sin^2\phi}{r^2}\left(E_{12} + C_{22}\right) & h_{26} &= \frac{\cos\phi\,\sin\phi}{r^2}\left(\overline{K}_{21} + K_{22}\right) \\
h_{31} &= -\frac{\cos\phi}{r} N_3 & h_{32} &= -\lambda^{-2} \\
h_{33} &= \frac{\cos\phi}{r}\left(1 - M_3\right) & h_{34} &= -\frac{\cos^2\phi}{r^2}\left(E_3 + \lambda^{-2}K_{22}\right) \\
h_{35} &= -\frac{\cos\phi\,\sin\phi}{r^2}\left(E_3 + \lambda^{-2}K_{22}\right) & h_{36} &= \frac{\cos^2\phi}{r^2}\left(K_3 + \lambda^{-2}D_{22}\right)
\end{aligned}\right\} \tag{A6}$$

(Equations continued on next page)

$$\left.\begin{aligned}
&h_{41} = -\frac{D_{11}}{\lambda^2 G} && h_{42} = 0 \\
&h_{43} = \frac{K_{11}}{G} && h_{44} = \frac{\cos\phi}{r} N_{12} \\
&h_{45} = \phi' + N_{12}\frac{\sin\phi}{r} && h_{46} = -\lambda^2 \frac{\cos\phi}{r} N_3 \\
&h_{51} = h_{52} = h_{53} = h_{55} = 0 && h_{54} = -\phi' \\
&h_{56} = -1 && h_{61} = -\frac{K_{11}}{\lambda^2 G} \\
&h_{62} = 0 && h_{63} = \frac{C_{11}}{G} \\
&h_{64} = -\frac{\cos\phi}{r\lambda^2} M_{12} && h_{65} = -\frac{\sin\phi}{r\lambda^2} M_{12} \\
&h_{66} = \frac{\cos\phi}{r} M_3 &&
\end{aligned}\right\} \tag{A6}$$

The elements of the $\widetilde{H}$ matrix are all zero except

$$\left.\begin{aligned}
&\tilde{h}_{11} = \frac{\phi'}{\eta}\beta \\
&\tilde{h}_{21} = \frac{\cos\phi}{r\eta}\beta + \frac{\beta'}{\eta} \\
&\tilde{h}_{26} = \frac{n'_{11}}{\eta} \\
&\tilde{h}_{46} = \frac{\beta}{2\eta}
\end{aligned}\right\} \tag{A7}$$

The elements of the e vector are

$$\left.\begin{aligned}
&e_1 = \frac{\cos\phi}{r}\left(N_{12}t_1^n + M_{12}t_1^m - t_2^n\right) - p_s \\
&e_2 = \frac{\sin\phi}{r}\left(N_{12}t_1^n + M_{12}t_1^m - t_2^n\right) - p \\
&e_3 = \frac{\cos\phi}{r}\left(N_3 t_1^n + M_3 t_1^m - t_2^m\right)
\end{aligned}\right\} \tag{A8}$$

(Equations continued on next page)

$$\left.\begin{aligned} e_4 &= \frac{D_{11}}{\lambda^2 G} t_1^n - \frac{K_{11}}{G} t_1^m \\ e_5 &= 0 \\ e_6 &= \frac{t_1^n}{\lambda^2 G} K_{11} - \frac{t_1^m}{G} C_{11} \end{aligned}\right\} \tag{A8}$$

The only nonzero elements of the mass matrix M are

$$M_{14} = M_{25} = 1 \tag{A9}$$

APPENDIX B

TIME DERIVATIVES AND INITIAL CONDITIONS

It is seen from equations (18) and (A9) that the time derivative terms arise only in the displacement vector y_i. Thus, as shown in reference 7, the $\ddot{y}_{i,m}$ terms can be written as

$$\ddot{y}_{i,m} = \frac{1}{(\Delta\tau)^2}\left(\bar{\alpha}_m y_{i,m} + \bar{\beta}_m y_{i,m-1} + \bar{\gamma}_m y_{i,m-2} + \bar{\delta}_m y_{i,m-3}\right) \tag{B1}$$

where m indicates a time step. The constants $\bar{\alpha}_m$, $\bar{\beta}_m$, $\bar{\gamma}_m$, and $\bar{\delta}_m$ are defined in reference 7 by use of Houbolt's initial starting procedure for homogeneous initial conditions of $y_{i,0} = \dot{y}_{i,0} = 0$. The procedure is presented here for general nonhomogeneous initial conditions where $y_{i,0}$ and $\dot{y}_{i,0}$ are given. From equations (2) and (3), $\ddot{y}_{i,0}$ can be calculated. Also the following difference equations at $m = 0$ can be used to define $\dot{y}_{i,0}$ and $\ddot{y}_{i,0}$ as

$$\dot{y}_{i,0} = \frac{1}{6(\Delta\tau)}\left(2y_{i,1} + 3y_{i,0} - 6y_{i,-1} + y_{i,-2}\right) \tag{B2}$$

$$\ddot{y}_{i,0} = \frac{1}{(\Delta\tau)^2}\left(y_{i,1} - 2y_{i,0} + y_{i,-1}\right) \tag{B3}$$

These two equations can be rewritten to define the fictional time points $y_{i,-1}$ and $y_{i,-2}$. Then use can be made of the general backward-difference equation

$$\ddot{y}_{i,m} = \frac{1}{(\Delta\tau)^2}\left(2y_{i,m} - 5y_{i,m-1} + 4y_{i,m-2} - y_{i,m-3}\right) \tag{B4}$$

Therefore, by using equations (B2) and (B3) to eliminate the fictional points from equation (B4) at $m = 1$ and $m = 2$, values of $\bar{\alpha}_m$, $\bar{\beta}_m$, $\bar{\gamma}_m$, and $\bar{\delta}_m$ of equation (B1) are obtained as follows:

At $m = 0$

$$\bar{\alpha}_0 = \bar{\beta}_0 = \bar{\gamma}_0 = \bar{\delta}_0 = 0$$

At $m = 1$, equation (B1) becomes

$$\ddot{y}_{i,1} = \frac{1}{(\Delta\tau)^2}\left(6y_{i,1} - 6y_{i,0}\right) + \frac{6}{\Delta\tau}\dot{y}_{i,0} - 2\ddot{y}_{i,0} \tag{B5}$$

and

$$\bar{\alpha}_1 = 6 \qquad \bar{\beta}_1 = -6 \qquad \bar{\gamma}_1 = \bar{\delta}_1 = 0$$

At $m = 2$, equation (B1) becomes

$$\ddot{y}_{i,2} = \frac{1}{(\Delta\tau)^2}\left(2y_{i,2} - 4y_{i,1} + 2y_{i,0}\right) - \ddot{y}_{i,0} \tag{B6}$$

and

$$\bar{\alpha}_2 = 2 \qquad \bar{\beta}_2 = -4 \qquad \bar{\gamma}_2 = 2 \qquad \bar{\delta}_2 = 0$$

At $m \geqq 3$, equations (B1) and (B4) are identical and

$$\bar{\alpha}_m = 2 \qquad \bar{\beta}_m = -5 \qquad \bar{\gamma}_m = 4 \qquad \bar{\delta}_m = -1$$

Equations (B4) to (B6) completely specify all time derivatives and initial conditions. Similar results for nonhomogeneous initial conditions were obtained in reference 16.

This procedure of using initial conditions is employed at every time point where there is a sudden change in load.

APPENDIX C

DEFINITION OF BOUNDARY CONDITIONS

The system of equations defined in equations (18) and appendix A requires three boundary conditions at each edge. These conditions are derived in reference 4 and are defined by a combination of the following variables:

$$\begin{Bmatrix} n_{11} \\ q \\ m_{11} \end{Bmatrix} \quad \text{or} \quad \begin{Bmatrix} u \\ w \\ \beta \end{Bmatrix}$$

Thus, z is a six-element vector and the boundary matrix contains only three equations at each end. Therefore z_i is divided into two subvectors x_i and y_i where

$$x_i = \begin{Bmatrix} n_{11} \\ q \\ m_{11} \end{Bmatrix}_i \qquad y_i = \begin{Bmatrix} u \\ w \\ \beta \end{Bmatrix}_i$$

as shown in figure 3 the left boundary is at station $i = 1$ and the right boundary is at $i = n$. Therefore

$$\left. \begin{aligned} \Omega_1 \delta x_1 + \Lambda_1 \delta y_1 &= l_1 \\ \Omega_n \delta x_n + \Lambda_n \delta y_n &= l_n \end{aligned} \right\} \tag{C1}$$

where the subscripts 1 and n refer to the first $(s = 0)$ and last $\left(s = \frac{S_{max}}{a}\right)$ stations, respectively. In the program the vector l at $i = 1$ is SL and at $i = n$ is SR. Both SL and SR are three-element arrays. The following conditions can be applied to either boundary:

For a pole point where $u = q = \beta = 0$, the matrices Ω, Λ, and l become

$$\Omega = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \Lambda = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad l = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \tag{C2}$$

APPENDIX C

For a pinned boundary where $u = w = m_{11} = 0$, the matrices Ω, Λ, and l become

$$\Omega = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \Lambda = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad l = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix} \tag{C3}$$

For a clamped boundary where $u = w = \beta = 0$, the matrices Ω, Λ, and l become

$$\Omega = \text{Null matrix} \qquad \Lambda = I \qquad l = \text{Null vector} \tag{C4}$$

For a free edge where $n_{11} = q = m_{11} = 0$, the matrices become

$$\Omega = I \qquad \Lambda = \text{Null matrix} \qquad l = \text{Null vector} \tag{C5}$$

Finally, for a boundary with general elastic constraints, Ω and Λ must be defined by the particular problem and read in through namelists ELBØL and ELBØR. The vector l is always read in through namelist GIVEN. For these boundary conditions, the elastic boundary conditions on the left (i = 1) edge are read in through namelist ELBØL and on the right (i = n) edge through ELBØR. In other words, all nine elements of both Ω and Λ must be specified.

APPENDIX D

DEFINITIONS OF A, B, C, D, E, AND q MATRICES

Equations (22), (26), (27), and (29) are related in the following manner:

$$f_k = F_{i-1/2}\begin{Bmatrix} x_{i-1} \\ y_{i-1} \end{Bmatrix} + G_{i-1/2}\begin{Bmatrix} x_i \\ y_i \end{Bmatrix} - L_{i-1/2} \tag{D1}$$

where $k = 1, 2, \ldots, 6$ and $i = 2, 3, \ldots, n$. For $k = 1, 2, 3$

$$\left.\begin{aligned}
&A_{2(i-1)} = \text{Null matrix} \\
&B_{2(i-1)} = \frac{\partial f_k}{\partial x_{i-1}} = \left(F_{i-1/2} + \delta_{2,j+1}\frac{\beta_{i-1/2}}{\Delta s\eta}\right)_{kj} && (j = 1, 2, 3) \\
&C_{2(i-1)} = \frac{\partial f_k}{\partial y_{i-1}} = \left\{F_{i-1/2} + \delta_{1,j-5}\frac{\left(n_{11}\phi\right)_{i-1/2}}{2\eta} + \delta_{2,j-4}\left[\frac{n_{11}}{\eta}\left(\frac{\cos\phi}{2r} - \frac{1}{\Delta s}\right)\right]_{i-1/2}\right\}_{kj} && (j = 4, 5, 6) \\
&D_{2(i-1)} = \frac{\partial f_k}{\partial x_i} = \left(G_{i-1/2} + \delta_{2,j+1}\frac{\beta_{i-1/2}}{\Delta s\eta}\right)_{kj} && (j = 1, 2, 3) \\
&E_{2(i-1)} = \frac{\partial f_k}{\partial y_i} = \left\{G_{i-1/2} + \delta_{1,j-5}\frac{\left(n_{11}\phi\right)_{i-1/2}}{2\eta} + \delta_{2,j-4}\left[\frac{n_{11}}{\eta}\left(\frac{\cos\phi}{2r} + \frac{1}{\Delta s}\right)\right]_{i-1/2}\right\}_{kj} && (j = 4, 5, 6) \\
&q_{2(i-1)} = -f_k\left(\bar{x}_{i-1}, \bar{y}_{i-1}, \bar{x}_i, \bar{y}_i, s\right)
\end{aligned}\right\} \tag{D2}$$

where the barred vectors indicate the approximate solutions. For $k = 4, 5, 6$

$$\left.\begin{aligned}
&A_{2i-1} = \frac{\partial f_k}{\partial x_{i-1}} = \left(F_{i-1/2}\right)_{kj} && (j = 1, 2, 3) \\
&B_{2i-1} = \frac{\partial f_k}{\partial y_{i-1}} = \left(F_{i-1/2} - \delta_{4,j-2}\frac{\beta_{i-1/2}}{4\eta}\right)_{kj} && (j = 4, 5, 6)
\end{aligned}\right\} \tag{D3}$$

(Equations continued on next page)

$$\left.\begin{aligned}
C_{2i-1} &= \frac{\partial f_k}{\partial x_i} = \left(G_{i-1/2}\right)_{kj} && (j = 1, 2, 3) \\
D_{2i-1} &= \frac{\partial f_k}{\partial y_i} = \left(G_{i-1/2} - \delta_{4,j-2}\,\frac{\beta_{i-1/2}}{4\eta}\right)_{kj} && (j = 4, 5, 6) \\
E_{2i-1} &= \text{Null matrix} \\
q_{2i-1} &= -f_k\left(x_{i-1},y_{i-1},x_i,y_i,s\right)
\end{aligned}\right\} \tag{D3}$$

Equations (C1) are related to equations (24) and (29) in the following manner. For $i = 1$

$$\left.\begin{aligned}
C_1 &= \Omega_L \\
D_1 &= \Lambda_L \\
E_1 &= \text{Null matrix} \\
q_1 &= l_L
\end{aligned}\right\} \tag{D4}$$

For $i = n$

$$\left.\begin{aligned}
A_{2n} &= \text{Null matrix} \\
B_{2n} &= \Omega_R \\
C_{2n} &= \Lambda_R \\
q_{2n} &= l_R
\end{aligned}\right\} \tag{D5}$$

APPENDIX D

The complete set of equations becomes

$$
\begin{bmatrix}
C_1 & D_1 & E_1 & F_1 & & & \\
B_2 & C_2 & D_2 & E_2 & & & \\
A_3 & B_3 & C_3 & D_3 & E_3 & F_3 & \\
 & & \ddots & & & & \\
 & A_{2(i-1)} & B_{2(i-1)} & C_{2(i-1)} & D_{2(i-1)} & E_{2(i-1)} & \\
 & & A_{2i-1} & B_{2i-1} & C_{2i-1} & D_{2i-1} & E_{2i-1} \\
 & & & & \ddots & & \\
 & & & A_{2n-1} & B_{2n-1} & C_{2n-1} & D_{2n-1} \\
 & & & & A_{2n} & B_{2n} & C_{2n}
\end{bmatrix}
\begin{Bmatrix}
x_1 \\ y_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_i \\ y_i \\ \cdot \\ \cdot \\ \cdot \\ x_n \\ y_n
\end{Bmatrix}
=
\begin{Bmatrix}
q_1 \\ q_2 \\ q_3 \\ \cdot \\ \cdot \\ \cdot \\ q_{2(i-1)} \\ q_{2i-1} \\ \cdot \\ \cdot \\ \cdot \\ q_{2n-1} \\ q_{2n}
\end{Bmatrix}
\tag{D6}
$$

where F_1 and F_3 are null matrices added for a later programing convenience as discussed in appendix E. Since they are null matrices at this point, they do not affect any of the previous definitions in equation (29).

APPENDIX E

RECURRENCE EQUATIONS

If for convenience δx_i and δy_i are represented by x_i and y_i, a recurrence solution to equation (D6) can be obtained based on the Potters method (ref. 9). To insure nonsingularity, elements c_{11}, c_{22}, and c_{33} of matrix C_1 must not be zero. If either c_{11} or c_{33} is zero in C_1, then row one or three of C_1, D_1, E_1, F_1, and q_1 must be interchanged with row one or three, respectively, of B_2, C_2, D_2, E_2, and q_2. If c_{22} is zero, then several row manipulations must take place. First, row two of C_1, D_1, E_1, F_1, and q_1 is placed in row two of B_2, C_2, D_2, E_2, and q_2; row two of B_2, C_2, D_2, E_2, and q_2 is placed in row three of A_4, B_4, C_4, D_4, E_4, and q_4; row three of A_4, B_4, C_4, D_4, E_4, and q_4 is placed in row three of A_3, B_3, C_3, D_3, E_3, and F_3; and row three of A_3, B_3, C_3, D_3, E_3, and q_3 is placed in row two of B_1, C_1, D_1, E_1, F_1, and q_1 to complete the cycle. Simpler substitutions could be made for specific shells and boundary conditions but the preceding row interchanges yield nonsingularity for all shells. Elements can exist in the F_1 and F_3 matrices of equation (D6) as a result of the row interchanging. Since C_1 is nonsingular then

$$x_1 = P_1 y_1 + Q_1 x_2 - C_1^{-1} F_1 y_2 + R_1 \tag{E1}$$

where

$$\left.\begin{aligned} P_1 &= -C_1^{-1} D_1 \\ Q_1 &= -C_1^{-1} E_1 \\ R_1 &= C_1^{-1} q_1 \end{aligned}\right\} \tag{E2}$$

and, in general

$$\left.\begin{aligned} x_i &= P_{2i-1} y_i + Q_{2i-1} x_{i+1} + R_{2i-1} \\ y_i &= P_{2i} x_{i+1} + Q_{2i} y_{i+1} + R_{2i} \end{aligned}\right\} \tag{E3}$$

where

$$\left.\begin{aligned}
G_r &= \left[\left(A_rP_{r-2} + B_r\right)P_{r-1} + A_rQ_{r-2} + C_r\right] \\
P_r &= G_r^{-1}\left[\left(A_rP_{r-2} + B_r\right)Q_{r-1} + D_r\right] \\
Q_r &= -G_r^{-1}E_r \\
R_r &= G_r^{-1}\left[q_r - A_rR_{r-2} - \left(A_rP_{r-2} + B_r\right)R_{r-1}\right]
\end{aligned}\right\} \tag{E4}$$

The exceptions (in addition to eqs. (E1) and (E2)) are as follows:

At $r = 2$

$$Q_2 = -G_r^{-1}\left(E_2 - B_2C_1^{-1}F_1\right) \tag{E5}$$

At $r = 3$

$$\left.\begin{aligned}
P_3 &= P_r + G_3^{-1}A_3C_1^{-1}F_1 \\
x_2 &= P_3y_2 + Q_3x_3 - G_3^{-1}F_3y_3 + R_3
\end{aligned}\right\} \tag{E6}$$

At $r = 4$

$$Q_4 = Q_r + G_4^{-1}\left[\left(A_4P_2 + B_4\right)G_3^{-1}F_3\right] \tag{E7}$$

At $r = 5$

$$P_5 = P_r + G_5^{-1}A_5G_3^{-1}F_3 \tag{E8}$$

Also at $r = 2N$, A_{2N}, D_{2N}, E_{2N}, and E_{2N-1} are null matrices and equations (E3) reduce to

$$\left.\begin{aligned}
y_n &= R_{2n} \\
x_n &= P_{2n-1}y_n + R_{2n-1}
\end{aligned}\right\} \tag{E9}$$

Thus, the procedure of solving equation (D6) is to sweep down the diagonal to solve for y_n and x_n and then back up the diagonal using equation (E3) to solve for all x_i and y_i.

APPENDIX E

The 3×3 matrix multiplications, additions, and inversions are performed by the CDC library subroutine MATRIX as detailed in reference 17.

APPENDIX F

PROGRAM LISTING

The program listing for the sample problem is as follows:

```
      PROGRAM SADAUS (INPUT, OUTPUT, TAPE5= INPUT)
C IDYM=0 OMITS THE TRANSIENT RESPONSE PROBLEM
C IDYM=1 INCLUDES THE TRANSIENT RESPONSE PROBLEM
C IDYM =2 THE STATIC BUCKLING LOAD WILL BE CALCULATED
C KMAX IS THE MAXIMUM NUMBER OF TIME STEPS WHEN IDYM= 1.
C KMAX IS THE MAXIMUM NUMBER OF STATIC BUCKLING STEPS WHEN IDYM =2
C   NTYPE EQUAL TO 1 IS A CYLINDER
C   NTYPE EQUAL TO 2 IS A CONE
C   NTYPE EQUAL TO 3 IS A SPHERE
C   NTYPE EQUAL TO 4 THE GEOMETRY MUST BE SPECIFIED IN GEOMTY.
C   IF LBCL = 1 THE LEFT BC IS A POLE POINT
C   IF LBCL = 2 THE LEFT BC IS PINNED
C   IF LBCL = 3 THE LEFT BC IS FIXED
C   IF LBCL = 4 THE LEFT BC IS FREE
C   IF LBCL = 5 THE LEFT BC IS A MATRIX OF ELASTIC CONTANTS
C   IF LBCR = 1 THE RIGHT BC IS A POLE POINT
C   IF LBCR = 2 THE RIGHT BC IS PINNED
C   IF LBCR = 3 THE RIGHT BC IS FIXED
C   IF LBCR = 4 THE RIGHT BC IS FREE
C   IF LBCR = 5 THE RIGHT BC IS A MATRIX OF ELASTIC CONSTANTS
C   IF NONLIN = 0 A LINEAR SOLUTION IS OBTAINED
C   IF NONLIN =1 A NONLINEAR SOLUTION IS OBTAINED
C     IF ISTART =0 THE PROBLEM WILL BE SOLVED FOR THE STATIC LOADS FIRST
C     IF ISTART =1 INITIAL DISPLACEMENTS AND VELOCITIES MUST BE GIVEN
      COMMON/BL1/ [illegible], [illegible], PHI0, SUA, DS, DR
      COMMON/BL  /OMEGA1(3,3), ALAMDAL(3,3), OMEGAR(3,3), ALAMDAR(3,3),
     1 [illegible](3), [illegible](3)
      COMMON/BL  /[illegible], I CU, [illegible]
      COMMON/BL  /[illegible]
      COMMON/BL  /[illegible](202,3,3),[illegible](202,3,3),C(202,3,3),D(202,3,3),E(202,3,3),
     1 [illegible](202,3)
      COMMON/BL / C11, C12, C22, K11, K12, K22, D11, D12, D22
      COMMON/BL//X[illegible](202,3), EPS2, EPS
      COMMON/BL / LAM, [illegible], [illegible], NU12, NU21, ETA
      COMMON/BL / IDYM, [illegible], NONLIN, CHAR
      COMMON /BL10/ R(101),PHI(101),DPHI(101)
       COMMON/BL11/ALFA1, ALFA2, T1, T2, ITEMP
      COMMON / BL12/[illegible]
      COMMON / BL13/X1(101,2), X2(101,2), X3(101,2)
      COMMON /BL14/ DS, DEL, [illegible], AN
      COMMON/ BL15/ [illegible]
      COMMON/BL1 / [illegible]
```

APPENDIX F

```
      DIMENSION X(202,3)
      DIMENSION XB(202,3)
      REAL LAMS
      REAL       NU12, NU21, K11, K12, K22
      REAL N12, N3, M12, M3, KB12, K3
      REAL LAM
    1 FORMAT(12X, *THE LEFT OMEGA MATRIX*,40X*THE LEFT LAMBDA MATRIX*,
     1  //5X*(* 3(E12.5, 5X),*)(N)   +(*3(E12.5, 5X)*)(U)   =* E12.5/
     2     5X*(* 3(E12.5, 5X),*)(Q)   +(*3(E12.5, 5X)*)(W)   =* E12.5/
     2     5X*(* 3(E12.5, 5X),*)(M)   +(*3(E12.5, 5X)*)(B)   =* E12.5)
    2 FORMAT(12X, *THE RIGHT OMEGA MATRIX*,40X*THE RIGHT LAMBDA MATRIX*,
     1  //5X*(* 3(E12.5, 5X),*)(N)   +(*3(E12.5, 5X)*)(U)   =* E12.5/
     2     5X*(* 3(E12.5, 5X),*)(Q)   +(*3(E12.5, 5X)*)(W)   =* E12.5/
     2     5X*(* 3(E12.5, 5X),*)(M)   +(*3(E12.5, 5X)*)(B)   =* E12.5)
    6 FORMAT(10X , I5, 2(4X, F11.5), 2(9X, I6), E15.5)
    7 FORMAT(80H
     1                                 )
   13 FORMAT(/5X,*STATION NO*5X*U DOT*15X*W DOT*15X*U DOTDOT*12X*W DOTDO
     1T*//(5X,I5, 4(5X, E15.8) ))
   80 FORMAT(30X*IF IDYM=0 THE SHELL IS LOADED STATICALLY*/30X*IF IDYM=1
     1THE SHELL IS LOADED DYNAMICALLY*/30X*IF IDYM=2 THE SHELL STATIC BU
     2CKLING LOAD IS CALCULATED*//30X*FOR THIS RUN, IDYM=*I20//30X*NUMBE
     3R OF STATIONS =*I19//30X*MERIDIAN/REFERENCE RATIO =* F20.6 //30X
     4     *RADIUS/REFERENCE RATIO =*,F22.6//30X,*THICK/ REF RAD RATIO =*
     5     F24.6//15X*E1/E0 =* F20.6,23X*E2/E0 =* F23.6//15X*NU12 =*F21.6
     6,23X*NU21 =*F24.6//15X*E0/S0 =*F20.6,23X*REF DIST =*F20.6//40X*IF
     5NONLIN = 0 ONLY LINEAR TERMS ARE USED*/40X*IF NONLIN = 1 NONLINEAR
     6 TERMS USED*//40X*FOR THIS RUN NONLIN = *I6/)
   85 FORMAT(//4X, *STATION NO*,7X,*N-S RESULTANT*,
     1 5X,*SHEAR FORCE*,7X, *M-S RESULTANT*, 5X, *U-DEFORMATION*, 5X,
     2*W-DEFORMATION*, 5X, *BETA ROTATION*//(4X,I6,6X,6(3X,E15.7))/)
   97 FORMAT(10X*NO. OF ITERATIONS= *I6, 10X*ERROR NORM =*E15.8, 10X*CYC
     1LES THIS ITERATION =*I6)
  203 FORMAT( 5X,*PL(I) =*E15.8,3X,*PS(I) =*E15.8,3X*SL(1) =*F15.8,3X
     1 *SL(2) =*E15.8,3X*SL(3) =*E15.8/5X*LOAD CHANGES =*I6,5X*DEL =*
     2F9.6,5X*ITERATIONS =*I6,4X*NO OF CYCLES =*I4,4X*XNORM =*E12.5)
  117 FORMAT(40X, *THE MAXIMUM SHELL RISE IS *F17.8)
  116 FORMAT(40X, *THE SHELL THICKNESS =* F17.8)
  231 FORMAT (5X*WMAX= *F15.8, 5X*I= *I6)
  241 FORMAT(//40X*THE GEOMETRY OF THE SHELL FOLLOWS*/5X*STATION NO*
     1 4X*MERIDIAN DIS* 8X*RADIAL DIS*10X*ANGLE(RAD)*10X*CURVATURE(DFI/D
     2S)*/)
  242 FORMAT(5X, I6, 4(5X, E15.6))
```

APPENDIX F

```
243 FORMAT(/10X*KOUNT* 10X*TAU* 12X*DEL* 12X *ITERATIONS* 5X
   1 *CYCLES* 4X *XNORM*)
244 FORMAT(*1*)
    NAMELIST/GIVEN/NTYPE, N, RO, SO, HO, EO, E1, E2, NU12, NU21, SIGO,
   1NONLIN, CONV, PHIC, LBCL, LBCR, PP, PPS,  SL, SR, CHAR,
   2  IDYM, FMAX, DTAU , T1, T2, ALFA1, ALFA2, ITEMP, IFREQ, ISTART
    NAMELIST/ELBCL/ OMEGAL, ALAMDAL
    NAMELIST/ELBCR/ OMEGAR, ALAMDAR
     DATA PI / 3.%41592653589793 /
101 READ GIVEN
    IF ( EOF, 5) 301, 302
301 STOP
302 CONTINUE
    PRINT GIVEN
    READ 7
    PRINT 7
    IF (LBCL .EQ. 5) READ ELBCL
    IF (LBCR .EQ. 5) READ ELBCR
    LAM= HO/CHAR
    SOA=SO/CHAR
    ROA=RO/CHAR
    ETA=EO/SIGO
    E10= E1/EO
    E20=E2/EO
    NONCONV=1
    IBUCK=20
    KSTOP=0
    IACC=0
    DAVE=0.
     INIT=0
    KOUNT=0
    ITER =0
    LCHANGE=0
    XNORM=0.
    DP=PP
    DPS=PPS
    DS1=SL(1)
    DS22=SL(2)
    DS3=SL(3)
    DT1=0.
    DT2=0.
    PRINT 244
    CALL GEOMTY(NTYPE)
```

APPENDIX F

```
      PRINT 80, IDYM, N, SOA, ROA, LAM,E10, E20, NU12, NU21, ETA, CHAR,
     1 NONLIN
      DS2=DS/2.
      SMER=-DS
      DEL = (PHI(2)- PHI(1))/2.
      PRINT 241
      DO 240 I=1, N
      RI= R(I)+DS2*COS(PHI(I))
      FI= PHI(I)+DEL
      SMER=SMER+DS
  240 PRINT 242,I, SMER, RI, FI, DPHI(I)
      IF(NTYPE.NE.3) GO TO 211
      ZO = ROA*(1.-COS(RPHI))*ETA
      ZSUM = ZO*ZO*ROA/(ETA*2.)
      ZOA= ZO/ETA
      PRINT 117, ZOA
  211 CONTINUE
      TT=T(1)
      PRINT 116, TT
      L=2*N
       L1=L-1
      EPS=DTAU
      EPS2=DTAU**2
      DO 21 J=1,L
      DO 21 M=1,3
      XOLD(J,M)=0.
      XB(J,M)=0.
   21 X(J,M)= 0.
      DO 25 I=1,N
      DO 25 M=1,2
      X1(I,M)=0.
      X2(I,M)=0.
   25 X3(I,M)=0.
      ITER=0
      NCYCLE=0
      TAU=-EPS
   92 ITER =ITER+1
      NCYCLE=NCYCLE+1
      CALL BCCON (LBCL, LBCR, X, ITER)
      IF (ITER .NE. 1) GO TO 11
      PRINT 1, ((C(1,M,MM), MM=1,3), (D(1,M,MM), MM=1,3), SL(M), M=1,3)
      PRINT 2, ((B(L,M,MM), MM=1,3), (C(L,M,MM), MM=1,3), SR(M), M=1,3)
   11 CONTINUE
```

```
      IF (ISTART .EQ. 1 ) CALL INITIAL(X, LBCL)
      IF (ISTART .NE. 1 ) GO TO 10
      IF (ITER .EQ. 1) GO TO 93
   10 DO 84 I= 2,N
      CALL ABCDES(X, I, NCYCLE)
   84 CONTINUE
      CALL FUNCT(X, N)
      CALL PCTTER(X)
      IF (ITER .EQ. 1) GO TO 94
      XNORM=0.
      DXSUM=0.
      DO 95 J=1,L
      DO 95 M=1,3
      XNORM=XNORM+XOLD(J,M)**2
   95 DXSUM=DXSUM+X(J,M)**2
      IF (XNORM .EQ. 0.) GO TO 94
      XNORM= SQRT(DXSUM/XNORM)
   94 DO 96 J=1,L
      DO 96 M=1,3
   96 X(J,M)=XOLD(J,M)
      IF (IDYM .EQ. 0)
     1PRINT 85, (I, (X(2*I-1,J),J=1,3),(X(2*I,J),J=1,3),I=1,N)
      IF ((IDYM .EQ. 0) .AND. (NCNLIN .EQ. 0)) GO TO 105
      IF ((IDYM .EQ. 0) .AND. (ITER .EQ. 1)) GO TO 92
      IF (XNORM .EQ. 0.) GO TO 93
      IF (ITER .EQ. 1)  GO TO 92
      IF ((NCYCLE .GT. IBUCK) .AND. (IDYM .EQ. 2))GO TO 201
      IF (XNORM-CONV) 93, 93, 92
   93 CONTINUE
      TAU=TAU+EPS
      IF ( MOD(KOUNT,IFREQ) .NE. 0) GO TO 9
    8 PRINT 85, (I, (X(2*I-1,J),J=1,3),(X(2*I,J),J=1,3),I=1,N)
  105 CALL ANSWERS(NTYPE, X, LBCL, LBCR)
      IF (IP(KOUNT) .EQ. 1) GO TO 103
      IF ((IDYM .EQ. 1) .AND. (MOD(KOUNT,IFREQ) .EQ. 0)) PRINT 243
  103 CONTINUE
      IF (IDYM .EQ. 0) GO TO 101
    9 CONTINUE
      IF(NTYPE.NE.3) GO TO 206
      WSUM= 0.
      DO 5 I=2,N
      J=2*I
    5 WSUM=WSUM+(X(J,2)+X(J-2,2))/2.*R(I)*COS(PHI(I))*DS
```

```
      DAVE=WSUM/ZSUM
  206 CONTINLE
      IF(IDYM.EQ.2) GO TO 204
      IF (ISTART .EQ. 1) GO TO 12
      IF (IP(KOUNT) .NE. 1) GO TO 3
      DO 4 J=4,L,2
      DO 4 M=1,2
      I=J/2
      XJ=(X(J,M)+X(J-2,M))/2.
      XD     =(11.*XJ-18.*X1(I,M)+9. *X2(I,M) -2.*X3(I,M))/(EPS*6.)
      X1(I,M)=XJ
    4 X2(I,M)= XD
   12 ISTART=0
      NCYCLE=C
      INIT=KOUNT
      IACC=1
      DO 26 I=2,N
      CALL ABCDFS(X, I, NCYCLE)
      J=2*I-1
      DO 26 M=1,2
       AX=0.
       BX=0.
       CX=0.
       DX=0.
       EX=0.
      DO 24 MM=1,3
      AX= AX+A(J   ,M,MM)*X(J-2,MM)
      BX= BX+B(J   ,M,MM)*X(J-1,MM)
      CX= CX+C(J   ,M,MM)*X(J   ,MM)
      DX= DX+D(J   ,M,MM)*X(J+1,MM)
      IF (I .EQ. N) GO TO 24
      EX= EX+E(J   ,M,MM)*X(J+2,MM)
   24 CONTINUE
   26 X3(I,M)=AX+BX+CX+DX+EX-S(J,M)
      PRINT 13, (I, X2(I,1), X2(I,2), X3(I,1), X3(I,2), I=1,N)
      PRINT 243
      IACC=0
    3 CONTINUE
      IF (IDYM .EQ. 1) PRINT 6, KCUNT, TAU, DAVE, ITER, NCYCLE, XNORM
      NCYCLE =0
      KOUNT=KOUNT+1
      IF (KOUNT .EQ. (KMAX+1)) GO TO 102
      IF (IDYM .EQ. 1) GO TO 92
```

APPENDIX F

```
 204 CONTINUE
     IF (IDYM .NE. 2 ) GO TO 102
     PRINT 911, T1, T2
 911 FORMAT(5X, *T1= * F15.7, 5X*T2=*F15.7)
     WMAX=0.
     DO 230 I=1,N
     J=2*I
     IF (X(J,2) .GE. WMAX) GO TO 230
     IO=I
     WMAX =X(J,2)
 230 CONTINUE
     PRINT 231,WMAX, IO
     KOUNT =KOUNT+1
     IF (NTYPE .NE. 3) DAVE= WMAX
     PRINT 203, FF, PPS, (SL(I), I=1,3), LCHANGE, DAVE, ITER, NCYCLE,
    1 XNORM
     IF ((NTYPE .EQ. 3) .AND. (ABS(DAVE) .GT. 2.0)) GO TO 101
 201 ONE=1.
     IF (KOUNT .GT. KMAX) GO TO 102
     IF (LCHANGE .EQ. 5) ONE=-1.
     PP=PP+DP*ONE
     PPS=PPS+DPS*ONE
     SL(1)=SL(1)+DS1*ONE
     SL(2)=SL(2)+DS22*ONE
     SL(3)=SL(3)+DS3*ONE
     T1=T1+DT1*ONE
     T2=T2+DT2*ONE
     IF ((KSTOP .EQ. (KOUNT-2)) .AND. (LCHANGE .EQ. 5)) ONE=-.5
     IF ((KSTOP .EQ. (KOUNT-2)) .AND. (LCHANGE .EQ. 5)) GO TO 226
     NONCONV=1
     IF (NCYCLE .LE. IBUCK ) GO TO 202
     DO 205 I=1,N
     J=2*I-1
     DO 205 M=1,2
     XOLD(J,1)=X1(I,M)
     XOLD(J,2)=X2(I,M)
     XOLD(J,3)=X3(I,M)
 205 J=J+1
 226 CONTINUE
     KSTOP= KOUNT
     NONCONV=0
     PP=PP-DP*2.*ONE
     PPS=PPS-DPS*2.*ONE
```

```
      SL(1)=SL(1)-DS1 *2.*ONE
      SL(2)=SL(2)-DS22*2.*ONE
      SL(3)=SL(3)-DS3 *2.*ONE
      T1=T1-DT1*2.*ONE
      T2=T2-DT2*2.*ONE
      DPS=DPS/5.
      DP=DP/5.
      DS1=DS1/5.
      DS2=DS22/5.
      DS3=DS3/5.
      DT1=DT1/5.
      DT2=DT2/5.
      LCHANGE=LCHANGE+1
      IF(LCHANGE.EQ.5) GO TO 215
      IF (LCHANGE .EQ. 6) GO TO 101
      GO TO 224
  215 CONTINUE
      PP=PP-DP
      PPS=PPS-DPS
      SL(1)=SL(1)-DS1
      SL(2)=SL(2)-DS22
      SL(3)=SL(3)-DS3
      T1=T1-DT1
      T2=T2-DT2
  208 DO 216 I=1,N
      J=2*I-1
      DO 216 M=1,2
      X(J,1)=X1(I,M)+5.*(X1(I,M)-XB(J,1))
      X(J,2)=X2(I,M)+5.*(X2(I,M)-XB(J,2))
      X(J,3)=X3(I,M)+5.*(X3(I,M)-XB(J,3))
      XOLD(J,1)=X(J,1)
      XOLD(J,2)=X(J,2)
      XOLD(J,3)=X(J,3)
  216 J=J+1
  224 CONTINUE
      PP=PP+DP
      PPS=PPS+DPS
      SL(1)=SL(1)+DS1
      SL(2)=SL(2)+DS22
      SL(3)=SL(3)+DS3
      T1=T1+DT1
      T2=T2+DT2
  202 NCYCLE =0
```

APPENDIX F

```
      DO 207  I=1,N
      J=2*I-1
      DO 207 M=1,2
      IF (NONCONV .EQ. 0 ) GO TO 209
      XB(J,1)=X1(I,M)
      XB(J,2)=X2(I,M)
      XB(J,3)=X3(I,M)
      X1(I,M)=X(J,1)
      X2(I,M)=X(J,2)
      X3(I,M)=X(J,3)
  209 X(J,1)=XOLD(J,1)
      X(J,2)=XOLD(J,2)
      X(J,3)=XOLD(J,3)
  207 J=J+1
      GO TO 92
  102 GO TO 101
      END OF PROGRAM
```

APPENDIX F

```
      SUBROUTINE POTTER(X)
C     THIS SUBROUTINE SOLVES A FIVE ELEMENT BANDWIDTH DIAGONAL MATRIX
C     WHERE EACH ELEMENT IS ITSELF A 3X3 MATRIX.  THE SOLUTION IS BASED
C     ON POTTERS METHOD.
      COMMON/BL1/ ROA, N, PHIO, SOA, DS, DR
      COMMON/BL5/A(202,3,3),B(202,3,3),C(202,3,3),D(202,3,3),E(202,3,3),
     1  S(202,3)
      COMMON/BL7/XOLD(202,3), EPS2, EPS
      COMMON/BL8/ LAM, E10, E20, NU12, NU21, ETA
      COMMON/BL9/ IDYN, KMAX, NONLIN, CHAR
      COMMON /BL10/ R(101),PHI(101),DPHI(101)
      DIMENSION X(202,3)
      DIMENSION AA(3,3), BB(3,3), CC(3,3), DD(3,3), EE(3,3), FF(3,3)
      DIMENSION SS(3), F1(3,3), F3(3,3), P(3,3), Q(3,3), AR(3), G(3,3)
      L=2*N
      L1=L-1
      DO 46 M=1,3
      DO 46 MM=1,3
      F1(M,MM)=0.
   46 F3(M,MM)=0.
      IF (NONLIN .EQ. 0) GO TO 11
       IF (C(1,2,2) .EQ. 1.) D(1,2,3)= D(1,2,3)+X(1,1)
       IF (C(1,2,2) .EQ. 1.) C(1,2,1)= C(1,2,1)+X(2,3)
       IF (B(L,2,2) .EQ. 1.) B(L1,2,1)=B(L1,2,1)+X(L,3)
       IF (B(L,2,2) .EQ. 1.) C(L,2,3)=C(L,2,3)+X(L1,1)
       DO 23 J=2,L1
      C(J,1,3)=C(J,1,3)+(X(J+2,3)+X(J,3))/(4.*ETA)
      E(J,1,3)=E(J,1,3)+(X(J+2,3)+X(J,3))/(4.*ETA)
       J=J+1
      I=(J+1)/2
      DFI=DPHI(I)
      CSR= COS(PHI(I))/R(I)
       D(J,1,3)=D(J,1,3)+ DFI/(2.*ETA)*X(J,1)
       C(J,1,1)=C(J,1,1)+ DFI/(2.*ETA)*X(J+1,3)
       B(J,1,3)=B(J,1,3)+ DFI/(2.*ETA)*X(J-2,1)
       A(J,1,1)=A(J,1,1)+ DFI/(2.*ETA)*X(J-1,3)
      D(J,2,3)=D(J,2,3)+ (CSR/(2.*ETA)+1./(ETA*DS))*X(J,1)
      C(J,2,1)=C(J,2,1)+ (CSR/(2.*ETA)+1./(ETA*DS))*X(J+1,3)
      B(J,2,3)=B(J,2,3)+ (CSR/(2.*ETA)-1./(ETA*DS))*X(J-2,1)
   23 A(J,2,1)=A(J,2,1)+ (CSR/(2.*ETA)-1./(ETA*DS))*X(J-1,3)
   11 CONTINUE
      DO 76 K=1,3,2
```

```
      IF (C(1,K,K) .EQ. 1.) GO TO 76
      DO 77 M=1,3
      P(K,M)= C(1,K,M)
      Q(K,M)= D(1,K,M)
      C(1,K,M)= B(2,K,M)
      D(1,K,M)= C(2,K,M)
      E(1,K,M)= D(2,K,M)
      E1( K,M)= E(2,K,M)
      B(2,K,M)= P(K,M)
      C(2,K,M)= Q(K,M)
      D(2,K,M)= 0.
   77 E(2,K,M)= 0.
      AR(K)=S(1,K)
      S(1,K)= S(2,K)
      S(2,K)= AR(K)
   76 CONTINUE
      IF (C(1,2,2) .EQ. 1.) GO TO 79
      DO 78 M=1,3
      P(2,M)=C(1,2,M)
      Q(2,M)=D(1,2,M)
      C(1,2,M)= A(3,3,M)
      D(1,2,M)= B(3,3,M)
      E(1,2,M)= C(3,3,M)
      E1( 2,M)= D(3,3,M)
      A(3,3,M)= 0.
      B(3,3,M)= A(4,3,M)
      C(3,3,M)= B(4,3,M)
      D(3,3,M)= C(4,3,M)
      E(3,3,M)= D(4,3,M)
      E3( 3,M)= E(4,3,M)
      A(4,3,M)= C(2,2,M)
      B(4,3,M)= D(2,2,M)
      C(4,3,M)= E(2,2,M)
      D(4,3,M)= 0.
      E(4,3,M)= 0.
      B(2,2,M)= P(2,M)
      C(2,2,M)= Q(2,M)
      D(2,2,M)= 0.
   78 E(2,2,M)= 0.
      AR(2)=S(1,2)
      S(1,2)= S(3,3)
      S(3,3)= S(4,3)
      S(4,3)= S(2,2)
```

```
      S(2,2)=AR(2)
   79 CONTINUE
      DO 51 M=1,3
      DO 52 MM=1,3
      CC(M,MM)= C(1, M,MM)
      DD(M,MM)=-D(1, M,MM)
   52 EE(M,MM)=-E(1, M,MM)
   51 SS(M)= S(1,M)
      CALL MATRIX(10, 3, 3, 0, CC, 3, CEE)
      CALL MATRIX(20, 3, 3, 3, CC, 3, DD, 3, DD, 3)
      CALL MATRIX(20, 3, 3, 3, CC, 3, EE, 3, EE, 3)
      CALL MATRIX(20, 3, 3, 3, CC, 3, F1, 3, F1, 3)
      CALL MATRIX(20, 3, 3, 1, CC, 3, SS, 3, SS, 3)
      DO 54 M=1,3
      DO 55 MM=1,3
      D(1,M,MM)= DD(M,MM)
      E(1,M,MM)= EE(M,MM)
      CC(M,MM)= C(2,M,MM)
   55 BB(M,MM)= B(2, M,MM)
   54 S(1,M)= SS(M)
      CALL MATRIX(20, 3, 3, 1, BB, 3, SS, 3, SS, 3)
      CALL MATRIX(20, 3, 3, 3, BB, 3, DD, 3,  G, 3)
      CALL MATRIX(21, 3, 3, 0, G, 3, CC, 3,  G, 3)
      CALL MATRIX(10, 3, 3, 0,  C, 3, GEE)
      CALL MATRIX(20, 3, 3, 3, BB, 3, EE, 3, DD, 3)
      CALL MATRIX(20, 3, 3, 3, BB, 3, F1, 3, AA, 3)
      DO 58 M=1,3
      DO 59 MM=1,3
          DD(M,MM)= -DD(M,MM)-D(2,M,MM)
   59     EE(M,MM)=  AA(M,MM)-E(2,M,MM)
   58     SS(M)= -SS(M)+S(2,M)
      CALL MATRIX(20, 3, 3, 3,  G, 3, DD, 3, DD, 3)
      CALL MATRIX(20, 3, 3, 3,  G, 3, EE, 3, EE, 3)
      CALL MATRIX(20, 3, 3, 1,  G, 3, SS, 3, SS, 3)
       L1=L-1
      DO 64 J=3,L1
        DO 65 M=1,3
        DO 66 MM=1,3
        D(J-1, M, MM)=DD(M,MM)
        E(J-1, M, MM)=EE(M,MM)
        AA(M,MM)= A(J,M,MM)
        BB(M,MM)= B(J,M,MM)
        P(M,MM)= D(J-2,M,MM)
```

```
66     Q(M,MM)= E(J-2,M,MM)
       AR(M)= S(J-2, M)
65     S(J-1,M)= SS(M)
      CALL MATRIX(20, 3, 3, 3, AA, 3,  P, 3,  P, 3)
      CALL MATRIX(20, 3, 3, 3, AA, 3,  Q, 3,  Q, 3)
      CALL MATRIX(20, 3, 3, 1, AA, 3, AR, 3, AR, 3)
      IF (J .EQ. 3) CALL MATRIX(20, 3, 3, 3, AA, 3, F1, 3, AA, 3)
      IF (J .EQ. 5) CALL MATRIX(20, 3, 3, 3, AA, 3, F3, 3, AA, 3)
      CALL MATRIX(21, 3, 3, 0, P, 3, BB, 3, BB, 3)
      IF (J .EQ. 4) CALL MATRIX(20, 3, 3, 3, BB, 3, F3, 3, AA, 3)
      CALL MATRIX(20, 3, 3, 3, BB, 3, DD, 3, DD, 3)
      CALL MATRIX(20, 3, 3, 3, BB, 3, EE, 3, EE, 3)
      CALL MATRIX(20, 3, 3, 1, BB, 3, SS, 3, SS, 3)
        DO 60 M=1,3
        DO 67 MM=1,3
        C(M,MM)= DD(M,MM)+ Q(M,MM)+C(J,M,MM)
       DD(M,MM)=-EE(M,MM)- D(J,M,MM)
       IF (J .EQ. 3)  DD(M, MM)= DD(M,MM)+AA(M,MM)
       IF (J .EQ. 5)  DD(M, MM)= DD(M,MM)+AA(M,MM)
      EE(M,MM)= -E(J, M, MM)
      IF (J .EQ. 4) EE(M,MM)= -E(J,M,MM)+AA(M,MM)
67    CONTINUE
60    SS(M)= S(J,M)-AR(M)-SS(M)
      CALL MATRIX(10, 3, 3, 0,  G, 3, GEE)
      CALL MATRIX(20, 3, 3, 3,  G, 3, DD, 3, DD, 3)
      CALL MATRIX(20, 3, 3, 3,  G, 3, EE, 3, EE, 3)
      CALL MATRIX(20, 3, 3, 1,  G, 3, SS, 3, SS, 3)
      IF (J .EQ. 3) CALL MATRIX(20, 3, 3, 3,  G, 3, F3, 3, F3, 3)
64    CONTINUE
      DO 68 M=1,3
      DO 69 MM=1,3
      D(L1,M,MM)= DD(M,MM)
       P(M,MM)= D(L-2, M,MM)
       Q(M,MM)=E(L-2, M,MM)
      BB(M,MM)= B(L,M,MM)
      AA(M,MM)= A(L,M,MM)
69    CC(M,MM)= C(L,M,MM)
      AR(M)= S(L-2,M)
68    S(L1, M)= SS(M)
      CALL MATRIX(20, 3, 3, 3, AA, 3,  P, 3,  P, 3)
      CALL MATRIX(20, 3, 3, 3, AA, 3,  Q, 3,  Q, 3)
      CALL MATRIX(20, 3, 3, 1, AA, 3, AR, 3, AR, 3)
      CALL MATRIX(21, 3, 3, 0, P, 3, BB, 3, BB, 3)
```

APPENDIX F

```
      CALL MATRIX(20, 3, 3, 1, BB, 3, SS, 3, SS, 3)
      CALL MATRIX(20, 3, 3, 3, BB, 3, DD, 3, DD, 3)
      DO 70 M=1,3
      DO 71 MM=1,3
   71 G(M,MM)= DD(M,MM)+Q(M,MM)+C(L,M,MM)
   70 SS(M)= S(L,M)-AR(M)-SS(M)
      CALL MATRIX(10, 3, 3, 0,  G, 3, GFE)
      CALL MATRIX(20, 3, 3, 1,  G, 3, SS, 3, SS, 3)
      DO 30 M=1,3
      DO 32 MM=1,3
   32 DD(M,MM)= D(L1,M,MM)
   30 X(L,M)= SS(M)
      CALL MATRIX(20, 3, 3, 1, DD, 3, SS, 3, SS, 3)
      DO 33 M=1,3
   33 X(L1,M)= SS(M)+S(L1,M)
      DO 34 K=2,L1
      J=L-K
      DO 35 M=1,3
      DO 36 MM=1,3
      P(M,MM)= D(J,M,MM)
   36 G(M,MM)= E(J,M,MM)
      AR(M)= X(J+1,M)
   35 SS(M)= X(J+2,M)
      CALL MATRIX(20, 3, 3, 1,  P, 3, AR, 3, AR, 3)
      CALL MATRIX(20, 3, 3, 1,  G, 3, SS, 3, SS, 3)
      DO 39 M=1,3
   39 X(J,M)= AR(M)+ SS(M)+ S(J,M)
      IF (J .GT. 3) GO TO 34
      IF (J .EQ. 2) GO TO 34
      DO 38 M=1,3
   38 SS(M)= X(J+3,M)
      IF (J .EQ. 1) CALL MATRIX(20, 3, 3, 1, F1, 3, SS, 3, SS, 3)
      IF (J .EQ. 3) CALL MATRIX(20, 3, 3, 1, F3, 3, SS, 3, SS, 3)
      DO 40 M=1,3
   40 X(J,M)= X(J,M)- SS(M)
   34 CONTINUE
      DO 22 J=1,L
       DO 22 M=1,3
   22 XOLD(J,M)=X(J,M)+XOLD(J,M)
      RETURN
      END OF SUBROUTINE POTTER
```

APPENDIX F

```
      SUBROUTINE ABCDFS(X, I, NCYCLE)
C     THIS SUBROUTINE DEFINES ALL ELEMENTS OF THE A, B, C, D AND E
C     MATRICES AS WELL AS THE S VECTOR WHICH IS EQUIVALENT TO THE Q
C     VECTOR IN THE USERS MANUAL
      COMMON/BL1/ RCA, N, PHIC, SCA, DS, DR
      COMMON/BL3/PP, IACC, INIT
      COMMON/BL4/PPS
      COMMON/BL5/A(202,3,3),B(202,3,3),C(202,3,3),D(202,3,3),E(202,3,3),
     1 S(202,3)
      COMMON/BL6/ C11, C12, C22, K11, K12, K22, D11, D12, D22
      COMMON/BL7/XCLD(202,3), EPS2, EPS
      COMMON/BL8/ LAM, E10, E20, NU12, NU21, ETA
      COMMON/BL9/ ICYM, KMAX, NONLIN, CHAR
      COMMON /BL10/ R(101),PHI(101),DPHI(101)
       COMMON/BL11/ALFA1, ALFA2, T1, T2, ITEMP
      COMMON/BL12/KCUNT
      DIMENSION X(202,3)
      REAL       NU12, NU21, K11, K12, K22
      REAL N12, N3, M12, M3, KB12, K3
      REAL LAM
      J= 2*(I-1)
      CALL STIF(I, LAM, E10, E20, NU12, NU21)
      CDK = (C11*D11-K11**2)/LAM**2
      ETANUS= ETA/(1.-NU21*NU12)
      ETAL= ETA/LAM
      GL2= CDK*LAM**2
      GL4= GL2*LAM**2
      N12= (C12*C11-K11*K12)/GL2
      M12= (K12*C11-K11*C12)/CDK
      E12= (2.*C12*K11*K12-C12*C12*D11-C11*K12*K12)/GL2
      KB12= (C12*K11*D12-C12*D11*K12+K12*K12*K11-K12*C11*D12)/GL2
      N3=(K12*D11-K11*D12)/GL4
      M3= (D12*C11-K11*K12)/GL2
      E3=(K11*K12*K12- C12*D11*K12 +C12*D12*K11-C11*D12*K12)/GL4
      K3= (2.*D12*K11*K12-D11*K12*K12-C11*D12*D12)/GL4
      UE=(D11*C12- K11*K12)/GL2
      UK= (K11*C12-C11*K12)/GL2
      BE= (K11*C12-C11*K12)/GL2
      BK= (C11*D12-K11*K12)/GL2
      RI= R(I)
      FI= PHI(I)
      DFI=DPHI(I)
```

APPENDIX F

```
CS= COS(FI)
SN = SIN(FI)
CSR=CS/RI
SNR=SN/RI
TN1=F10*ETANUS*(ALFA1+NU12*ALFA2)* TDZ(I)
TN2=F20*ETANUS*(ALFA2+NU21*ALFA1)* TDZ(I)
TM1=F10*ETAL   *(ALFA1+NU12*ALFA2)*TZDZ(I)
TM2=F20*ETAL   *(ALFA2+NU21*ALFA1)*TZDZ(I)
A(J,1,1)=0.
A(J,1,2)=0.
A(J,1,3)=0.
A(J,2,1)=0.
A(J,2,2)=0.
A(J,2,3)=0.
A(J,3,1)=0.
A(J,3,2)=0.
A(J,3,3)=0.
B(J,1,1)=-D11/(GL2*2.)
B(J,1,2)=0.
B(J,1,3)=K11/(CDK*2.)
B(J,2,1)=0.
B(J,2,2)=0.
B(J,2,3)= 0.
B(J,3,1)=-K11/(GL2*2.)
B(J,3,2)= 0.
B(J,3,3)=C11/(CDK*2.)
C(J,1,1)= -1./DS+LE*CSR/2.
C(J,1,2)= (DFI+LE*SNR)/2.
C(J,1,3)= CSR*UK/2.
C(J,2,1)= -DFI/2.
C(J,2,2)= -1./DS
C(J,2,3)= -.5
C(J,3,1)=  BF*CSR/2.
C(J,3,2)=  BF*SNR/2.
C(J,3,3)= -1./DS+BK*CSR/2.
D(J,1,1)=-D11/(GL2*2.)
D(J,1,2)=  0.
D(J,1,3)= K11/(CDK*2.)
D(J,2,1)=0.
D(J,2,2)=0.
D(J,2,3)= 0.
D(J,3,1)=-K11/(GL2*2.)
D(J,3,2)= 0.
```

```
      D(J,3,3)=C11/(CDK*2.)
      E(J,1,1)=  1./DS+UE*CSR/2.
      E(J,1,2)= (DFI+UE*SNR)/2.
      E(J,1,3)= CSR*UK/2.
      E(J,2,1)= -DFI/2.
      E(J,2,2)=  1./DS
      E(J,2,3)= -.5
      E(J,3,1)=  BE*CSR/2.
      E(J,3,2)=  BE*SNR/2.
      E(J,3,3)=  1./DS+BK*CSR/2.
      S(J,1)=D11/(LAM**2*CDK)*TN1 -K11/CDK*TM1
      S(J,2)=0.
      S(J,3)= K11/(LAM**2*CDK)*TN1-C11/CDK*TM1
      IF (NONLIN .EQ. 0) GO TO 10
      S(J,1)= S(J,1)- (X(J+2,3)+X(J,3))**2/ETA/8.
   10 J=J+1
      A(J,1,1)= -1./DS+CSR*(1.-N12)/2.
      A(J,1,2)= DFI/2.
      A(J,1,3)=-CSR* M12/2.
      A(J,2,1)= -(DFI+SNR* N12)/2.
      A(J,2,2)= (-1./DS+CSR/2.)
      A(J,2,3)=-SNR*M12/2.
      A(J,3,1)=-CSR*N3/2.
      A(J,3,2)= -.5/LAM**2
      A(J,3,3)=-1./DS+CSR*(1.-M3)/2.
      B(J,1,1)=  -(E12+C22)*CSR*CSR/2.
      B(J,1,2)=  -(E12+C22)*CSR*SNR/2.
      B(J,1,3)=  (KB12+K22)*CSR*CSR/2.
      B(J,2,1)=  -(E12+C22)*CSR*SNR/2.
      B(J,2,2)=  -(E12+C22)*SNR*SNR/2.
      B(J,2,3)=  (KB12+K22)*CSR*SNR/2.
      B(J,3,1)= -CSR*CSR*(E3+K22/LAM**2)/2.
      B(J,3,2)= -CSR*SNR*(E3+K22/LAM**2)/2.
      B(J,3,3)=  CSR*CSR*(K3+D22/LAM**2)/2.
      C(J,1,1)=  1./DS+CSR*(1.-N12)/2.
      C(J,1,2)= DFI/2.
      C(J,1,3)=-CSR* M12/2.
      C(J,2,1)= -(DFI+SNR* N12)/2.
      C(J,2,2)= ( 1./DS+CSR/2.)
      C(J,2,3)=-SNR*M12/2.
      C(J,3,1)=-CSR*N3/2.
      C(J,3,2)= -.5/LAM**2
      C(J,3,3)= 1./DS+CSR*(1.-M3)/2.
```

APPENDIX F

```
      D(J,1,1)=  -(E12+C22)*CSR*CSR/2.
        D(J,1,2)=  -(E12+C22)*CSR*SNR/2.
      D(J,1,3)=   (KB12+K22)*CSR*CSR/2.
      D(J,2,1)=  -(E12+C22)*CSR*SNR/2.
      D(J,2,2)=  -(E12+C22)*SNR*SNR/2.
      D(J,2,3)=   (KB12+K22)*CSR*SNR/2.
      D(J,3,1)= -CSR*CSR*(E3+K22/LAM**2)/2.
      D(J,3,2)= -CSR*SNR*(E3+K22/LAM**2)/2.
      D(J,3,3)=  CSR*CSR*(K3+D22/LAM**2)/2.
      E(J,1,1)=0.
      E(J,1,2)=0.
      E(J,1,3)=0.
      E(J,2,1)=0.
      E(J,2,2)=0.
      E(J,2,3)=0.
      E(J,3,1)=0.
      E(J,3,2)=0.
      E(J,3,3)=0.
      S(J,1)= -PS(I)  +CSR*(N12*TN1+M12*TM1-TN2)
      S(J,2)= -PL(I)  +SNR*(N12*TN1+M12*TM1-TN2)
      S(J,3)= CSR*(N3*TN1+M3*TM1-TM2)
      IF (IDYM .NE. 1) GO TO 83
      IF (IACC .EQ. 1) GO TO 82
      AL= 2./EPS2*T(I)
      IF (KOUNT .EQ. INIT) AL=0.
      IF (KOUNT .EQ. INIT+1) AL= 6./EPS2*T(I)
      B(J,1,1)=B(J,1,1)-AL/2.
      D(J,1,1)=D(J,1,1)-AL/2.
      B(J,2,2)=B(J,2,2)-AL/2.
      D(J,2,2)=D(J,2,2)-AL/2.
   83 CONTINUE
      IF (IDYM .EQ. 1) CALL DYNAMIC(S,N,J, I, NCYCLE)
   82 IF (NONLIN .EQ. 0) GO TO 84
      S(J,1)= S(J,1)-DFI/ETA*(X(J+1, 3)*X(J,1)+X(J-1,3)*X(J-2,1))/2.
      S(J,2)=S(J,2)-CSR/ETA*(X(J+1,3)*X(J,1)+X(J-1,3)*X(J-2,1))/2.-
     1  (X(J+1,3)*X(J,1)-X(J-1,3)*X(J-2,1))/ETA/DS
   84 CONTINUE
      RETURN
      END OF SUBROUTINE ABCDES
```

APPENDIX F

```
      SUBROUTINE GEOMTY (NTYPE)
C    THIS SUBROUTINE CALCULATES THE GEOMETRY AT THE STATION MIDPOINT(I-1/2)
C     IF NTYPE = 4 THE USER MUST DEFINE R(I), PHI(I) AND DPHI(I) 36
C     CORRESPOND WITH HIS DESIRED GEOMETRY.  THESE DEFINITIONS SHOULD BE
C     LOCATED BETWEEN THE LABEL STATEMENTS 50 AND 60.
      COMMON/BL1/ ROA, N, PHIO, SOA, DS, DR
      COMMON /BL10/ R(101),PHI(101),DPHI(101)
      COMMON /BL14/ DS2, DEL, RPHI, AN
       DATA PI / 3.141592653589793 /
      RPHI = PHIO* PI / 180.0
      N1= N-1
      AN =N1
      IF (NTYPE .NE. 1    )  GO TO 10
C     THE SHELL IS A CYLINDER WHEN NTYPE IS 1
      DS = SOA/ AN
      DO 20 I = 1,N
      R(I) = ROA
      PHI(I) = RPHI
   20 DPHI(I) = 0.0
      GO TO 60
   10 IF (NTYPE .NE. 2    ) GO TO 30
C       THE SHELL IS A CONE WHEN NTYPE IS 2
      DS = SOA/ AN
      DR = DS * COS(RPHI)
      R(1)=ROA-DR/2.
      PHI(1) = RPHI
      DPHI(1) = 0.0
      DO 40 I = 2, N
      R(I) = R(I-1) + DR
      PHI(I) = RPHI
   40  DPHI(I) = 0.0
      GO TO 60
   30 IF (NTYPE .NE. 3) GO TO 50
C     THE SHELL IS A SPHERICAL CAP WHEN NTYPE IS 3
      SOA = ROA*RPHI
      DS = SOA/ AN
      DFI=RPHI/AN
      AOR=1./ROA
      DPHI(1)=AOR
      PHI(1) = -DFI/2.
      R(1) = ROA*SIN(PHI(1))
      DO 70 I=2,N
      PHI(I) = PHI(I -1) + DFI
      R(I) = ROA * SIN(PHI(I))
   70 DPHI(I)=AOR
      GO TO 60
   50 IF (NTYPE .NE. 4) GO TO 60
   60 RETURN
      END OF SUBROUTINE GEOMTY
```

APPENDIX F

```
      SUBROUTINE BOCON(LBCL, LBCR, X, ITER)
C     BOCON DEFINES THE BOUNDARY CONDITIONS TO BE USED.
      COMMON/BL1/ RCA, N, PHIO, SOA, DS, DR
      COMMON/BL2 /OMEGAL(3,3), ALAMDAL(3,3), OMEGAR(3,3), ALAMDAR(3,3),
     1 SL(3), SR(3)
      COMMON/BL5/A(202,3,3),B(202,3,3),C(202,3,3),D(202,3,3),E(202,3,3),
     1  S(202,3)
      COMMON /BL10/ R(101),PHI(101),DPHI(101)
      DIMENSION X(202,3)
      DATA PI / 3.141592653589793 /
      L=2*N
      L1= L-1
      DO 1  J=1,3
      DO 2  K=1,3
      A(1,J,K)=0.
      B(1,J,K)=0.
      C(1,J,K)=0.
      D(1,J,K)=0.
      E(1,J,K)=0.
      A(2*N,J,K)=0.
      B(2*N,J,K)=0.
      C(2*N,J,K)=0.
      D(2*N,J,K)=0.
    2 E(2*N,J,K)=0.
      S(1,J)=SL(J)
    1 S(2*N,J)=SR(J)
      IF (LBCL .NE. 1) GO TO 3
      IF (ITER .EQ. 1) PRINT 4
    4 FORMAT(40X, *THE LEFT BOUNDARY CONDITION IS A POLE POINT*)
      D(1,1,1)=1.
      C(1,2,2)=1.
      D(1,3,3)=1.
      GO TO 9
    3 IF (LBCL .NE. 2) GO TO 5
      IF (ITER .EQ. 1) PRINT 6
    6 FORMAT(40X, *THE LEFT BOUNDARY CONDITION IS PINNED*)
      D(1,1,1)=1.
      D(1,2,2)=1.
      C(1,3,3)=1.
      GO TO 9
    5 IF (LBCL .NE. 3) GO TO 7
      IF (ITER .EQ. 1) PRINT 8
```

APPENDIX F

```
  8 FORMAT(40X, *THE LEFT BOUNDARY CCNCITICN IS FIXED*)
    C(1,1,1)=1.
    D(1,2,2)=1.
    C(1,3,3)=1.
    GO TO 9
  7 IF (LBCL .NE. 4) GO TO 19
    IF (ITER .EQ. 1) PRINT 10
 10 FORMAT(40X, *THE LEFT BOUNDARY CONDITION IS FREE*)
    C(1,1,1)=1.
    C(1,2,2)=1.
    C(1,3,3)=1.
    S(1,2)= S(1,2)-X(1,1)*X(2,3)
    GO TO 9
 19 IF (LBCL .NE. 5) GO TO 9
    IF (ITER .EQ. 1) PRINT 20
 20 FORMAT(40X, *THE LEFT BOUNDARY IS AN ELASTIC CONSTRAINT*)
     DO 21 M=1,3
     DO 21 MM=1,3
     C(1,M,MM)=OMEGAL(M,MM)
 21  D(1,M,MM)=ALAMDAL(M,MM)
  9 IF (LBCR .NE. 1) GO TO 11
    IF (ITER .EQ. 1) PRINT 12
 12 FORMAT(40X, *THE RIGHT BOUNDARY CONDITION IS A POLE POINT*)
    C(2*N,1,1)=1.
    P(2*N,2,2)=1.
    C(2*N,3,3)=1.
    GO TO 17
 11 IF (LBCR .NE. 2) GO TO 13
    IF (ITER .EQ. 1) PRINT 14
 14 FORMAT(40X, *THE RIGHT BOUNDARY CONDITION IS PINNED*)
    C(2*N,1,1)=1.
    C(2*N,2,2)=1.
    P(2*N,3,3)=1.
    GO TO 17
 13 IF (LBCR .NE. 3) GO TO 15
    IF (ITER .EQ. 1) PRINT 16
 16 FORMAT(40X, *THE RIGHT BOUNDARY CONDITION IS FIXED*)
    C(2*N,1,1)=1.
    C(2*N,2,2)=1.
    C(2*N,3,3)=1.
    GO TO 17
 15 IF (LBCR .NE. 4) GO TO 22
    IF (ITER .EQ. 1) PRINT 18
```

```
18 FORMAT(40X, *THE RIGHT BOUNDARY CONDITION IS FREE*)
   B(2*N,1,1)=1.
   B(2*N,2,2)=1.
   B(2*N,3,3)=1.
   S(L,2)=S(L,2)-X(L1,1)*X(L,3)
   GO TO 17
22 IF (LBCR .NE. 5) GO TO 17
   IF (ITER .EQ. 1) PRINT 23
23 FORMAT(40X, *THE RIGHT BOUNDARY IS AN ELASTIC CONSTRAINT*)
    DO 24 M=1,3
    DO 24 MM=1,3
    B(L,M,MM)=OMEGAR(M,MM)
24  C(L,M,MM)=ALAMDAR(M,MM)
17 RETURN
   END OF BOCON
```

APPENDIX F

```
      SUBROUTINE FUNCT(X, N)
C     THIS SUBROUTINE CALCULATES THE NEGATIVE OF S=AX+BX+CX+DX+EX-S
      COMMON/BL5/A(202,3,3),B(202,3,3),C(202,3,3),D(202,3,3),E(202,3,3),
     1  S(202,3)
      DIMENSION AA(3,3), BB(3,3), CC(3,3), DD(3,3), EE(3,3)
      DIMENSION XA(3), XB(3), XC(3), XD(3), XE(3)
      DIMENSION X(202,3)
      L=2*N
      L1=L-1
      DO 1 J=1,L
       IF (J .NE. 1) GO TO 2
      DO 3 M=1,3
      XA(M)=0.
      XB(M)=0.
      XC(M)=X(1,M)
      XD(M)=X(2,M)
    3 XE(M)=X(3,M)
      GO TO 10
    2 IF (J .NE. 2) GO TO 4
      DO 5 M=1,3
      XA(M)=0.
      XB(M)=X(1,M)
      XC(M)= X(2,M)
      XD(M)=X(3,M)
    5 XE(M)=X(4,M)
      GO TO 10
    4 IF (J .NE. L) GO TO 6
      DO 7 M=1,3
      XA(M)=X(L-2,M)
      XB(M)=X(L-1,M)
      XC(M)=X(L,M)
      XD(M)=0.
    7 XE(M)=0.
      GO TO 10
    6 IF (J .NE. (L-1)) GO TO 8
      DO 9 M=1,3
      XA(M)= X(L-3,M)
      XB(M)=X(L-2,M)
      XC(M)=X(L-1,M)
      XD(M)=X(L,M)
    9 XE(M)=0.
      GO TO 10
```

```
 8 DO 11 M=1,3
   XA(M)= X(J-2,M)
   XB(M)= X(J-1,M)
   XC(M)=X(J,M)
   XD(M)=X(J+1,M)
11 XE(M)=X(J+2,M)
10 DO 12 M=1,3
   DO 12 MM=1,3
   AA(M,MM)=A(J,M,MM)
   BB(M,MM)=B(J,M,MM)
   CC(M,MM)=C(J,M,MM)
   DD(M,MM)=D(J,M,MM)
12 EE(M,MM)=E(J,M,MM)
   CALL MATRIX(20, 3, 3, 1, AA, 3, XA, 3, XA, 3)
   CALL MATRIX(20, 3, 3, 1, BB, 3, XB, 3, XB, 3)
   CALL MATRIX(20, 3, 3, 1, CC, 3, XC, 3, XC, 3)
   CALL MATRIX(20, 3, 3, 1, DD, 3, XD, 3, XD, 3)
   CALL MATRIX(20, 3, 3, 1, EE, 3, XE, 3, XE, 3)
   DO 13 M=1,3
13 S(J,M)= S(J,M) -XA(M)- XB(M)-XC(M)- XD(M)-XE(M)
 1 CONTINUE
   RETURN
   END OF SUBROUTINE FUNCT
```

APPENDIX F

```
      SUBROUTINE ANSWERS(NTYPE, X, LBCL, LBCR)
C     THIS SUBROUTINE CALCULATES AND PRINTS N-THETA AND M-THETA.
C     THE LOAD VECTORS ARE ALSO PRINTED WITH THESE VALUES.
      COMMON/BL1/ ROA, N, PHIO, SOA, DS, DR
      COMMON/BL6/ C11, C12, C22, K11, K12, K22, D11, D12, D22
      COMMON/BL8/ LAM, E10, E20, NU12, NU21, ETA
      COMMON /BL10/ R(101),PHI(101),DPHI(101)
       COMMON/BL11/ALFA1, ALFA2, T1, T2, ITEMP
      COMMON /BL14/ DS2, DEL, RPHI, AN
      DIMENSION X(202,3)
      REAL LAM, K11, K12, K22
      REAL N12, N3, M12, M3, KB12, K3
      REAL NU12, NU21
      CALL GEOMTY(NTYPE)
      L=2*N
    1 FORMAT(/5X*STATION*7X*NTHETA*14X*MTHETA*14X *RADIAL LOAD*7X*MERIDI
     1AN LOAD*)
       PRINT 1
       DO 2 I=1,N
      CALL STIF(I)
      ETAL= ETA/LAM
      ETANUS= ETA/(1.-NU21*NU12)
      TN1=E10*ETANUS*(ALFA1+NU12*ALFA2)* TDZ(I)
      TN2=E20*ETANUS*(ALFA2+NU21*ALFA1)* TDZ(I)
      TM1=E10*ETAL   *(ALFA1+NU12*ALFA2)*TZDZ(I)
      TM2=E20*ETAL   *(ALFA2+NU21*ALFA1)*TZDZ(I)
      UI=X(2*I,1)
      WI=X(2*I,2)
      BI=X(2*I,3)
      IF(I.NE.1)  GO TO 10
      FI=PHI(1)+DEL
      DFI= DPHI(1)
      RI= (R(1)+R(2))/2.
      GO TO 11
   10 FI=PHI(I)+DEL
      DFI= DPHI(I)
      RI= R(I)+DS2*COS(PHI(I))
   11 CONTINUE
      IF (I .NE. 1) GO TO 3
      IF (LBCL .NE. 1) GO TO 12
      UP= (-3.*X(2,1)+4.*X(4,1)-X(6,1))/(2.*DS)
      BP= (-3.*X(2,3)+4.*X(4,3)-X(6,3))/(2.*DS)
```

```
      GO TO 15
    3 IF (I .NE. N) GC TO 12
      IF (LBCR .NE. 1) GO TO 12
      LP= (X(L-4,1)-4.*X(L-2,1)+3.*X(L,1))/(2.*DS)
      BP= (X(L-4,3)-4.*X(L-2,3)+3.*X(L,3))/(2.*DS)
   15 AK22=-BP
      EM22=UP+DFI*WI+BI**2/(2.*ETA)
      GO TO 13
   12 SNR= SIN(FI)/RI
      CSR=COS(FI)/RI
      EM22= CSR*UI+SNR    *WI
      AK22= -CSR*BI
   13 CDK = (C11*D11-K11**2)/LAM**2
      GL2= CDK*LAM**2
      CL4= GL2*LAM**2
      N12= (C12*D11-K11*K12)/GL2
      M12= (K12*C11-K11*C12)/CDK
      E12= (2.*C12*K11*K12-C12*C12*D11-C11*K12*K12)/GL2
      KB12= (C12*K11*D12-C12*D11*K12+K12*K12*K11-K12*C11*D12)/GL2
      N3=(K12*D11-K11*D12)/GL4
      M3= (D12*C11-K11*K12)/GL2
      E3=(K11*K12*K12- C12*D11*K12 +C12*D12*K11-C11*D12*K12)/GL4
      K3= (2.*D12*K11*K12-D11*K12*K12-C11*D12*D12)/GL4
      AN22=N12*X(2*I-1,1)+M12*X(2*I-1,3)+(E12+C22)*EM22+(KB12+K22)*AK22
     2 +N12*TN1+ M12*TM1-TN2
      AM22=N3 *X(2*I-1,1)+M3 *X(2*I-1,3)+(E3+K22/LAM**2)*EM22
     1   +(K3+D22/LAM**2)*AK22
     2 +N3*TN1+M3*TM1-TM2
      PPL=PL(I,N)
      PPS=PS(I,N)
      PRINT 6, I, AN22, AM22, PPL, PPS
    6 FORMAT(4X,I5,4(5X, F15.8))
    2 CONTINUE
      RETURN
      END OF ANSWERS SUBROUTINE
```

APPENDIX F

```
      SUBROUTINE STIF(I)
C     THIS SUBROUTINE DEFINES C11, C12, C22, K11, K12, K22, D11, D12
C     AND D22.
      COMMON/BL6/ C11, C12, C22, K11, K12, K22, D11, D12, D22
      COMMON/BL8/ LAM, E10, E20, NU12, NU21, ETA
      COMMON/BL12/KCUNT
      COMMON/BL15/ HO
      REAL MULC, MULD
      REAL LAM, NU12, NU21, K11, K12, K22
C   E10= E1/E0 AND E2C= E2/E0
      MULC= T(I)/(1.-NU12*NU21)
      MULD= LAM**2*T(I)**2*MULC/12.
      K11=0.
      K12=0.
      K22=0.
      C11=E10*MULC
      C12= NU12*E1C*MULC
      C22=E20*MULC
      D11= E10*MULD
      D12= NU12*E10*MULD
      D22= E20*MULD
      RETURN
      END OF SUBROUTINE STIF
```

APPENDIX F

```
      SUBROUTINE DYNAMIC(S, N, J, I, NCYCLE)
C     THIS SUBROUTINE SETS ALPHA, BETA, GAMMA AND DELTA FOR THE
C     BACKWARD TIME DERIVATIVES AND SAVES THE BACKWARD TIME STATION
C     SOLUTIONS OF U AND W.
      COMMON/BL3/PP, IACC, INIT
      COMMON/BL7/XOLD(202,3), EPS2, EPS
      COMMON/BL12/KOUNT
      COMMON/BL13/X1(101,2), X2(101,2), X3(101,2)
      DIMENSION S(202,3)
      IF (KOUNT .NE. INIT) GO TO 1
      BE=0.
      GA=0.
      DE=1.
       IF (IP(KOUNT) .EQ. 1) GO TO 4
      IF (NCYCLE .GT. 1) GO TO 4
      GO TO 4
    1 IF (KOUNT .NE. (INIT+1)) GO TO 2
      IF (NCYCLE .GT. 1) GO TO 10
      DO 6 M=1,2
    6 X1(I,M)=(XOLD(J+1,M)+ XOLD(J-1,M))/2.
   10 BE= -6./EPS2*T(I)
      GA= -6./EPS *T(I)
      DE=-2.
      GO TO 4
    2 IF (KOUNT .NE. (INIT+2)) GO TO 3
      IF (NCYCLE .GT. 1) GO TO 11
      DO 7 M=1,2
      X2(I,M)=X1(I,M)
    7 X1(I,M)=(XOLD(J+1,M)+ XOLD(J-1,M))/2.
   11 BE= -4./EPS2*T(I)
      GA=  2./EPS2*T(I)
      DE=-1.
      GO TO 4
    3 IF (NCYCLE .GT. 1) GO TO 12
      DO 8 M=1,2
      X3(I,M)= X2(I,M)
      X2(I,M)=X1(I,M)
    8 X1(I,M)=(XOLD(J+1,M)+ XOLD(J-1,M))/2.
   12 BE= -5./EPS2*T(I)
      GA=  4./EPS2*T(I)
      DE= -1./EPS2*T(I)
    4 DO 9 M=1,2
    9 S(J,M)=S(J,M)+BE*X1(I,M)+GA*X2(I,M)+X3(I,M)*DE
      RETURN
      END OF DYNAMIC
```

APPENDIX F

```
      SUBROUTINE INITIAL(X, LBCL)
C THIS SUBROUTINE INITIALIZES THE X AND X DOT VECTORS
C     THIS SUBROUTINE IS REQUIRED IF X OR XDOT (INITIAL CONDITIONS) ARE
C     OTHER THAN ZERO FOR DYNAMIC PROBLEMS.
      COMMON/BL1/ ROA, N, PHIO, SOA, DS, DR
      COMMON/BL6/ C11, C12, C22, K11, K12, K22, D11, D12, D22
      COMMON/BL8/ LAM, E10, E20, NU12, NU21, ETA
      COMMON/BL9/ IDYM, KMAX, NONLIN, CHAR
      COMMON /BL10/ R(101),PHI(101),DPHI(101)
      COMMON/BL13/X1(101,2), X2(101,2), X3(101,2)
      DIMENSION X(202,3)
      REAL LAM, K11, K12, K22
      DO 16 I=1,N
      J=2*I
      X(J,1)=DV(1,I)
      X(J,2)=DV(2,I)
      X2(I,1)=DV(3,I)
   16 X2(I,2)=DV(4,I)
      DO 1 I=1,N
      J=2*I
      IF (I .NE. 1) GO TO 3
      WP=(-3.*X(2,2)+4.*X(4,2)-X(6,2))/(2.*DS)
      GO TO 5
    3 IF (I .NE. N) GO TO 4
      WP=(X(J-4,2)-4.*X(J-2,2)+3.*X(J,2))/(2.*DS)
      DFI=DPHI(N)+(DPHI(N)-DPHI(N-1))/2.
      GO TO 1
    4 WP=(X(J+2,2)-X(J-2,2))/(2.*DS)
    5 DFI=(DPHI(I)+DPHI(I+1))/2.
    1 X(J,3)=WP-DFI*X(J,1)
      DO 2 I=1,N
      J=2*I
      CALL STIF(I, LAM, E10, E10, NU12, NU21)
      IF (I .NE. 1) GO TO 6
      UP=(-3.*X(2,1)+4.*X(4,1)-X(6,1))/(2.*DS)
      PP=(-3.*X(2,3)+4.*X(4,3)-X(6,3))/(2.*DS)
      GO TO 8
    6 IF (I .NE. N) GO TO 7
      UP=(X(J-4,1)-4.*X(J-2,1)+3.*X(J,1))/(2.*DS)
      PP=(X(J-4,3)-4.*X(J-2,3)+3.*X(J,3))/(2.*DS)
      RI=R(N)+COS(PHI(N))*DS/2.
      FI=PHI(N)+(PHI(N)-PHI(N-1))/2.
```

APPENDIX F

```
      DFI=DPHI(N)+(DPHI(N)-DPHI(N-1))/2.
      GO TO 9
    7 UP=(X(J+2,1)-X(J-2,1))/(2.*DS)
      BP=(X(J+2,3)-X(J-2,3))/(2.*DS)
    8 RI=(R(I)+R(I+1))/2.
      FI=(PHI(I)+PHI(I+1))/2.
      DFI=(DPHI(I)+DPHI(I+1))/2.
    9 IF (LBCL .NE. 1) GO TO 10
      IF (I .EQ. 1) GO TO 11
   10 SNR=SIN(FI)/RI
      CSR=COS(FI)/RI
      EM22=CSR*X(J,1)+SNR*X(J,2)
      AK22=-CSR*X(J,3)
   11 EM11=UP+DFI*X(J,2)
       IF (NONLIN .EQ. 1) EM11= EM11+X(J,3)**2/(2.*ETA)
      AK11=-BP
      IF (LBCL .NE. 1) GO TO 12
      IF (I .NE. 1) GO TO 12
       EM22 =EM11
      AK22=AK11
   12 M=J-1
      X(M,1)=C11*EM11+C12*EM22+K11*AK11+K12*AK22
      X(M,3)=(K11*EM11+K12*EM22+D11*AK11+D12*AK22)/LAM**2
      IF (LBCL .NE. 1) GO TO 17
      IF (I .EQ. 1) X(M,3)=0.
      GO TO 2
   17  X(M,2)=CSR*(X(M,2)-(K12*EM11+K22*EM22+D12*AK11+D22*AK22))
    2 CONTINUE
      DO 13 I=1,N
      J=2*I-1
      IF (I .NE. 1) GO TO 14
      X(J,2)=X(J,2)+LAM**2*(-3.*X(2,3)+4.*X(4,3)-X(6,3))/(2.*DS)
      GO TO 13
   14 IF (I .NE. N) GO TO 15
      X(J,2)=X(J,2)+LAM**2*(3.*X(J,3)-4.*X(J-2,3)+X(J-4,3))/(2.*DS)
      GO TO 13
   15 X(J,2)= X(J,2)+LAM**2*(X(J+2,3)-X(J-2,3))/(2.*DS)
   13 CONTINUE
      RETURN
       END OF INITIAL
```

APPENDIX F

```
      FUNCTION PL(I)
      COMMON/BL1/ ROA, N, PHIO, SOA, DS, DR
      COMMON/BL3/PP, IACC, INIT
      COMMON/BL8/ LAM, E10, E20, NU12, NU21, ETA
      COMMON/BL12/KOUNT
      REAL LAM
      REAL NU12, NU21
      PCL   = 2.*LAM*ETA/(3.*(1.-NU12*NU21))**.5*(T(I)/ROA)**2*E10
      PLL=PP*PCL
      PL=PLL
      IF (KOUNT .EQ. 0 ) PL=0.
      IF (KOUNT .GT.20 ) PL=0.
      IF (IACC .NE. 1 ) GO TO 1
      IF (KOUNT .EQ. 0) PL= PLL
      IF (KOUNT .EQ.20) PL= 0.
    1 CONTINUE
      RETURN
      END
```

```
      FUNCTION PS(I)
      COMMON/BL1/ ROA, N, PHIO, SOA, DS, DR
      COMMON/BL3/PP, IACC, INIT
      COMMON/BL4/PPS
      COMMON/BL8/ LAM, E10, E20, NU12, NU21, ETA
      COMMON/BL12/KOUNT
      PS=PPS
      RETURN
      END
```

```
      FUNCTION IP(KOUNT)
      IP=0
      IF (KOUNT .EQ.  0) IP=1
      IF (KOUNT .EQ. 20) IP=1
      RETURN
      END
```

APPENDIX F

```
      FUNCTION DV(M,I)
C     DV(1,I) SETS THE INITIAL U DISPLACEMENT
C     DV(2,I) SETS THE INITIAL W DISPLACEMENT
C     DV(3,I) SETS THE INITIAL U VELOCITY
C     DV(4,I) SETS THE INITIAL W VELOCITY
      DV=0.
       RETURN
      END OF DV
```

```
      FUNCTION T(I)
      COMMON/BL15/ HO
C     T(I) MUST BE NONDIMENSIONAL THICKNESS, THEREFORE DIVIDE BY HO
      T=1./HO
      RETURN
      END
```

```
      FUNCTION TDZ(I)
      COMMON/BL1/ ROA, N, PHIO, SOA, DS, DR
       COMMON/BL11/ALFA1, ALFA2, T1, T2, ITEMP
      IT=ITEMP+1
      AN=ITEMP+1
       H=T(I)/2.
      TDZ=T1 *T(I)+T2*(H**IT-(-H)**IT)/AN
      RETURN
      END
```

```
      FUNCTION TZDZ(I)
      COMMON/BL1/ ROA, N, PHIO, SOA, DS, DR
       COMMON/BL11/ALFA1, ALFA2, T1, T2, ITEMP
      IT=ITEMP+2
      AN=ITEMP+2
       H=T(I)/2.
      TZDZ=T2*(H**IT-(-H)**IT)/AN
      RETURN
      END
```

APPENDIX G

SAMPLE PRINTOUT

The printout for the sample problem is as follows:

```
$GIVEN

NTYPE   =  3,
N       =  26,
RO      =  0.1E+03,
SO      =  0.0,
HO      =  0.1E+01,
EO      =  0.1E+08,
E1      =  0.1E+08,
E2      =  0.1E+08,
NU12    =  0.3E+00,
NU21    =  0.3E+00,
SIGO    =  0.1E+01,
NONLIN  =  1,
CONV    =  0.1E-02,
PHIO    =  0.158094E+02,
LBCL    =  1,
LBCR    =  3,
PP      = -0.6E+00,
PPS     =  0.0,
SL      =  0.0,  0.0,  0.0,
SR      =  0.0,  0.0,  0.0,
CHAR    =  0.1E+03,
IDYM    =  1,
KMAX    =  40,
DTAU    =  0.25E+00,
```

```
T1      =  0.0,

T2      =  0.0,

ALFA1   =  0.0,

ALFA2   =  0.0,

ITEMP   =  0,

IFREQ   =  4,

ISTART  =  0,

$END
SAMPLE PROBLEM FOR A CLAMPED SPHERICAL CAP WITH LAMBDA - 5
```

```
IF IDYM=0 THE SHELL IS LOADED STATICALLY
IF IDYM=1THE SHELL IS LOADED DYNAMICALLY
IF IDYM=2 THE SHELL STATIC BUCKLING LOAD IS CALCULATED

FOR THIS RUN, IDYM=                      1

NUMBER OF STATIONS =                    26

MERIDIAN/REFERENCE RATIO =                .275926

RADIUS/REFERENCE RATIO =                 1.000000

THICK/ REF RAD RATIO =                    .010000

E1/E0 =         1.000000                   E2/E0 =            1.000000

NU12 =           .300000                   NU21 =              .300000

E0/S0 =   10000000.000000                  REF DIST =       100.000000

              IF NONLIN = 0 ONLY LINEAR TERMS ARE USED
              IF NONLIN = 1 NONLINEAR TERMS USED

              FOR THIS RUN NONLIN =      1
```

APPENDIX G

THE GEOMETRY OF THE SHELL FOLLOWS

STATION NO	MERIDIAN DIS	RADIAL DIS	ANGLE(RAD)	CURVATURE(DFI/DS)
1	0.	-5.602033E-08	-2.775558E-17	1.000000E+00
2	1.103704E-02	1.103693E-02	1.103704E-02	1.000000E+00
3	2.207409E-02	2.207257E-02	2.207409E-02	1.000000E+00
4	3.311113E-02	3.310553E-02	3.311113E-02	1.000000E+00
5	4.414817E-02	4.413445E-02	4.414817E-02	1.000000E+00
6	5.518522E-02	5.515799E-02	5.518522E-02	1.000000E+00
7	6.622226E-02	6.617482E-02	6.622226E-02	1.000000E+00
8	7.725930E-02	7.718359E-02	7.725930E-02	1.000000E+00
9	8.829635E-02	8.818295E-02	8.829635E-02	1.000000E+00
10	9.933339E-02	9.917157E-02	9.933339E-02	1.000000E+00
11	1.103704E-01	1.101481E-01	1.103704E-01	1.000000E+00
12	1.214075E-01	1.211112E-01	1.214075E-01	1.000000E+00
13	1.324445E-01	1.320596E-01	1.324445E-01	1.000000E+00
14	1.434816E-01	1.429919E-01	1.434816E-01	1.000000E+00
15	1.545186E-01	1.539067E-01	1.545186E-01	1.000000E+00
16	1.655556E-01	1.648029E-01	1.655556E-01	1.000000E+00
17	1.765927E-01	1.756789E-01	1.765927E-01	1.000000E+00
18	1.876297E-01	1.865335E-01	1.876297E-01	1.000000E+00
19	1.986668E-01	1.973655E-01	1.986668E-01	1.000000E+00
20	2.097038E-01	2.081733E-01	2.097038E-01	1.000000E+00
21	2.207409E-01	2.189559E-01	2.207409E-01	1.000000E+00
22	2.317779E-01	2.297117E-01	2.317779E-01	1.000000E+00
23	2.428150E-01	2.404396E-01	2.428150E-01	1.000000E+00
24	2.538520E-01	2.511381E-01	2.538520E-01	1.000000E+00
25	2.648890E-01	2.618061E-01	2.648890E-01	1.000000E+00
26	2.759261E-01	2.724422E-01	2.759261E-01	1.000000E+00

THE MAXIMUM SHELL RISE IS .03782669

THE SHELL THICKNESS = 1.00000000

THE LEFT BOUNDARY CONDITION IS A POLE POINT

THE RIGHT BOUNDARY CONDITION IS FIXED

THE LEFT OMEGA MATRIX / THE LEFT LAMBDA MATRIX

0.	0.	0.	(N)	+	1.00000E+00	0.	0.	(U)	=	0.
0.	1.00000E+00	0.	(Q)	+	0.	0.	0.	(W)	=	0.
0.	0.	0.	(M)	+	0.	0.	1.00000E+00	(B)	=	0.

THE RIGHT OMEGA MATRIX / THE RIGHT LAMBDA MATRIX

0.	0.	0.	(N)	+	1.00000E+00	0.	0.	(U)	=	0.
0.	0.	0.	(Q)	+	0.	1.00000E+00	0.	(W)	=	0.
0.	0.	0.	(M)	+	0.	0.	1.00000E+00	(B)	=	0.

STATION NO	N-S RESULTANT	SHEAR FORCE	M-S RESULTANT	U-DEFORMATION	W-DEFORMATION	BETA ROTATION
1	0.	0.	0.	0.	0.	0.
2	0.	0.	0.	0.	0.	0.
3	0.	0.	0.	0.	0.	0.
4	0.	0.	0.	0.	0.	0.
5	0.	0.	0.	0.	0.	0.
6	0.	0.	0.	0.	0.	0.
7	0.	0.	0.	0.	0.	0.
8	0.	0.	0.	0.	0.	0.
9	0.	0.	0.	0.	0.	0.
10	0.	0.	0.	0.	0.	0.
11	0.	0.	0.	0.	0.	0.
12	0.	0.	0.	0.	0.	0.
13	0.	0.	0.	0.	0.	0.
14	0.	0.	0.	0.	0.	0.
15	0.	0.	0.	0.	0.	0.
16	0.	0.	0.	0.	0.	0.
17	0.	0.	0.	0.	0.	0.
18	0.	0.	0.	0.	0.	0.
19	0.	0.	0.	0.	0.	0.
20	0.	0.	0.	0.	0.	0.
21	0.	0.	0.	0.	0.	0.
22	0.	0.	0.	0.	0.	0.
23	0.	0.	0.	0.	0.	0.
24	0.	0.	0.	0.	0.	0.
25	0.	0.	0.	0.	0.	0.
26	0.	0.	0.	0.	0.	0.

STATION	NTHETA	MTHETA	RADIAL LOAD	MERIDIAN LOAD
1	0.	0.	0.	0.
2	0.	0.	0.	0.
3	0.	0.	0.	0.
4	0.	0.	0.	0.
5	0.	0.	0.	0.
6	0.	0.	0.	0.
7	0.	0.	0.	0.
8	0.	0.	0.	0.
9	0.	0.	0.	0.
10	0.	0.	0.	0.
11	0.	0.	0.	0.
12	0.	0.	0.	0.
13	0.	0.	0.	0.
14	0.	0.	0.	0.
15	0.	0.	0.	0.
16	0.	0.	0.	0.
17	0.	0.	0.	0.
18	0.	0.	0.	0.
19	0.	0.	0.	0.
20	0.	0.	0.	0.
21	0.	0.	0.	0.
22	0.	0.	0.	0.
23	0.	0.	0.	0.
24	0.	0.	0.	0.
25	0.	0.	0.	0.
26	0.	0.	0.	0.

STATION NO	U DOT	W DOT	U DOTDOT	W DOTDOT
1	0.	0.	0.	0.
2	0.	0.	0.	-7.26273039E+04
3	0.	0.	0.	-7.26273039E+04
4	0.	0.	0.	-7.26273039E+04
5	0.	0.	0.	-7.26273039E+04
6	0.	0.	0.	-7.26273039E+04
7	0.	0.	0.	-7.26273039E+04
8	0.	0.	0.	-7.26273039E+04
9	0.	0.	0.	-7.26273039E+04
10	0.	0.	0.	-7.26273039E+04
11	0.	0.	0.	-7.26273039E+04
12	0.	0.	0.	-7.26273039E+04
13	0.	0.	0.	-7.26273039E+04
14	0.	0.	0.	-7.26273039E+04
15	0.	0.	0.	-7.26273039E+04
16	0.	0.	0.	-7.26273039E+04
17	0.	0.	0.	-7.26273039E+04
18	0.	0.	0.	-7.26273039E+04
19	0.	0.	0.	-7.26273039E+04
20	0.	0.	0.	-7.26273039E+04
21	0.	0.	0.	-7.26273039E+04
22	0.	0.	0.	-7.26273039E+04
23	0.	0.	0.	-7.26273039E+04
24	0.	0.	0.	-7.26273039E+04
25	0.	0.	0.	-7.26273039E+04
26	0.	0.	0.	-7.26273039E+04

KOUNT	TAU	DEL	ITERATIONS	CYCLES	XNORM
0	0.00000	0.00000	1	0	0.
1	.25000	-.00955	4	3	5.13873E-08
2	.50000	-.03342	7	3	1.60563E-06
3	.75000	-.06606	10	3	5.68624E-06

STATION NO	N-S RESULTANT	SHEAR FORCE	M-S RESULTANT	U-DEFORMATION	W-DEFORMATION	BETA ROTATION
1	-2.9536080E+04	0.	2.2678369E+04	0.	-2.7213944E+04	0.
2	-2.9536507E+04	1.0823427E+02	2.8651020E+04	7.2302182E+01	-2.7232676E+04	-2.3794093E+03
3	-2.9547457E+04	2.0092919E+02	4.2857752E+04	1.4491813E+02	-2.7276970E+04	-5.8643852E+03
4	-2.9569003E+04	2.6284702E+02	6.3857895E+04	2.1825654E+02	-2.7369510E+04	-1.1267650E+04
5	-2.9604770E+04	2.7904913E+02	8.7943598E+04	2.9295358E+02	-2.7534351E+04	-1.9114122E+04
6	-2.9659037E+04	2.3625043E+02	1.1002992E+05	3.6990004E+02	-2.7798572E+04	-2.9427722E+04
7	-2.9735975E+04	1.2494226E+02	1.2393358E+05	4.5022388E+02	-2.8195972E+04	-4.1592358E+04
8	-2.9838686E+04	-5.7701113E+01	1.2291078E+05	5.3525564E+02	-2.8709495E+04	-5.4259585E+04
9	-2.9968121E+04	-3.0459378E+02	1.0050515E+05	6.2637881E+02	-2.9363079E+04	-6.5336673E+04
10	-3.0122001E+04	-5.9597947E+02	5.1696759E+04	7.2492612E+02	-3.0114017E+04	-7.2090635E+04
11	-3.0293913E+04	-6.9818565E+02	-2.5753228E+04	8.3189418E+02	-3.0897264E+04	-7.1396769E+04
12	-3.0472829E+04	-1.1649886E+03	-1.2997832E+05	9.4749537E+02	-3.1613341E+04	-6.0141339E+04
13	-3.0643271E+04	-1.3424182E+03	-2.5401796E+05	1.0704409E+03	-3.2131420E+04	-3.5756736E+04
14	-3.0786302E+04	-1.3771512E+03	-3.9552718E+05	1.1969946E+03	-3.2298726E+04	3.1722273E+03
15	-3.0881279E+04	-1.2275708E+03	-5.0750267E+05	1.3200325E+03	-3.1956460E+04	5.6331964E+04
16	-3.0908078E+04	-8.7532111E+02	-6.0017726E+05	1.4285638E+03	-3.0961237E+04	1.2126179E+05
17	-3.0849245E+04	-3.3416854E+02	-6.4395012E+05	1.5093152E+03	-2.9209544E+04	1.9322207E+05
18	-3.0691517E+04	3.4728171E+02	-6.2287211E+05	1.5436291E+03	-2.6661564E+04	2.6544018E+05
19	-3.0426504E+04	1.0915345E+03	-5.2789742E+05	1.5207502E+03	-2.3360093E+04	3.2975011E+05
20	-3.0050808E+04	1.8071110E+03	-3.5897597E+05	1.4314766E+03	-1.9440594E+04	3.7754024E+05
21	-2.9566361E+04	2.4095469E+03	-1.2521299E+05	1.2760124E+03	-1.5130233E+04	4.0082387E+05
22	-2.8981643E+04	2.8424471E+03	1.5719226E+05	1.0637499E+03	-1.0735558E+04	3.9318638E+05
23	-2.8313785E+04	3.0907408E+03	4.6877244E+05	8.1143043E+02	-6.6218535E+03	3.5037444E+05
24	-2.7590540E+04	3.1792366E+03	7.9114827E+05	5.3908605E+02	-3.1886670E+03	2.7039573E+05
25	-2.6850791E+04	3.1539024E+03	1.1101973E+06	2.6487187E+02	-8.4675365E+02	1.5317362E+05
26	-2.6143115E+04	3.0494846E+03	1.4164056E+06	0.	0.	0.

STATION	NTHETA	MTHETA	RADIAL LOAD	MERIDIAN LOAD
1	-2.95441172E+04	2.05951423E+04	-7.26273039E+04	0.
2	-2.95428186E+04	2.65597227E+04	-7.26273039E+04	0.
3	-2.95769316E+04	3.49924742E+04	-7.26273039E+04	0.
4	-2.96507017E+04	4.75047834E+04	-7.26273039E+04	0.
5	-2.97841122E+04	6.24386209E+04	-7.26273039E+04	0.
6	-2.99999064E+04	7.74010341E+04	-7.26273039E+04	0.
7	-3.03176440E+04	8.94421377E+04	-7.26273039E+04	0.
8	-3.07465333E+04	9.52812993E+04	-7.26273039E+04	0.
9	-3.12775841E+04	9.16545059E+04	-7.26273039E+04	0.
10	-3.18763917E+04	7.57877822E+04	-7.26273039E+04	0.
11	-3.24784323E+04	4.59610977E+04	-7.26273039E+04	0.
12	-3.29889603E+04	2.08351564E+03	-7.26273039E+04	0.
13	-3.32891747E+04	-5.33394903E+04	-7.26273039E+04	0.
14	-3.32490901E+04	-1.17487879E+05	-7.26273039E+04	0.
15	-3.27457223E+04	-1.82388537E+05	-7.26273039E+04	0.
16	-3.16833667E+04	-2.40531281E+05	-7.26273039E+04	0.
17	-3.00117691E+04	-2.83414555E+05	-7.26273039E+04	0.
18	-2.77385172E+04	-3.03365017E+05	-7.26273039E+04	0.
19	-2.49339923E+04	-2.94860568E+05	-7.26273039E+04	0.
20	-2.17298197E+04	-2.55514043E+05	-7.26273039E+04	0.
21	-1.83136057E+04	-1.86413570E+05	-7.26273039E+04	0.
22	-1.49229158E+04	-9.15657540E+04	-7.26273039E+04	0.
23	-1.18401095E+04	2.27586554E+04	-7.26273039E+04	0.
24	-9.38800207E+03	1.50496473E+05	-7.26273039E+04	0.
25	-7.92555514E+03	2.86004291E+05	-7.26273039E+04	0.
26	-7.84293439E+03	4.24921677E+05	-7.26273039E+04	0.

KOUNT	TAU	DEL	ITERATIONS	CYCLES	XNORM
4	1.00000	-.10287	13	3	9.30392E-06
5	1.25000	-.13994	16	3	1.08932E-05
6	1.50000	-.17430	19	3	1.31620E-05
7	1.75000	-.20398	22	3	1.06050E-05

APPENDIX H

CONVERSION OF U.S. CUSTOMARY UNITS TO SI UNITS

The International System of Units (SI) was adopted by the Eleventh General Conference on Weights and Measures in 1960. (See ref. 18.) Conversion factors for the units used in this report are given in the following table:

Physical quantity	U.S. Customary Unit	Conversion factor (*)	SI Unit (**)
Length	in.	2.54×10^{-2}	meter (m)
Modulus of axial stress, elasticity	psi	6.895×10^{3}	newton/meter^2 (N/m^2)
Temperature	degree Fahrenheit	$K = (^{o}F + 459.67)/1.8$	kelvin (K)

*Multiply value given in U.S. Customary Unit by conversion factor to obtain equivalent value in SI unit.

**The prefix giga (G) is used to indicate 10^9 units.

REFERENCES

1. Schaeffer, Harry G.: Computer Program for Finite-Difference Solutions of Shells of Revolution Under Asymmetric Loads. NASA TN D-3926, 1967.

2. Budiansky, Bernard; and Radkowski, Peter P.: Numerical Analysis of Unsymmetrical Bending of Shells of Revolution. AIAA J., vol. 1, no. 8, Aug. 1963, pp. 1833-1842.

3. Ball, R. E.: A Geometrically Nonlinear Analysis of Arbitrarily Loaded Shells of Revolution. NASA CR-909, 1968.

4. Sanders, J. Lyell, Jr.: Nonlinear Theories for Thin Shells. Quart. Appl. Math., vol. XXI, no. 1, Apr. 1963, pp. 21-36.

5. Ambartsumyan, S. A.: Theory of Anisotropic Shells. NASA TT F-118, 1964.

6. Houbolt, John C.: A Recurrence Matrix Solution for the Dynamic Response of Aircraft in Gusts. NACA Rep. 1010, 1951. (Supersedes NACA TN 2060.)

7. Johnson, Donald E.; and Greif, Robert: Dynamic Response of a Cylindrical Shell: Two Numerical Methods. AIAA J., vol. 4, no. 3, Mar. 1966, pp. 486-494.

8. Sylvester, R. J.; and Meyer, F.: Two Point Boundary Problems by Quasilinearization. J. Soc. Ind. Appl. Math., vol. 13, no. 2, June 1965, pp. 586-602.

9. Potters, M. L.: A Matrix Method for the Solution of a Linear Second Order Difference Equation in Two Variables. MR 19, Math. Centrum (Amsterdam), 1955.

10. Anon.: Control Data 6400/6500/6600 Computer Systems FORTRAN Reference Manual. Publ. No. 60174900B, Control Data Corp., Nov. 1967.

11. Jones, Robert M.: Buckling of Circular Cylindrical Shells With Multiple Orthotropic Layers and Eccentric Stiffeners. AIAA J., vol. 6, no. 12, Dec. 1969, pp. 2301-2305.

12. Archer, Robert R.; and Lange, Charles G.: Nonlinear Dynamic Behavior of Shallow Spherical Shells. AIAA J., vol. 3, no. 12, Dec. 1965, pp. 2313-2317.

13. Budiansky, Bernard; and Roth, Robert S.: Axisymmetric Dynamic Buckling of Clamped Shallow Spherical Shells. Collected Papers on Instability of Shell Structures – 1962, NASA TN D-1510, 1962, pp. 597-606.

14. Stephens, Wendell B.; and Fulton, Robert E.: Axisymmetric Static and Dynamic Buckling of Spherical Caps Due to Centrally Distributed Pressures. AIAA J., vol. 7, no. 11, Nov. 1969, pp. 2120-2126.

15. Mendelson, Alexander: Plasticity: Theory and Application. Macmillan Co., c.1968, pp. 186-190.

16. Rossettos, J. N.; and Parisse, R. F.: The Dynamic Response of Cylindrical and Conical Panels. Trans. ASME, Ser. E: J. Appl. Mech., vol. 36, no. 2, June 1969, pp. 271-276.

17. Anon.: Control Data 6000 Series Computer Systems Matrix Algebra Subroutines Reference Manual. Publ. No. 60135200, Control Data Corp., June 1966.

18. Comm. on Metric Pract.: ASTM Metric Practice Guide. NBS Handbook 102, U.S. Dep. Com., Mar. 10, 1967.

TABLE 1.- SUBROUTINE DESCRIPTIONS

FORTRAN name	Description
GEØMTY	Calculates shell geometry, that is, r, ϕ, and ϕ'
BØCØN	Sets proper boundary conditions
ABCDES	Calculates coefficients of A, B, C, D, E, and q matrices
FUNCT	Calculates negative of $f_k\left(\bar{z}_i,\bar{z}_{i-1},s\right)$ in equation (27)
PØTTER	Calculates P, Q, and R matrices and then solves for the Newton-Raphson corrections δz
DYNAMIC	Calculates $\bar{\alpha}$, $\bar{\beta}$, $\bar{\gamma}$, and $\bar{\delta}$ and stores $X_{m,l-1}$, $X_{m,l-2}$, and $X_{m,l-3}$
ANSWERS	Calculates n_{22} and m_{22} and prints n_{22}, m_{22}, p, and p_s
STIF	Calculates constitutive coefficients in equations (14)
INITIAL	Initializes $y_{i,0}$ and $\dot{y}_{i,0}$ vectors when the initial conditions are other than zero
MATRIX	Perform basic matrix operations as detailed in reference 16

TABLE 2.- GLOSSARY OF FORTRAN VARIABLE NAMES

Variable	Program name	Description
a	CHAR	Reference length
C_{11}, C_{12}, C_{22}	C11, C12, C22	Constants in equations (10) and (11)
D_{11}, D_{12}, D_{22}	D11, D12, D22	Constants in equations (12) and (13)
E_o	EØ	Reference modulus of elasticity
E_1/E_o, E_2/E_1	E10, E20	Nondimensional moduli of elasticity
e_{11}, e_{22}	EM11, EM22	Membrane strains, used in ANSWERS
H_o	HØ	Reference shell thickness
h	T(I)	Nondimensional shell thickness
j	ITEMP	Temperature exponent in equation (17)
K_{11}, K_{12}, K_{22}	K11, K12, K22	Constants in equations (10) to (13)
m	KØUNT	Step in time
n	N	Number of stations
p, p_s	PL, PS	Nondimensional loads, see FUNCTION PL and FUNCTION PS
r_i	R(I)	Nondimensional radial distance at station i-1/2 defined in GEØMTY
s	SMER	Nondimensional meridional distance
x (1st element)	X(J,1)	n_{11} when j odd
x (2d element)	X(J,2)	q when j odd
x (3d element)	X(J,3)	m_{11} when j odd
y (1st element)	X(J,1)	u when j even
y (2d element)	X(J,2)	w when j even
y (3d element)	X(J,3)	β when j even
z	XOLD	Matrix of unknown
δz	X	Matrix of Newton-Raphson corrections to z
α_1, α_2	ALFA1, ALFA2	Constants in equations (16)
$\bar{\alpha}$, $\bar{\beta}$, $\bar{\gamma}$, $\bar{\delta}$	AL, BE, GA, DE	Constants in equation (21)

TABLE 2.- GLOSSARY OF FORTRAN VARIABLE NAMES – Concluded

Variable	Program name	Description
η	ETA	E_o/σ
κ_{11}, κ_{22}	AK11, AK22	Nondimensional principal curvatures
ν_{12}, ν_{21}	NU12, NU21	Poisson's ratios
λ	LAM	Ratio of H_o/a
λ_s	LAMS	Shell parameter
t_1^m, t_2^m	TM1, TM2	Thermal moment resultants
t_1^n, t_2^n	TN1, TN2	Thermal force resultants
$\Delta\tau$	ΔTAU	Nondimensional time increment
Δs	DS	Nondimensional meridional increment
ϕ_i	PHI(I)	Colatitude angle at i-1/2
ϕ_i'	DPHI(I)	Colatitude angle change at i-1/2

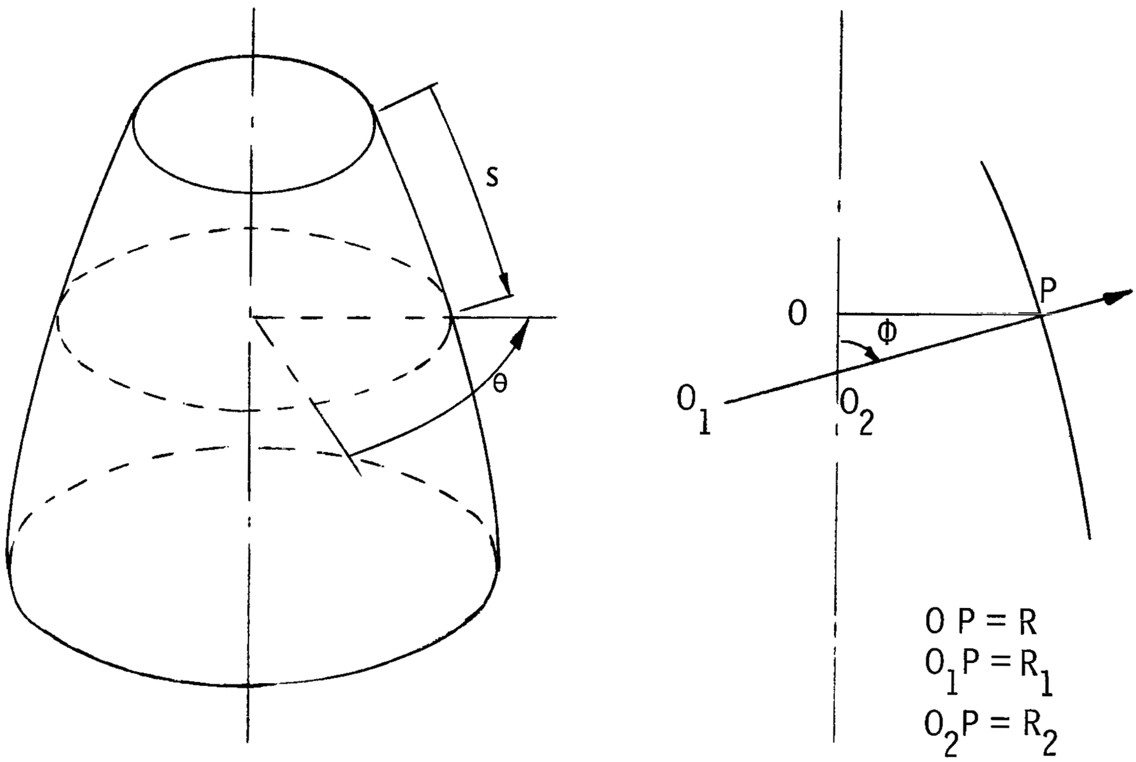
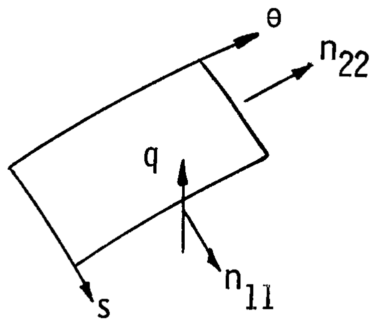
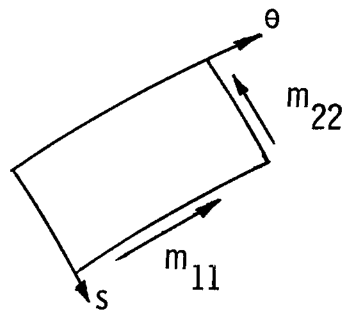


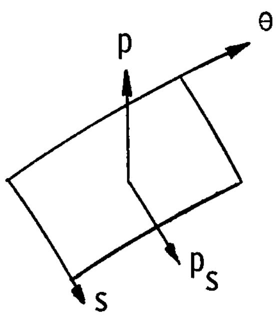
Figure 1.- Surface geometry and coordinates.



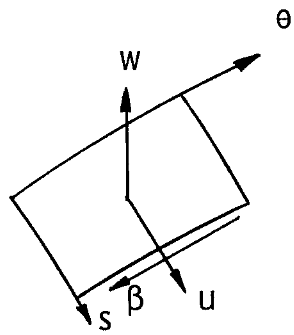
(a) Membrane and transverse force resultants.



(b) Moment resultants.

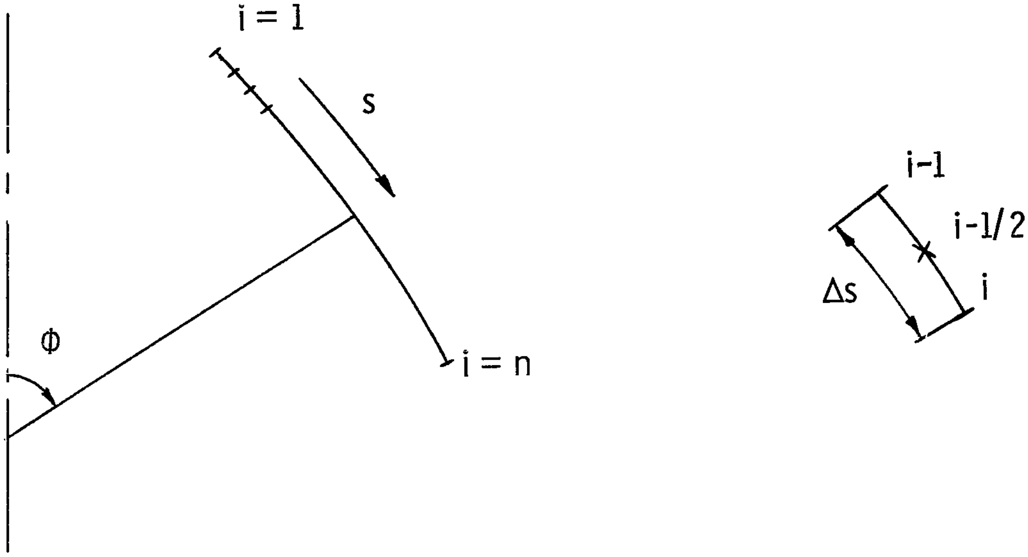


(c) Load per unit area.



(d) Displacements and rotation.

Figure 2.- Positive sense of forces, moments, loads, displacements, and rotations.



(a) Shell meridian divided into equal increments.

(b) Location of i-1/2 station.

Figure 3.- Locations of boundaries, stations, and midpoints of shell and shell meridian increments.

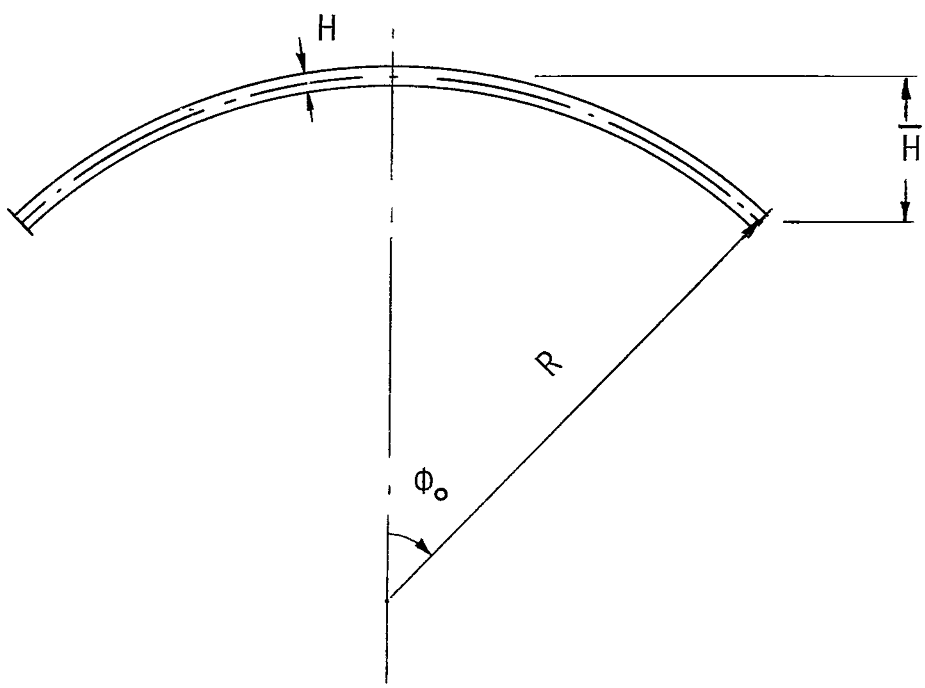


Figure 4.- Spherical cap geometry.

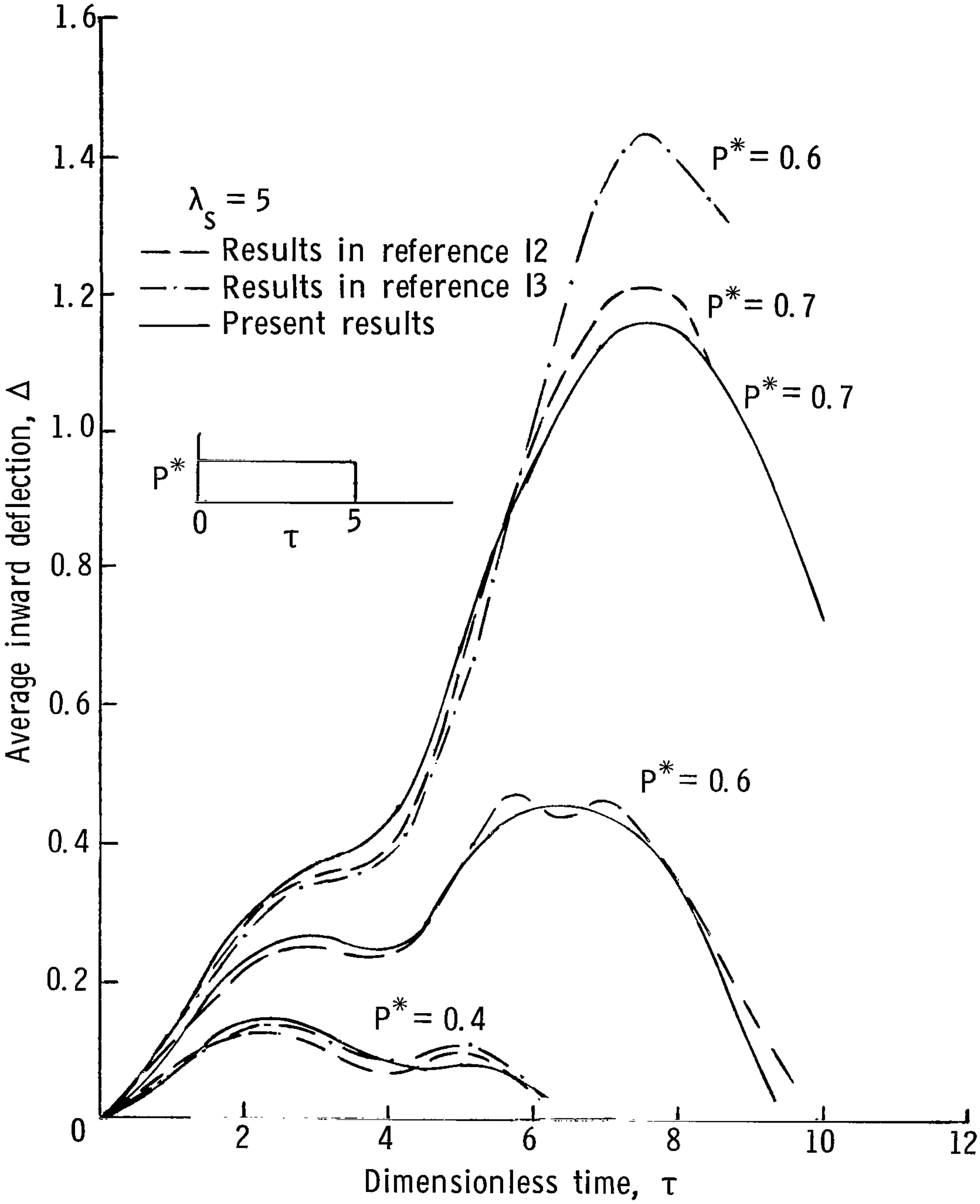


Figure 5.- Deflection response of clamped spherical cap for various uniformly distributed pressures. (Load duration is from $\tau = 0$ to $\tau = 5$.)

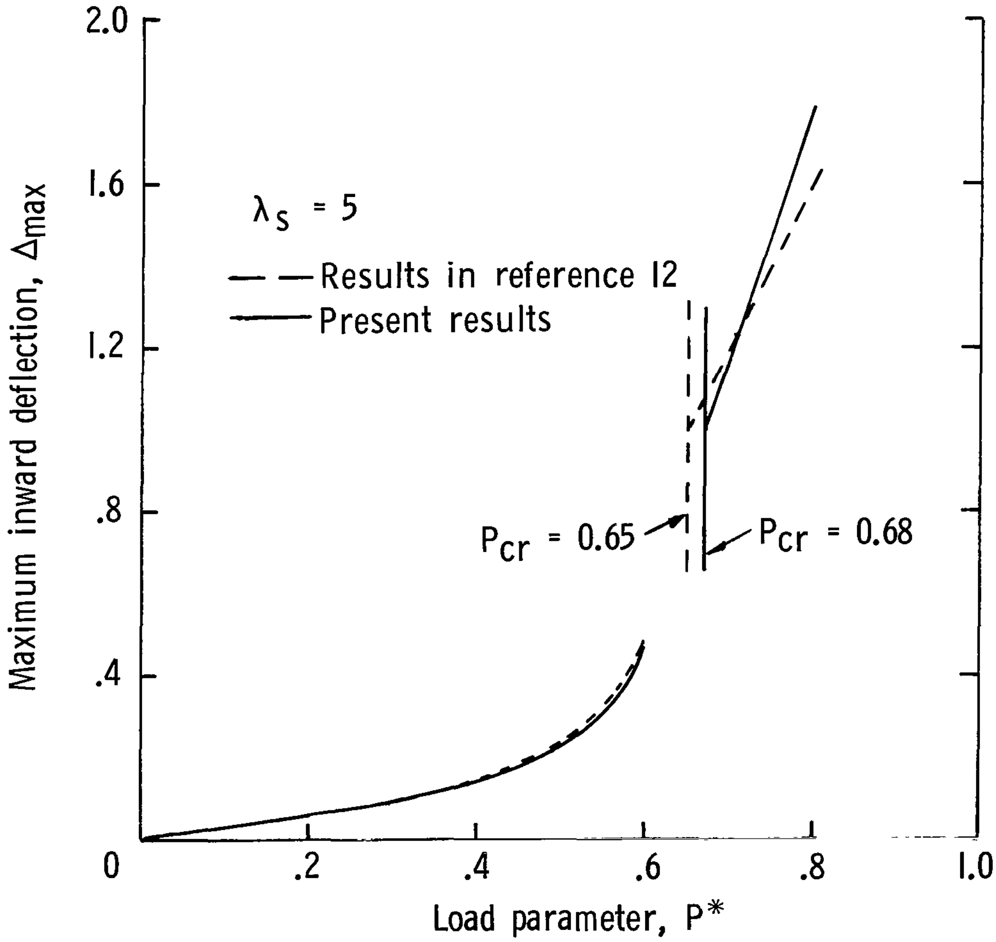
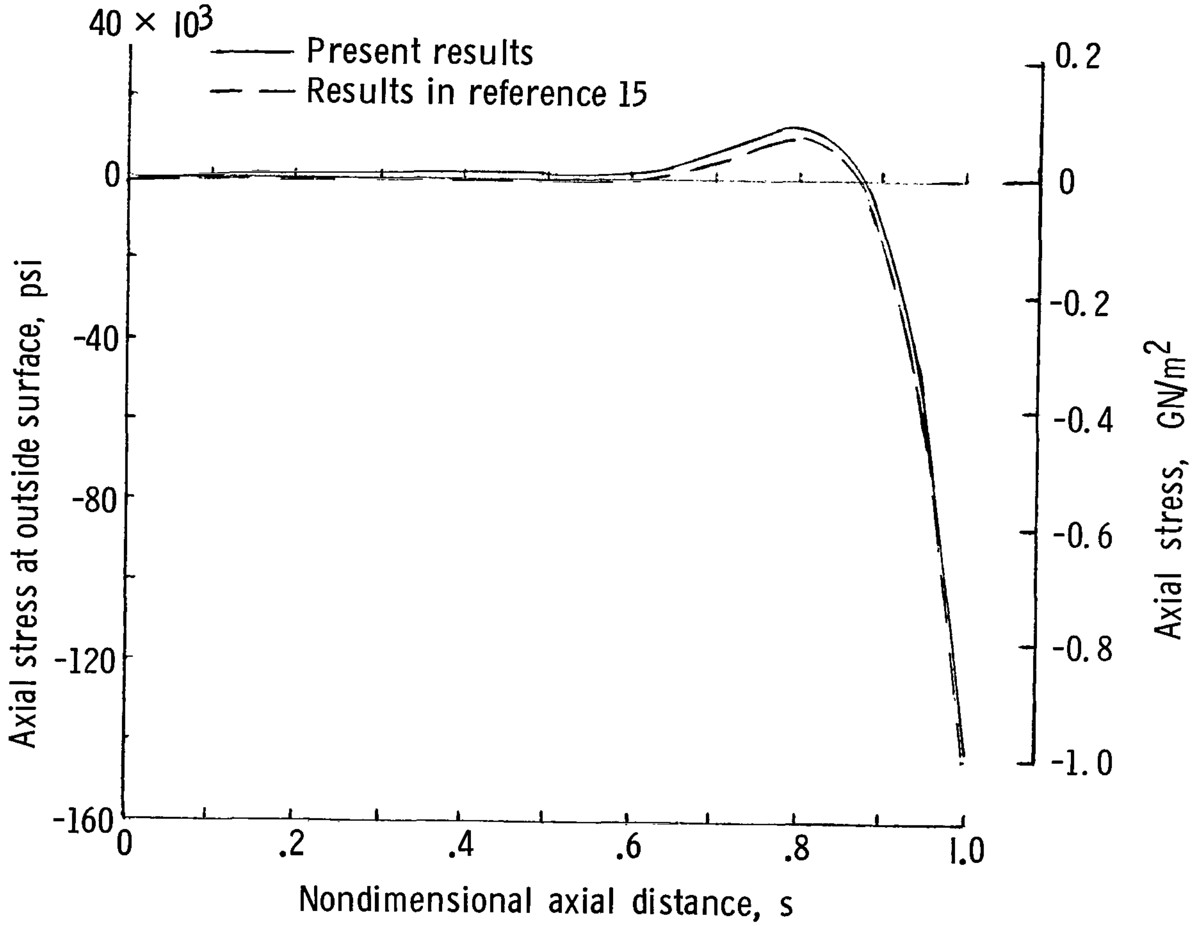
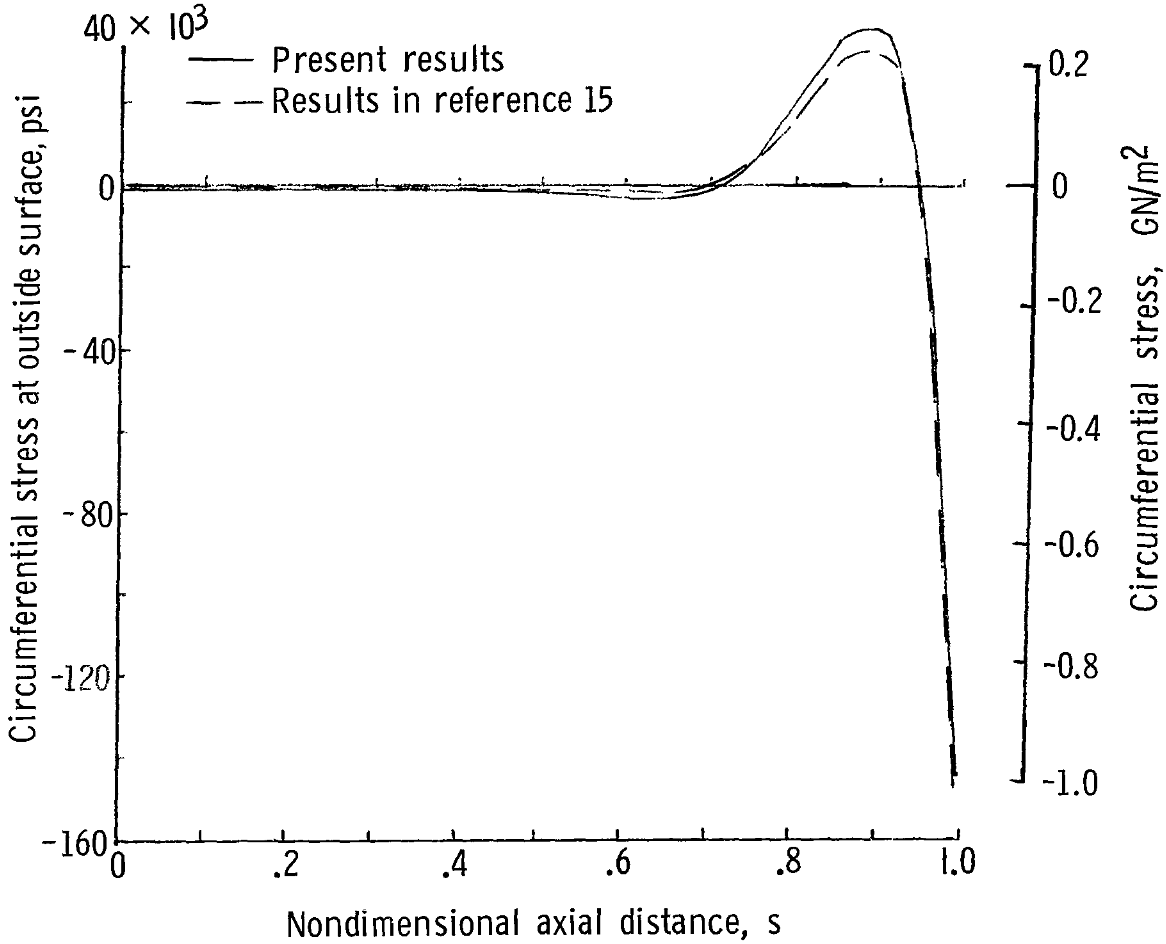


Figure 6.- Dynamic buckling of clamped spherical cap subjected to uniformly distributed step pulses. (Load duration is from $\tau = 0$ to $\tau = 5$.)



(a) Axial stress.

Figure 7.- Outer surface thermal stress variation along shell length.



(b) Circumferential stress.

Figure 7.- Concluded.

NASA-Langley, 1971 —— 32 L-6299

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
WASHINGTON, D. C. 20546

OFFICIAL BUSINESS

FIRST CLASS MAIL

POSTAGE AND FEES PAID
NATIONAL AERONAUTICS AND
SPACE ADMINISTRATION

12U 001 57 51 3DS 71012 00903
AIR FORCE WEAPONS LABORATORY /WLOL/
KIRTLAND AFB, NEW MEXICO 87117

ATT E. LOU BOWMAN, CHIEF,TECH. LIBRARY

POSTMASTER: If Undeliverable (Section 158 Postal Manual) Do Not Return

---

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

— NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION DIVISION

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546