NASA TECHNICAL NOTE

NASA TN D-6059

C.1

NASA TN D-6059

TECH LIBRARY KAFB, NM

0132830

LOAN COPY: RE
AFWL (DO
KIRTLAND AF

COMPUTER PROGRAM FOR FINITE-DIFFERENCE SOLUTIONS OF SHELLS OF REVOLUTION UNDER ASYMMETRIC DYNAMIC LOADING

by Wendell B. Stephens and Martha P. Robinson

Langley Research Center
Hampton, Va. 23365

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • JANUARY 1971

TECH LIBRARY KAFB, NM

0132830

1. Report No. NASA TN D-6059	2. Government Accession No.	3. Rec.
4. Title and Subtitle COMPUTER PROGRAM FOR FINITE-DIFFERENCE SOLUTIONS OF SHELLS OF REVOLUTION UNDER ASYMMETRIC DYNAMIC LOADING		5. Report Date January 1961
		6. Performing Organization Code
7. Author(s) Wendell B. Stephens and Martha P. Robinson		8. Performing Organization Report No. L-6298
		10. Work Unit No. 124-08-20-04
9. Performing Organization Name and Address NASA Langley Research Center Hampton, Va. 23365		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Note
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

A general computer program written in FORTRAN IV language which determines the linear asymmetric bending behavior of a statically or dynamically loaded elastic thin shell of revolution is presented. The loading may be applied either mechanically or thermally. The variables are separated by representing the loads, displacements, and stresses by Fourier series expansions in the circumferential direction. The resulting set of equations is solved numerically by using finite-difference approximations in the meridional direction and backward differences in the time direction. A three-layered cross section which is symmetric about the middle surface is allowed. The boundary conditions are taken in a general form which allows the program to handle elastic restraints specifying a linear combination of edge forces and displacements. Nonhomogeneous initial conditions are also allowed. The data input procedure is described in detail and sample calculations are included.

17. Key Words (Suggested by Author(s)) Finite-difference solutions; Asymmetric static or dynamic loading; Fourier series summations in circumferential direction	18. Distribution Statement Unclassified – Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	66	$3.00

*For sale by the National Technical Information Service, Springfield, Virginia 22151

COMPUTER PROGRAM FOR FINITE-DIFFERENCE SOLUTIONS OF SHELLS OF REVOLUTION UNDER ASYMMETRIC DYNAMIC LOADING

By Wendell B. Stephens and Martha P. Robinson
Langley Research Center

SUMMARY

A general computer program written in FORTRAN IV language which determines the linear asymmetric bending behavior of a statically or dynamically loaded elastic thin shell of revolution is presented. The loading may be applied either mechanically or thermally. The variables are separated by representing the loads, displacements, and stresses by Fourier series expansions in the circumferential direction. The resulting set of equations is solved numerically by using finite-difference approximations in the meridional direction and backward differences in the time direction. A three-layered cross section which is symmetric about the middle surface is allowed. The boundary conditions are taken in a general form which allows the program to handle elastic restraints specifying a linear combination of edge forces and displacements. Nonhomogeneous initial conditions are also allowed. The data input procedure is described in detail and sample calculations are included.

INTRODUCTION

The linear elastic behavior of any shell of revolution with a static asymmetric load has been programed in reference 1 by using Fourier series expansions along the circumference along the meridian of the shell. The programed analysis contained in reference 1 is based on the analytic formulation presented in reference 2. The shell theory used is that of reference 3. The program in reference 1 calculates the Fourier coefficients of the series but does not perform the summations of the series. This paper extends the analytic formulation and computer program of reference 1 to include asymmetric dynamic response of shells of revolution and includes provisions for summation of the terms of the series. Numerical integration of the dynamic equations is similar to that given in reference 3 for a cylindrical shell and is based on Houbolt's backward difference method (refs. 4 and 5). The loading on the shell may be either mechanical or thermal and these loads may be either static or dynamic. The thickness of the shell may vary along the meridian, but the shell cross section must be symmetric about the shell middle surface. In addition, the initial conditions include provisions for initial deformations. This paper is a user's document which contains the necessary instructions for preparation of input data and subprograms. The program is written in the Control Data version (ref. 6) of FORTRAN IV language for operation in the scope 3.0 digital computer. The program requires an octal storage of 70 000 memory words. The output of the program lists the shell description and the nondimensional Fourier series summations of the displacements, rotations, moments, and force resultants in a tabular (columnwise) format.

The program presented in this paper is illustrated by two dynamic-response example problems. In the first problem a comparison is made with exact results for axisymmetric deformation of a cylindrical shell under initial deformation. The second example demonstrates the input preparation for a conical shell with asymmetric deformations. The second example problem is a planetary entry "aeroshell" subject to asymmetric pressures as the shell passes through the atmosphere with a small oscillation. The data preparation is discussed in detail for this practical application.

SYMBOLS

The units used for the physical quantities 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 reference 7 and those used in the present investigation are presented in appendix A.

a	reference (or characteristic) length
b	nondimensional membrane stiffness defined in appendix C
d	nondimensional bending stiffness defined in appendix C
E_o	reference modulus of elasticity
$\hat{f}_\xi^{(n)}$	nondimensional transverse meridional shear (see appendix B)
f	frequency
h	shell thickness
h_o	reference thickness

L	last boundary station
$M_\xi, M_\theta, M_{\xi\theta}, M_{\theta\xi}$	bending-moment resultants
M_T	nondimensional thermal-moment resultant defined in appendix C
$\overline{M}_{\xi\theta}$	modified twisting moment (eq. (8))
$m_\xi^{(n)}, m_\theta^{(n)}, m_{\xi\theta}^{(n)}$	nondimensional Fourier coefficients for bending moments (see appendix B)
N	total number of stations
$N_\xi, N_\theta, N_{\xi\theta}, N_{\theta\xi}$	membrane force resultants
$\overline{N}_{\xi\theta}$	modified membrane shear (eq. (7))
n	Fourier index
0	first boundary station
$p^{(n)}, p_\xi^{(n)}, p_\theta^{(n)}$	Fourier coefficients for loads (see appendix B)
Q_ξ, Q_θ	transverse shear resultants
q, q_ξ, q_θ	distributed loads in normal, meridional, and circumferential directions, respectively
$\bar{q}$	aerodynamic pressure
r	radial distance from axis of symmetry to shell middle surface
s	shell meridian
$T^{(n)}$	Fourier coefficient for temperature
$T_1^{(n)}(\xi)$	Fourier coefficient for midplane temperature variation (eq. (30))

$\Delta T_1^{(n)}(\xi)$	Fourier coefficient for temperature gradient per unit thickness normal to middle surface (eq. (30))
$\overline{T}$	inverse of frequency f; period
t	face sheet thickness
$\bar{t}$	time
$t_\xi^{(n)},t_\theta^{(n)},\hat{t}_{\xi\theta}^{(n)}$	nondimensional Fourier coefficients for membrane force resultants
$t_T^{(T)}$	nondimensional thermal-force resultant defined in appendix C
U_ξ,U_θ	meridional and circumferential displacements
$u_\xi^{(n)},u_\theta^{(n)}$	nondimensional Fourier coefficients for meridional and circumferential displacements (see appendix B)
W	normal displacement
$w^{(n)}$	nondimensional Fourier coefficient for normal displacement (see appendix B)
α	coefficient of thermal expansion
β	angle between a normal to the shell and fluid flow
$\overline{\alpha},\overline{\beta},\overline{\delta},\overline{\gamma},\hat{\alpha},\hat{\beta}$	coefficients of backward difference expression
$\gamma = \rho'/\rho$	
Δ	meridional increment between interior stations
ϵ	time increment, $\epsilon = \sqrt{\frac{E_O}{\rho a^2}}\,\Delta\bar{t}$
ζ	coordinate normal to and orginating at midsurface of shell, positive outward
θ	circumferential coordinate

$\lambda = h_o/a$	
ν	Poisson's ratio
$\xi = s/a$	
ρ	nondimensional radius, r/a
$\bar{\rho}$	shell density
ρ_a	density of face sheets
σ_o	reference stress
τ	nondimensional time (eq. (9))
Φ_ξ, Φ_θ	meridional rotation
ϕ	colatitude angle
$\phi_\xi^{(n)}$	nondimensional meridional rotation (see appendix B)
ψ	angle of attack
$\overline{\psi}$	amplitude of aeroshell oscillation (eq. (36))
$\omega_\xi, \omega_\theta$	nondimensional curvatures (eqs. (1) and (2))

Matrices:

A,B,C,D,E, F,G,H,J,Ω,Λ	4×4 matrices
e,z,y, f,g,l	4×1 matrices

A prime indicates a derivative with respect to ξ. A dot indicates a derivative with respect to τ. Note that the superscript n is dropped from the Fourier coefficients when doing so would not cause confusion.

ANALYTICAL FORMULATION

Shell Geometry

The shell geometry and coordinates are shown in figure 1 and are identical to those used in reference 1. A point on the shell is specified by coordinates ξ,θ,ζ where $\xi = s/a$ is the nondimensional meridional coordinate, s is the meridional coordinate, a is a reference dimension of the shell, θ is the circumferential coordinate, and ζ is a coordinate normal to and originating at the middle surface, positive outward.

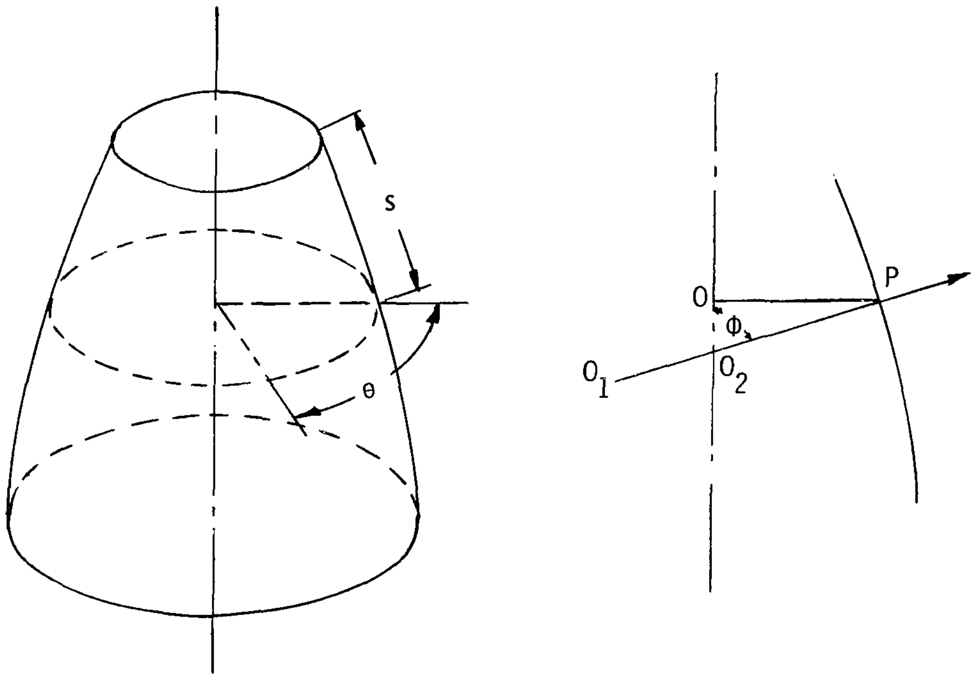


Figure 1.- Surface geometry and coordinates.
$OP = r$; $O_1P = a/\omega_\xi$; and $O_2P = a/\omega_\theta$.

If the shape of the middle surface is given by $\rho = \rho(\xi)$ where $\rho = r/a$ and r is the distance OP, the nondimensional principal curvatures can be written as

$$\omega_\theta = \frac{\left[1 - (\rho')^2\right]^{1/2}}{\rho} \tag{1}$$

$$\omega_\xi = \frac{\gamma' + \gamma^2}{\omega_\theta} \tag{2}$$

where

$$\gamma = \frac{\rho'}{\rho} \tag{3}$$

and the prime indicates a differentiation with respect to ξ.

Dynamic Response Terms

The equations of motion for a shell are obtained from the equilibrium equations of reference 1 by the addition of the acceleration terms as follows:

$$\omega_\xi\left[\left(\rho M_\xi\right)' + \frac{\partial}{\partial\theta}\overline{M}_{\xi\theta} - \rho' M_\theta\right] + a\left[\left(\rho N_\xi\right)' + \frac{\partial}{\partial\theta}\overline{N}_{\xi\theta} - \rho' N_\theta\right]$$

$$+ \frac{1}{2}\left(\omega_\xi - \omega_\theta\right)\frac{\partial}{\partial\theta}\overline{M}_{\xi\theta} + a^2\rho q_\xi = \rho E_o h\,\frac{\partial^2 U_\xi}{\partial\tau^2} \tag{4}$$

$$a\left(\frac{\partial}{\partial\theta}N_\theta + \frac{\partial}{\partial\xi}\rho\overline{N}_{\xi\theta} + \rho' N_{\xi\theta}\right) + \omega_\theta\left(\frac{\partial}{\partial\theta}M_\theta + \frac{\partial}{\partial\xi}\rho\overline{M}_{\xi\theta} + \rho'\overline{M}_{\xi\theta}\right)$$

$$+ \frac{\rho}{2}\frac{\partial}{\partial\xi}\left[\left(\omega_\theta - \omega_\xi\right)\overline{M}_{\xi\theta}\right] + a^2\rho q_\theta = \rho E_o h\,\frac{\partial^2 U_\theta}{\partial\tau^2} \tag{5}$$

$$\frac{\partial}{\partial\xi}\left(\frac{\partial}{\partial\xi}\rho M_\xi + \frac{\partial}{\partial\xi}\overline{M}_{\xi\theta} - \rho' M_\theta\right) + \frac{1}{\rho}\frac{\partial}{\partial\theta}\left(\frac{\partial}{\partial\theta}M_\theta + \frac{\partial}{\partial\xi}\rho\overline{M}_{\xi\theta} + \rho'\,\overline{M}_{\xi\theta}\right)$$

$$- a\rho\left(\omega_\xi N_\xi + \omega_\theta N_\theta\right) + a^2\rho q = \rho E_o h\,\frac{\partial^2 W}{\partial\tau^2} \tag{6}$$

where $\overline{N}_{\xi\theta}$ and $\overline{M}_{\xi\theta}$ are defined in reference 3 as

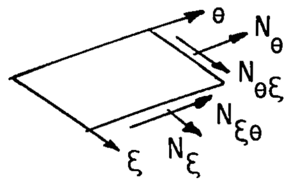
$$\overline{N}_{\xi\theta} = \frac{1}{2}\left(N_{\xi\theta} + N_{\theta\xi}\right) + \frac{1}{4a}\left(\omega_\theta - \omega_\xi\right)\left(M_{\xi\theta} - M_{\theta\xi}\right) \tag{7}$$

$$\overline{M}_{\xi\theta} = \frac{1}{2}\left(M_{\xi\theta} + M_{\theta\xi}\right) \tag{8}$$

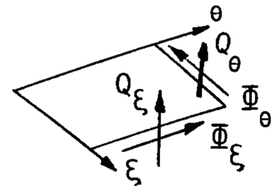
The nondimensional time τ is

$$\tau = \sqrt{\frac{E_o}{\rho a^2}}\,\bar{t} \tag{9}$$

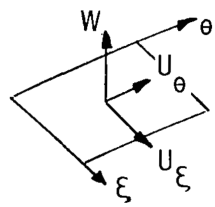
The positive directions of the forces, moments, rotations, and displacements are shown in figure 2.



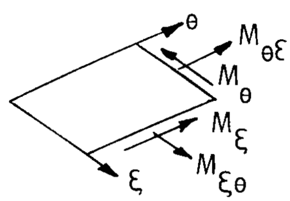
(a) Membrane force resultants.



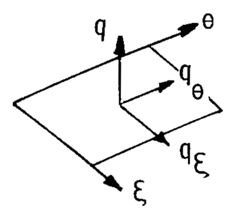
(b) Rotations and transverse force resultants.



(c) Displacements.



(d) Moment resultants.



(e) Loads per unit area.

Figure 2.- Positive sense of force, moments, shears, displacements, and loads on a shell segment.

The Fourier series expansions of the unknowns in equations (4) to (8) and of Φ_ξ and Q_ξ are summarized in appendix B in appropriate nondimensional form. By utilizing these relationships and the stress-strain and strain-displacement relationships, the governing second-order partial differential equations in matrix notation become

$$Ez^{(n)''} + Fz^{(n)'} + Gz^{(n)} = e + D\ddot{z}^{(n)} \tag{10}$$

where

$$z^{(n)} = \begin{Bmatrix} u_\xi^{(n)} \\ u_\theta^{(n)} \\ w^{(n)} \\ m_\xi^{(n)} \end{Bmatrix} \tag{11}$$

and $u_\xi^{(n)}$, $u_\theta^{(n)}$, $w^{(n)}$, and $m_\xi^{(n)}$ are amplitudes of the Fourier harmonics of the three displacements and the meridional moment. The superscript n refers to the nth Fourier harmonic and is omitted hereafter for purposes of simplification in notation. The dots represent derivatives with respect to time τ. The elements of the E, F, G, and e matrices are defined in both references 1 and 2 and are included in appendix C. The D matrix is

$$D = \frac{h}{h_o} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The vector y of stresses and moments is related to the vector z of displacements and rotations by the equation

$$y = Hz' + Jz + f \tag{12}$$

where the H, J, and f matrices are defined in reference 2 and appendix C. The vector y is defined to be

$$y = \begin{Bmatrix} t_\xi \\ \hat{t}_{\xi\theta} \\ \hat{f}_\xi \\ \phi_\xi \end{Bmatrix} \tag{13}$$

The t_ξ, $t_{\xi\theta}$, $\hat{f}_\xi$, and ϕ_ξ components are the Fourier coefficients (see appendix B) of the axial-stress resultant, transverse-stress resultant, shear stress, and rotation variables. It is convenient to express the boundary conditions at either ξ_0 or ξ_L, the end points of the meridional generatrix, in terms of the equation

$$\Omega y + \Lambda z = l \tag{14}$$

where the Ω and Λ matrices and l vector are chosen so that the prescribed displacements and forces at the boundary are satisfied. It should be noted that the dynamic inertia terms affect only the equilibrium equations but not the boundary conditions.

Computational Techniques

The central finite-difference approximations used in equation (10) along the interior stations of the meridian are defined in reference 1 as

$$z_i' = \frac{z_{i+1} - z_{i-1}}{2\Delta} \tag{15}$$

$$z_i'' = \frac{z_{i+1} - 2z_i + z_{i-1}}{\Delta^2} \tag{16}$$

where

$$\Delta = \frac{\xi_L}{N - 2} \tag{17}$$

and where $i = 2, 3, \ldots, N - 1$. Here N is the number of meridional stations. The first derivative of z and value of z at the boundaries of the meridian are

$$\left.\begin{aligned} z_0 &= \frac{1}{2}\left(z_1 + z_2\right) \\ z_0' &= \frac{1}{\Delta}\left(-z_1 + z_2\right) \\ z_L &= \frac{1}{2}\left(z_{N-1} + z_N\right) \\ z_L' &= \frac{1}{\Delta}\left(-z_{N-1} + z_N\right) \end{aligned}\right\} \tag{18}$$

The first shell edge is located midway between stations 1 and 2 and the end of the meridian is located midway between stations $N - 1$ and N. The first and final shell edges are denoted by subscripts 0 and L, respectively.

The time derivatives can be represented by backward differences (as in refs. 5 and 6):

$$\ddot{z}_{i,l} = \bar{\alpha}_l z_{i,l} + \bar{\beta}_l z_{i,l-1} + \bar{\gamma}_l z_{i,l-2} + \bar{\delta}_l z_{i,l-3} + \alpha_l \dot{z}_{i,0} + \hat{\beta}_l \ddot{z}_{i,0}$$

$$(l = 0, 1, 2, \ldots, \quad i = 1, 2, \ldots, N) \qquad (19)$$

where $\bar{\alpha}_l$, $\bar{\beta}_l$, $\bar{\gamma}_l$, and $\bar{\delta}_l$ are constants which depend on the time step and which, in general ($l \geqq 3$), define a four-point backward second derivative. The constants $\hat{\alpha}_l$ and $\hat{\beta}_l$ are required for including nonhomogeneous initial conditions $z_{i,0}$ and $\dot{z}_{i,0}$. The first subscript on z denotes the spatial station and the second subscript denotes the time station. Since $z_{i,0}$ and $\dot{z}_{i,0}$ are given initial conditions that allow $\ddot{z}_{i,0}$ to be calculated from equation (10), fictitious time points at $l = -1$ and $l = -2$ can be obtained by using the difference expression

$$\dot{z}_{i,0} = \frac{1}{6\epsilon}\left(2z_{i,1} + 3z_{i,0} - 6z_{i,-1} + z_{i,-2}\right) \qquad (20)$$

$$\ddot{z}_{i,0} = \frac{1}{\epsilon^2}\left(z_{i,1} - 2z_{i,0} + z_{i,-1}\right) \qquad (21)$$

where ϵ is a time increment of τ. Thus equations (20) and (21), the given initial conditions $z_{i,0}$ and $\dot{z}_{i,0}$, and equations (10) and (12) in finite-difference form comprise enough information to define the coefficients of equation (19). Therefore, at $l = 0$,

$$\left.\begin{aligned} &\bar{\alpha}_0 = \bar{\beta}_0 = \bar{\gamma}_0 = \bar{\delta}_0 = \hat{\alpha}_0 = 0 \\ &\hat{\beta}_0 = 1 \end{aligned}\right\} \qquad (22)$$

at $l = 1$,

$$\left.\begin{aligned} &\bar{\alpha}_1 = \frac{6}{\epsilon^2} \\ &\bar{\beta}_1 = -\frac{6}{\epsilon^2} \\ &\bar{\gamma}_1 = \bar{\delta}_1 = 0 \\ &\hat{\alpha}_1 = \frac{6}{\epsilon} \\ &\hat{\beta}_1 = -2 \end{aligned}\right\} \qquad (23)$$

at $l = 2$,

$$\left.\begin{aligned}
\bar{\alpha}_2 &= \frac{2}{\epsilon^2} \\
\bar{\beta}_2 &= -\frac{4}{\epsilon^2} \\
\bar{\gamma}_2 &= \frac{2}{\epsilon^2} \\
\bar{\delta}_2 &= 0 \\
\hat{\alpha}_2 &= 0 \\
\hat{\beta}_2 &= -1
\end{aligned}\right\} \tag{24}$$

and at $l \geqq 3$,

$$\left.\begin{aligned}
\bar{\alpha}_l &= \frac{2}{\epsilon^2} \\
\bar{\beta}_l &= -\frac{5}{\epsilon^2} \\
\bar{\gamma}_l &= \frac{4}{\epsilon^2} \\
\bar{\delta}_l &= -\frac{1}{\epsilon^2} \\
\hat{\alpha}_l &= \hat{\beta}_l = 0
\end{aligned}\right\} \tag{25}$$

Equation (19) with the constants in equation (25) is the standard four-point backward difference expression for a second derivative. In defining the initial conditions $z_{i,0}$ and $\dot{z}_{i,0}$, only the displacement quantities which are the first three elements of z need to be prescribed since the components in the last row in matrix D are all zero.

Thus, by utilizing equations (10) to (19), the governing equations become

$$\left.\begin{aligned}
&B_1 z_1 + A_1 z_2 = g_1 \\
&A_1 z_{i+1,l} + B_i z_{i,l} + C_i z_{i-1,l} = g_i \qquad (i = 2, 3, \ldots, N-1) \\
&C_N z_{N-1} + B_N z_N = g_N
\end{aligned}\right\} \tag{26}$$

where

$$\left.\begin{aligned}
A_1 &= \left[\Omega_0\left(\frac{J_0}{2} + \frac{H_0}{\Delta}\right) + \frac{\Lambda_0}{2}\right] \\
B_1 &= \left[\Omega_0\left(\frac{J_0}{2} - \frac{H_0}{2}\right) + \frac{\Lambda_0}{2}\right] \\
g_1 &= l_0 - \Omega_0 f_0 \\
B_N &= \left[\Omega_L\left(\frac{J_L}{2} + \frac{H_L}{2}\right) + \frac{\Lambda_L}{2}\right] \\
C_N &= \left[\Omega_L\left(\frac{J_L}{2} - \frac{H_L}{2}\right) + \frac{\Lambda_L}{2}\right] \\
g_N &= l_L - \Omega_L f_L \\
A_i &= \frac{2E_i}{\Delta} + F_i \\
B_i &= \frac{-4E_i}{\Delta} + 2\,\Delta G_i - 2\,\Delta \bar{\alpha}_l D \\
C_i &= \frac{2E_i}{\Delta} - F_i \\
g_i &= 2\,\Delta e_i + 2\,\Delta D\left(\bar{\beta}_l z_{i,l-1} + \bar{\gamma}_l z_{i,l-2} \bar{\delta} z_{i,l-3} + \hat{\alpha}_l \dot{z}_{i,0} + \hat{\beta}_l \ddot{z}_{i,0}\right)
\end{aligned}\right\} \qquad (27)$$

Thus equations (26) are the complete set of equations required to solve for the vector $z_{i,l}$ at all stations i out to and including time step l.

Step Loads

Step loads or suddenly applied loads represent a discontinuity in the load history at a point in time k and, in essence, impart an acceleration to the shell. At this point in time k,

$$\left.\begin{aligned}
z_{i,k}^{-} &= z_{i,k}^{+} \\
\dot{z}_{i,k}^{-} &= \dot{z}_{i,k}^{+}
\end{aligned}\right\} \qquad (28)$$

but

$$\ddot{z}_{i,k}^{-} \neq \ddot{z}_{i,k}^{+} \tag{29}$$

The superscripts - and + indicate quantities obtained by taking the limits of the quantities at a time k from below and above, respectively. The acceleration $\ddot{z}_{i,k}^{+}$ is obtained from equation (10) with loads at k^{+} imposed along with the deflection and velocity at k. Thus, the problem becomes essentially another initial-value problem with initial conditions given at time k. With $z_{i,k}$, $\dot{z}_{i,k}$, and $\ddot{z}_{i,k}^{+}$ vectors computed, the problem is continued similarly to the sequence of equations (20) to (25).

COMPUTER PROGRAM

Program Organization

The program developed in this paper takes the basic program in reference 1 and modifies that program in two major ways. First, the program HGS of reference 1 is modified to include a time-step cycling procedure to account for the inertia terms. The second major change is that the Fourier coefficients for 11 variables $\left(N_\xi,\ N_\theta,\ \overline{N}_{\xi\theta},\ Q_\xi,\ M_\xi,\ M_\theta,\ \overline{M}_{\xi\theta},\ U_\xi,\ U_\theta,\ W,\text{ and }\Phi_\xi\right)$ contained in appendix A are summed for the series truncated at some value n and for each point in time l. The value n at which the Fourier series are truncated is supplied by the user.

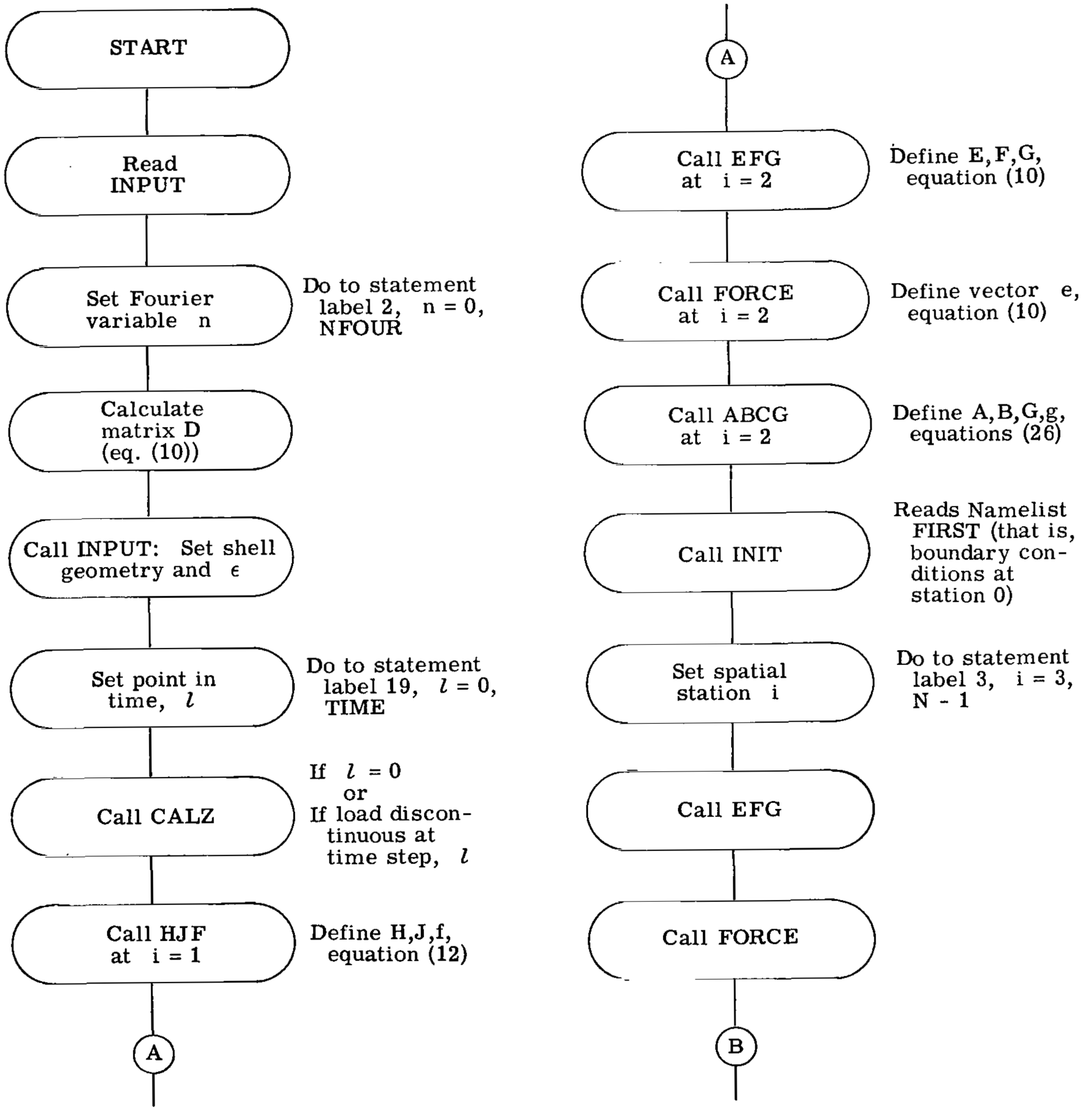
To accomplish the first modification the following subroutines were altered or added to program HGS of reference 1.

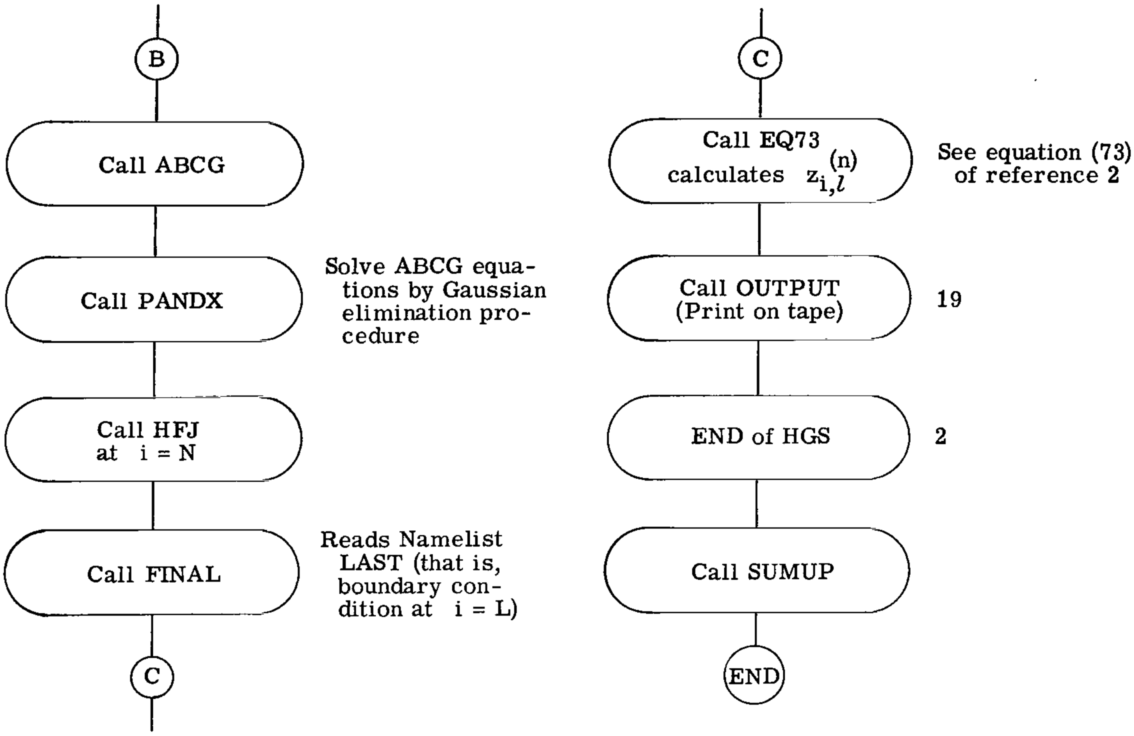
Subroutine	Remark
MAIN	MAIN structures the program so that HGS will calculate coefficients of the 11 output variables at all interior stations i and boundary stations 0 and L for each station in time l and each Fourier variable n. It has been modified to include the inertial terms and the time-stepping cycles.
INPUT	INPUT is a user-prepared subroutine which supplies the shell geometry. In addition, the time increment ϵ is supplied by this subroutine for the time-dependent problem.
CALZ	The subroutine CALZ stores or computes $z_{i,0}$, $\dot{z}_{i,0}$, and $\ddot{z}_{i,0}$ and is the only new subroutine added to HGS.

ABCG — The subroutine ABCG sets the coefficients $\overline{\alpha}$, $\overline{\beta}$, $\overline{\gamma}$, $\overline{\delta}$, $\hat{\alpha}$, and $\hat{\beta}$ in equation (19) as well as elements of the A, B, C, and g matrices in equation (26). Note that B and g include time-dependent terms.

OUTPUT — The subroutine OUTPUT controls program printing. In addition, it stores the vectors $z_{i,l}$, $z_{i,l-1}$, and $z_{i,l-2}$ for use in time step $l + 1$ computations.

The flow chart of the present program follows.





The second modification, that of summing the 11 variables at each time step, is accomplished by a followup program called SUMUP. Here SUMUP is actually a separate program that uses the same control cards as the HGS program. Thus, there are two programs but only one deck. The program HGS stores the information for all required time stations l and Fourier integers n of the 11 output variables on tape. The program SUMUP rewinds and reads the tape. Then SUMUP performs the series summation of the truncated series by using the Fourier coefficient taped from HGS.

A listing of HGS and SUMUP is included in appendix D. Note that all subroutines which are not discussed in this report contain COMMENT cards in the listing explaining their functions. In addition, an asterisk in the right-hand column denotes all the statements that differ from the program presented in reference 1. The glossary of FORTRAN names of the variables is given in reference 1.

Input Data

The input data are those of reference 1 with some deletions and additions. These input data are now read into program HGS through the Namelists INPUTD, FIRST, and LAST. The variable type R stands for a real (floating point) value and I denotes an integer quantity. The complete list of Namelist INPUTD quantities and their definitions follow:

Name	FORTRAN variable type	Description
NU	R	Poisson's ratio
TKN	R	reference thickness, h_o
CHAR	R	reference shell dimension, a
EALSIG	R	$E\alpha/\sigma_o$, thermal coefficient used only if thermal stresses are calculated
IND5	I	an integer value of 0, 1, 2, or 3 = 0, no poles; = 1, pole at $\xi = 0$; = 2, poles at $\xi = 0$ and ξ_L = 3, pole at $\xi = \xi_L$
NMAX	I	total number of meridional stations
FREQ	I	integer which controls the frequency for printing numerical results. Results are printed at every FREQth station.
NTIME	I	number of time steps of size ϵ which will be taken
JUMP	I	integer array (five elements) defining time stations at which a load is suddenly applied. If not required, then set JUMP(i) > NTIME.
RUNTYPE	I	integer defining type of problem to be run: = 1, static case = 2, dynamic-response problem with either $z_{i,0}$ and/or $\dot{z}_{i,0}$ specified by functions ZINIT and ZDINIT = 3, dynamic response problem
NFOUR	I	last Fourier value to be summed
KTHTIME	I	time-station interval at which coefficients of the 11 variables from N = 0 to NFOUR will be printed out. If none other than at the zeroeth time station are desired, set KTHTIME > NTIME.

The namelists FIRST and LAST describe the boundary conditions at station 0 and L, respectively, and are required when IND5 of Namelist INPUTD fails to define a pole point at that station. These namelists define the elements of the matrices in equation (14). The input quantities of Namelist FIRST are

Name	Type	Description
OMEG1	R	a 4×4 array defining the Ω matrix of equation (14)
CAPL1	R	a 4×4 array defining the Λ matrix of equation (14)
EL1	R	a 4×1 array defining the l vector of equation (14)

Similarly, the Namelist LAST quantities are

OMEGL	R	a 4×4 array defining the Ω matrix of equation (14)
CAPLL	R	a 4×4 array defining the Λ matrix of equation (14)
ELL	R	a 4×1 array defining the l vector of equation (14)

In addition to these input values there are various user-prepared subprograms. These subprograms, for the most part, are adequately described in reference 1. They include:

Subroutine INPUT(NMAX)	defining the arrays of ρ, γ, ω_θ, ω_ξ, ω_ξ' and the increments Δ and ϵ
FUNCTION HHT(K, DEL)	defining h/h_o
FUNCTION DHHT(K, DEL)	defining $(h/h_o)'$
FUNCTION HRA(K, DEL)	defining h/t
FUNCTION DHRA(K, DEL)	defining $(h/t)'$
FUNCTION P(K, DEL)	defining $p^{(n)}$
FUNCTION PX(K, DEL)	defining $p_\xi^{(n)}$
FUNCTION PT(K, DEL)	defining $p_\theta^{(n)}$
FUNCTION ZINIT(KK)	defines $z_{i,0}$

FUNCTION ZDINIT(KK) defines $\dot{z}_{i,0}$ where

KK = 1 corresponds to u_ξ and $\dot{u}_\xi$
KK = 2 corresponds to u_ξ and $\dot{u}_\theta$
KK = 3 corresponds to w and $\dot{w}$

FUNCTION TEMP(K, DEL) defining $T_1^{(n)}$

FUNCTION DELT(K, DEL) defining $\Delta T_1^{(n)}$

FUNCTION DTEMP(K, DEL) defining $T_1^{(n)'}$

FUNCTION DDELT(K, DEL) defining $\Delta T_1^{(n)'}$ where h and t are the shell and face sheet thickness, respectively.

The last four functions define the quantities and derivatives of the temperature equation

$$T^{(n)} = T_1^{(n)}(\xi) + \Delta T_1^{(n)}(\xi)\,\zeta \tag{30}$$

where $T_1^{(n)}(\xi)$ is the Fourier coefficient of the temperature at the reference surface and $\Delta T_1^{(n)}(\xi)$ is the temperature difference between the inner and outer surfaces per unit thickness.

The two additional functions, ZINIT and ZDINIT, have been added for use with the initial conditions required for RUNTYPE = 2. These functions define the initial replacements and velocities required by the initial conditions.

Program Output

The output comes from both programs HGS and SUMUP. The output from the program HGS consists of a complete list of the input data and of the shell geometry in tabular form (that is, column format) at all stations. This output is followed by tabular listing of the Fourier coefficients of the 11 variables in program HGS at the FREQth stations and at the KTHTIME cycle for each value of n. The output of program SUMUP then follows with the complete summation of the Fourier series to n equals NFOUR for the 11 nondimensional variables at each KTHTIME cycle. The printout of SUMUP is at a value of θ where

$$\theta = AK\pi \tag{31}$$

and AK is an INPUTD quantity.

Estimation of Increment Size

A discussion of the meridional increment size is contained in reference 1. The size of the time increment ϵ from equations (19) to (24) also affects the results. Basically, as the increment ϵ tends toward infinity, the dynamic response is damped out. Thus, the increment ϵ must be made sufficiently small. As shown in reference 4, the Houbolt method of initiating the problem and using backward differences is a stable method of numerical integration for transient problems. The selection can be made by comparing the results of increment size where increment is decreased by one-half until agreement between the results is obtained within suitable limits.

EXAMPLE PROBLEMS

Cylindrical Shell

The purpose of this example is simply to demonstrate the accuracy of the program. The problem is that of a simply supported cylindrical shell loaded laterally with an axisymmetric sinusoidal pressure oscillating with respect to time. The simply supported boundaries are free to displace in the u_ξ-direction at each end. The pressure $q(\xi,\bar{t})$ becomes

$$q(\xi,\bar{t}) = p^{(0)} \sin \frac{\pi\xi}{\xi_L} \cos 2\pi f\bar{t}$$

For this particular example, the frequency f and pressure amplitude $p^{(0)}$ are taken to be

$$f = 10\,000 \text{ Hz}$$

$$p^{(0)} = 1$$

The geometric parameters of the shell are

$$s/r = 10$$

$$r/h = 100$$

$$h/t = 2$$

Since at $\bar{t} = 0$, there is an applied load on the structure, the initial displacements are assumed to be

$$U_\xi = \bar{u} \cos \frac{\pi\xi}{\xi_L}$$

$$U_\theta = 0$$

$$W = \bar{w} \sin \frac{\pi\xi}{\xi_L}$$

where the amplitudes $\bar{u}$ and $\bar{w}$ are calculated from exact linear theory (ref. 3) to be

$$\bar{u} = 0.107388$$

$$\bar{w} = 1.11396$$

Further, the initial velocity becomes

$$\dot{z}_{i,0} = 0$$

for all i.

In figure 3 the dynamic response of the deflection W at $\xi = 0.5$ is compared with the exact solution by axisymmetric linear theory. (See ref. 3.) The curve in figure 3 shows that this numerical solution and the exact solution are in close agreement. The numerical input for this problem was

$$NMAX = 21$$

$$\nu = 0.3$$

$$\Delta\bar{t} = 0.1 \text{ sec}$$

$$\rho = 0.000259 \frac{\text{lbm}}{\text{in}^3} \left(7.221 \frac{\text{kg}}{\text{m}^3}\right)$$

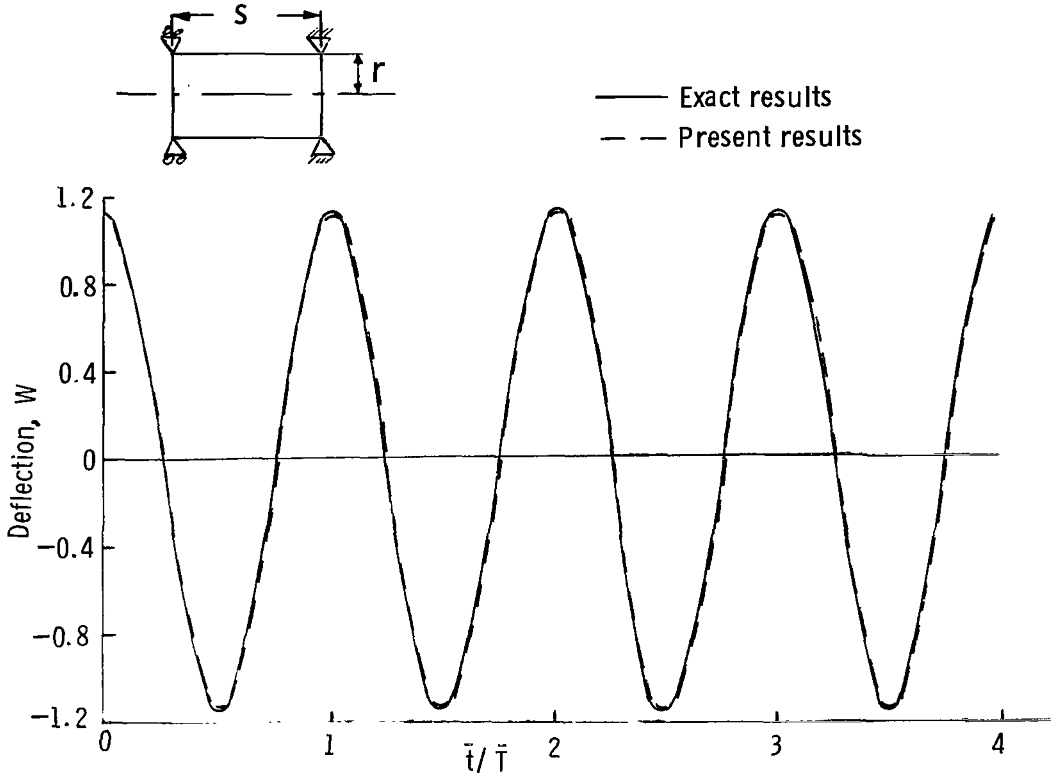


Figure 3.- Dynamic response of the deflection W at $\xi_L/2$ for cylindrical shell example.

Conical Shell

Description.- In order to demonstrate the data preparation required for using this program, a practical aeroshell problem is analyzed. The simply supported 120° truncated conical shell of sandwich construction studied in reference 8 has been selected. For this shell the loading is the aerodynamic pressure q acting laterally on the shell as it passes through the atmosphere. In addition, the shell axis has a small wobble or angle of attack ψ which oscillates as a function of time, and thereby causes the loading to be applied asymmetrically. The purpose of such an analysis could be to ascertain whether the oscillations of the shell axis cause any stress buildup in the stress resultant N_θ. It would be expected that any increase in N_θ would have an important effect on shell instability. The shell cross section along with its physical dimensions is shown in figure 4. The face sheets are made of aluminum and have a density ρ_a of 0.1 lb/in^3 (2.8 Mg/m^3) and the core honeycomb has a density equal to $0.03\rho_a$. Thus, the average density of the shell $\bar{\rho}$ is given by

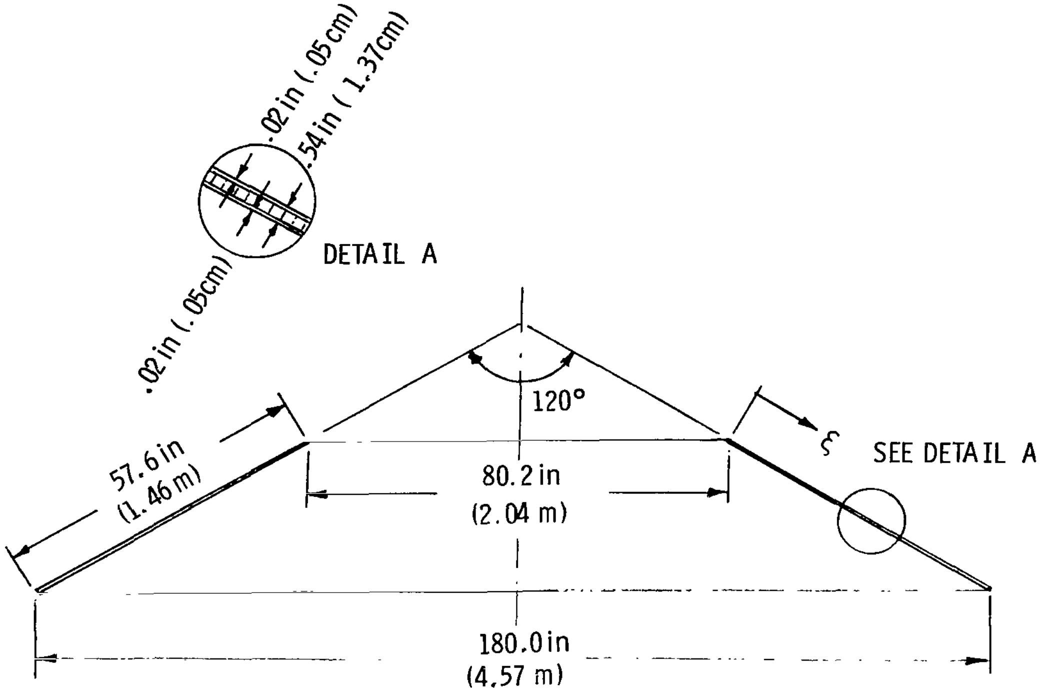


Figure 4.- Physical dimensions of the conical shell.

$$\bar{\rho} = \rho_a\left(\frac{2t}{h} + 0.03\frac{h - 2t}{h}\right) \tag{32}$$

The dynamic forces are the Newtonian impact forces on the shell as it passes through the atmosphere with a small wobble ψ and are defined to be

$$p = -2\bar{q} \cos^2 \beta \tag{33}$$

where β is the angle between the normal to the shell reference surface and direction of the fluid flow and $\bar{q}$ is the aerodynamic pressure. Since ϕ is the colatitude angle (see fig. 1) of a point on the shell meridian, equation (33) becomes

$$p = -2\bar{q}\left(\frac{1}{2} \sin^2 \psi \sin^2 \phi + \cos^2 \psi \cos^2 \phi + 2 \cos \psi \cos \phi \sin \psi \sin \phi \cos \theta + \frac{1}{2} \sin^2 \psi \sin^2 \phi \cos^2 \theta\right) \tag{34}$$

Thus from equation (34) and appendix B, the Fourier pressure coefficients become

$$\left.\begin{aligned} p^{(0)} &= -\bar{q}\left(\sin^2 \psi \sin^2 \phi + 2 \cos^2 \psi \cos^2 \phi\right) \\ p^{(1)} &= -4\bar{q}\left(\cos \psi \cos \phi \sin \psi \sin \phi\right) \\ p^{(2)} &= -\bar{q}\left(\sin^2 \psi \sin^2 \phi\right) \end{aligned}\right\} \tag{35}$$

Here the angle $\psi(t)$ is given by

$$\psi = \bar{\Psi} \sin 2\pi f \bar{t} \tag{36}$$

where $\bar{\Psi}$ is the amplitude of the oscillation and, for this example, is 5° or $\pi/36$. The frequency of oscillation f is 10 Hz.

From equation (12) the simply supported boundary at $\xi = 0$ yields the conditions:

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} t_{\xi} \\ t_{\xi\theta} \\ \hat{f}_{\xi} \\ \phi_{\xi} \end{Bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} u_{\xi} \\ u_{\theta} \\ w \\ m_{\xi} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \tag{37}$$

The edge at $\xi = \xi_L$ is pinned and restrained from horizontal displacement but allowed to displace in the vertical (axial) direction; thus,

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\sin \phi & 0 & \cos \phi & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} t_{\xi} \\ t_{\xi\theta} \\ \hat{f}_{\xi} \\ \phi_{\xi} \end{Bmatrix} + \begin{bmatrix} \cos \phi & 0 & \sin \phi & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} u_{\xi} \\ u_{\theta} \\ w \\ m_{\xi} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} \tag{38}$$

The initial conditions are taken as

$$z_{i,0} = \dot{z}_{i,0} = 0 \tag{39}$$

Namelist INPUT.- In keeping with the definitions in section on input data, the Namelist INPUTD values become

NU = 0.32

TKN = 0.54

CHAR = 90.

EALSIG = 0

IND5 = 0

NFOUR = 2

NMAX = 76

FREQ = 2

JUMP(1) = 82,83,84,85,86

NTIME = 81

RUNTYPE = 3

KTHTIME = 82

AK = 0.

The boundary conditions required for Namelists FIRST and LAST have been given in equations (37) and (38) and correspond with equation (12). In Namelist format the values for Namelists FIRST and LAST are:

OMEG1(1,1) = 16*0.

CAPL1(1,1) = 1.,4*0.,1.,4*0.,1.,4*0.,1.

EL1(1) = 4*0.

OMEGL(1,1) = 2*0.,-0.5,7*0.,0.866,5*0.

CAPLL(1,1) = .866,4*0.,1.0,2*0.,.5,6*0.,1.0

ELL(1) = 4*0.

User-prepared subprograms.- The pertinent statements for the conical geometry to be described in subroutine INPUT (see appendix D) are

```
      RI = 40.1
      RO = 90.0
      PI = 3.1415926535979
      ANG = PI/6
      DEL = (RO-RI)/(RO*COS(ANG)*FLOAT(NMAX-2))
      R(NMAX) = 1.0
      R(I) = RI/RO
      DELR = (RO-RI)/FLOAT(NMAX-2)
      R(2) = R(1) + DELR/2
      NM1 = NMAX -1
      DO1I = 3, NMI
    1 R(I) = R(I - 1) + DELR
      DO2I = 1,NMAX
      GAM(I) = COS(ANG)/R(I)
      OMT(I) = SIN(ANG)/R(I)
      OMXI(I) = 0.
    2 DEOMX(I) = 0.
C  IF RUNTYPE = 2 OR 3 THEN SET EE AND RHO TO FIND TIME INCREMENT, EPS.
C  HERE EE IS THE MODULUS OF ELASTICITY AND RHO IS DENSITY (LB/IN**3)
      GACC = 386.088527
      EE = 10500000.
      RHOA = 0.1
```

```
RHO = RHOA*2.75/27.

SS = EE/RHO

EORHO = SS*GACC

EPS = SQRT(EORHO/CHAR**2)*TIM
```

A complete listing of the program is contained in appendix D. The statements required for the function subprograms become

```
HHT = 1.0

DHHT = 0.

HRA = 27.

DHRA = 0.

PX = 0.

PT = 0.

TEMP = 0.

DELT = 0.

DTEMP = 0.

DDELT = 0.

ZINIT = 0.

ZDINIT = 0.
```

It should be noted that the left-hand side of these statements defines the function subprogram in which the statement is used. The statements required for function P(K) are more involved. By setting $\bar{q}$ in equation (35) equal to unity and recalling equation (36), the required statements become

```
PI = 3.14159265358979

CPS = 10.

TIM = 0.05/CPS
```

```
TAU = FLOAT(JTIME)*TIM

AO = 5.

A = AO*PI/180.

FEE = PI/6.

PSI = A*SIN(CPS*TAU*2.*PI)

Q = 1./LAM

SA = SIN(PSI)

CA = COS(PSI)

SF = SIN(FEE)

CF = COS(FEE)

P0 = -Q*(SA*SA*SF*SF + 2.*CA*CA*CF*CF)

P1 = -4.*Q*CA*SA*CF*SF

P2= -Q*SA*SA*SF*SF

IF(N.EQ.0.)P = P0

IF(N.EQ.1.)P = P1

IF(N.EQ.2.)P = P2
```

Output.- The results generated by the program HGS are summed by the followup program SUMUP. The tabular results at $\theta = 0$ (that is, AK = 0 in Namelist INPUTD) are printed for the 11 variables and are shown for time $\bar{t} = 0$. It should be noted that the output is nondimensional. In other words, the multiplicative constants involving σ_o, h_o, a, and E_o in front of the summation signs in appendix B are not included for the output results. In order to get dimensional results, these nondimensional results must be multiplied by the appropriate constants of appendix B.

THE TIMESTEP IS 0 TIME= 0.

STA	N XI	N THETA	N XITHETA	SHEAR	M XI	M THETA	M XITHETA	U XI	UTHETA	W	PHI
1	4.247E+02	1.359E+02	0.	-1.447E+01	-1.455E-11	2.616E+03	0.	9.038E-12	0.	9.095E-12	0.
2	4.219E+02	1.102E+02	0.	-1.276E+01	-1.716E+03	2.025E+03	0.	2.225E+01	0.	-3.389E+02	0.
4	4.102E+02	1.523E+01	0.	-6.928E+00	-6.147E+03	3.797E+02	0.	1.149E+02	0.	-1.654E+03	0.
6	3.965E+02	-6.597E+01	0.	-2.922E+00	-8.188E+03	-5.733E+02	0.	2.112E+02	0.	-2.861E+03	0.
8	3.814E+02	-1.323E+02	0.	-3.210E-01	-8.657E+03	-1.035E+03	0.	3.097E+02	0.	-3.931E+03	0.
10	3.656E+02	-1.847E+02	0.	1.245E+00	-8.163E+03	-1.166E+03	0.	4.089E+02	0.	-4.857E+03	0.
12	3.493E+02	-2.251E+02	0.	2.033E+00	-7.137E+03	-1.088E+03	0.	5.078E+02	0.	-5.647E+03	0.
14	3.329E+02	-2.555E+02	0.	2.428E+00	-5.874E+03	-8.893E+02	0.	6.055E+02	0.	-6.320E+03	0.
16	3.167E+02	-2.782E+02	0.	2.457E+00	-4.563E+03	-6.312E+02	0.	7.014E+02	0.	-6.895E+03	0.
18	3.009E+02	-2.952E+02	0.	2.299E+00	-3.320E+03	-3.546E+02	0.	7.950E+02	0.	-7.393E+03	0.
20	2.854E+02	-3.082E+02	0.	2.039E+00	-2.207E+03	-8.566E+01	0.	8.861E+02	0.	-7.836E+03	0.
22	2.705E+02	-3.185E+02	0.	1.733E+00	-1.255E+03	1.603E+02	0.	9.744E+02	0.	-8.241E+03	0.
24	2.560E+02	-3.274E+02	0.	1.414E+00	-4.708E+02	3.746E+02	0.	1.060E+03	0.	-8.623E+03	0.
26	2.420E+02	-3.355E+02	0.	1.096E+00	1.466E+02	5.525E+02	0.	1.143E+03	0.	-8.994E+03	0.
28	2.285E+02	-3.435E+02	0.	7.859E-01	6.039E+02	6.914E+02	0.	1.223E+03	0.	-9.365E+03	0.
30	2.154E+02	-3.519E+02	0.	4.310E-01	9.075E+02	7.892E+02	0.	1.301E+03	0.	-9.742E+03	0.
32	2.027E+02	-3.607E+02	0.	1.756E-01	1.061E+03	8.436E+02	0.	1.377E+03	0.	-1.013E+04	0.
34	1.903E+02	-3.702E+02	0.	-1.378E-01	1.066E+03	8.516E+02	0.	1.450E+03	0.	-1.053E+04	0.
36	1.783E+02	-3.803E+02	0.	-4.659E-01	9.135E+02	8.091E+02	0.	1.521E+03	0.	-1.095E+04	0.
38	1.665E+02	-3.903E+02	0.	-8.141E-01	6.118E+02	7.113E+02	0.	1.590E+03	0.	-1.137E+04	0.
40	1.550E+02	-4.014E+02	0.	-1.184E+00	1.379E+02	5.528E+02	0.	1.657E+03	0.	-1.181E+04	0.
42	1.438E+02	-4.113E+02	0.	-1.575E+00	-5.117E+02	3.284E+02	0.	1.722E+03	0.	-1.224E+04	0.
44	1.328E+02	-4.214E+02	0.	-1.979E+00	-1.343E+03	3.348E+01	0.	1.786E+03	0.	-1.266E+04	0.
46	1.220E+02	-4.298E+02	0.	-2.385E+00	-2.360E+03	-3.352E+02	0.	1.847E+03	0.	-1.306E+04	0.
48	1.115E+02	-4.362E+02	0.	-2.775E+00	-3.556E+03	-7.781E+02	0.	1.907E+03	0.	-1.342E+04	0.
50	1.013E+02	-4.400E+02	0.	-3.126E+00	-4.918E+03	-1.293E+03	0.	1.965E+03	0.	-1.372E+04	0.
52	9.145E+01	-4.402E+02	0.	-3.405E+00	-6.413E+03	-1.872E+03	0.	2.020E+03	0.	-1.394E+04	0.
54	8.197E+01	-4.351E+02	0.	-3.580E+00	-8.014E+03	-2.502E+03	0.	2.073E+03	0.	-1.406E+04	0.
56	7.293E+01	-4.259E+02	0.	-3.605E+00	-9.544E+03	-3.164E+03	0.	2.124E+03	0.	-1.406E+04	0.
58	6.441E+01	-4.117E+02	0.	-3.434E+00	-1.123E+04	-3.830E+03	0.	2.171E+03	0.	-1.390E+04	0.
60	5.650E+01	-3.899E+02	0.	-3.015E+00	-1.266E+04	-4.465E+03	0.	2.215E+03	0.	-1.357E+04	0.
62	4.928E+01	-3.607E+02	0.	-2.793E+00	-1.381E+04	-5.022E+03	0.	2.256E+03	0.	-1.305E+04	0.
64	4.284E+01	-3.245E+02	0.	-1.214E+00	-1.453E+04	-5.443E+03	0.	2.292E+03	0.	-1.231E+04	0.
66	3.726E+01	-2.802E+02	0.	2.753E-01	-1.464E+04	-5.675E+03	0.	2.324E+03	0.	-1.135E+04	0.
68	3.263E+01	-2.235E+02	0.	2.225E+00	-1.392E+04	-5.631E+03	0.	2.351E+03	0.	-1.017E+04	0.
70	2.902E+01	-1.710E+02	0.	4.574E+00	-1.218E+04	-5.231E+03	0.	2.373E+03	0.	-8.777E+03	0.
72	2.646E+01	-1.080E+02	0.	7.552E+00	-9.153E+03	-4.385E+03	0.	2.390E+03	0.	-7.203E+03	0.
74	2.498E+01	-4.201E+01	0.	1.117E+01	-4.512E+03	-2.997E+03	0.	2.402E+03	0.	-5.495E+03	0.
76	2.450E+01	7.345E+00	0.	1.415E+01	0.	-1.526E+03	0.	2.407E+03	0.	-4.169E+03	0.

For this particular example, the maximum stress resultant value of N_θ which occurs at $\xi \approx 0.70\xi_L$ is plotted in figure 5. Also included are the results when the oscillation frequency f is 1 Hz or 100 Hz. In figure 5 the time axis is made nondimensional by $\bar{T}$, the time required for one cycle of oscillation ($\bar{T} = 1/f$), for ease in comparing different oscillation frequencies. The time increment ϵ taken for this example was $0.05\bar{T}$. The difference in the results for frequencies of 1 Hz and 10 Hz is negligible whereas for the 100-Hz case, there is substantial increase in compressive stress. The time-response values at 1 Hz and 10 Hz settle down almost immediately to a harmonic steady-state response pattern whereas the 100-Hz curve is still nonharmonic after four cycles because of the larger change in $\ddot{z}_{i,0}$. The change in pressure is occurring so rapidly in this case that the rate of change in loading is approximating a step load.

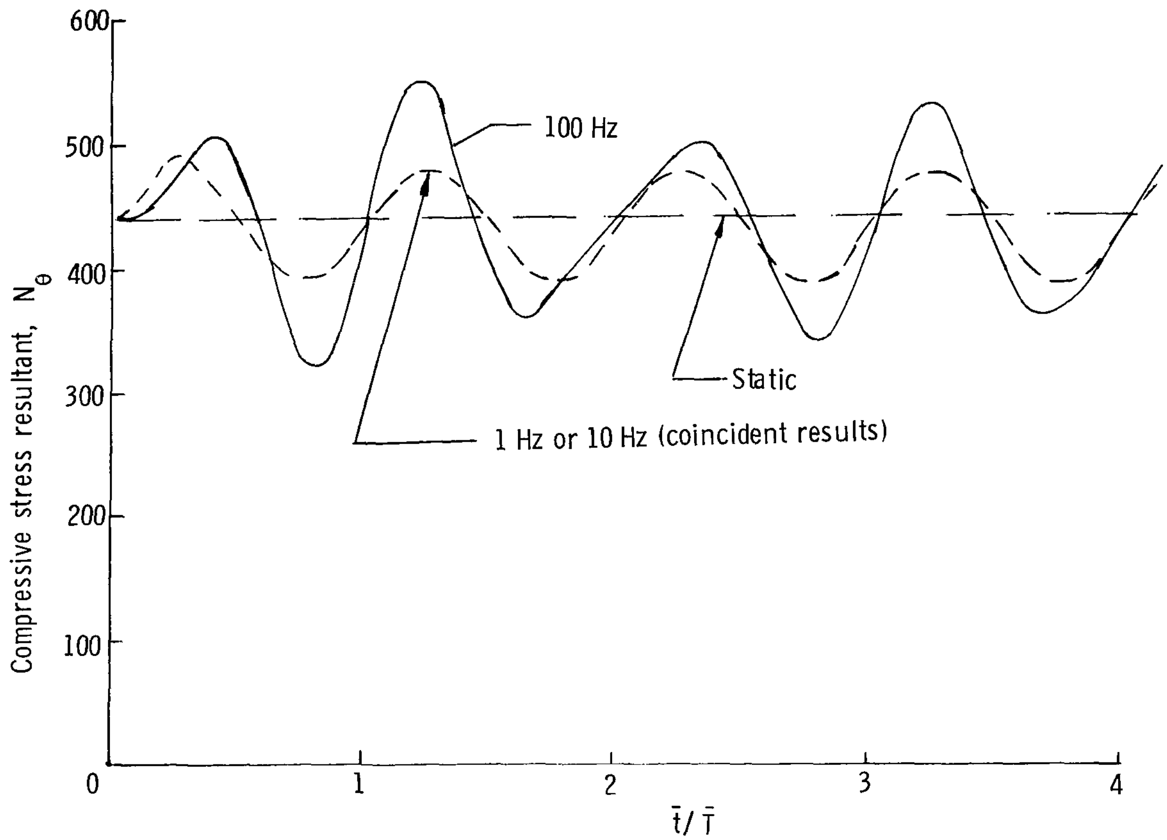


Figure 5.- Maximum stress resultant ($\xi \approx 0.70$ and $\theta = 0^\circ$) as a function of time for various oscillation frequencies.

CONCLUDING REMARKS

This report describes the development of a computer program for the linear asymmetric bending behavior of a statically or dynamically loaded elastic thin shell of revolution subjected to either thermal or mechanical loads. The program is an extension of the static analysis described in NASA TN D-3926 to include dynamic loads and summation of Fourier components. These changes required the addition of the dynamic terms in the equations as well as the restructuring of the program to include the initial conditions and backward numerical integration of time derivatives. Two examples, demonstrating the flexibility of the program as well as the data preparation required, have also been included. The first example, response of a thin cylindrical shell to an oscillating pressure, demonstrates the accuracy of the program for dynamic-response analysis. The second example is used to demonstrate the preparation of user-supplied data for a practical analysis.

Langley Research Center,
National Aeronautics and Space Administration,
Hampton, Va., September 22, 1970.

APPENDIX A

CONVERSION OF U.S. CUSTOMARY UNITS TO SI UNITS

The units used for the physical quantities 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 reference 7 and those used herein are given in the following table:

Physical quantity	U.S. Customary Unit	Conversion factor (a)	SI Unit (b)
Length	in.	0.0254	meters (m)
Density	lbm/in^3	27.68×10^3	$kilograms/meter^3$ (kg/m^3)
Modulus, elastic	psi = lbf/in^2	6895	$newtons/meter^2$ (N/m^2)

[a]Multiply value given in U.S. Customary Unit by conversion factor to obtain equivalent value in SI Unit.

[b]Prefixes to indicate multiple of units are as follows:

Prefix	Multiple
micro (μ)	10^{-6}
milli (m)	10^{-3}
kilo (k)	10^3
giga (G)	10^9

APPENDIX B

FOURIER SERIES EXPANSIONS

The Fourier series expansion of the dependent variables in the circumferential direction is presented in this appendix. The constant terms to the left of the summation symbol are required to nondimensionalize the series coefficients in a consistent manner.

$$N_\xi = \sigma_o h_o \sum_{n=0}^{\infty} t_\xi^{(n)} \cos n\theta$$

$$N_\theta = \sigma_o h_o \sum_{n=0}^{\infty} t_\theta^{(n)} \cos n\theta$$

$$\overline{N}_{\xi\theta} = \sigma_o h_o \sum_{n=1}^{\infty} \hat{t}_{\xi\theta}^{(n)} \sin n\theta$$

$$M_\xi = \frac{\sigma_o h_o^3}{a} \sum_{n=0}^{\infty} m_\xi^{(n)} \cos n\theta$$

$$M_\theta = \frac{\sigma_o h_o^3}{a} \sum_{n=0}^{\infty} m_\theta^{(n)} \cos n\theta$$

$$\overline{M}_{\xi\theta} = \frac{\sigma_o h_o^3}{a} \sum_{n=1}^{\infty} m_{\xi\theta}^{(n)} \sin n\theta$$

$$U_\xi = \frac{a\sigma_o}{E_o} \sum_{n=0}^{\infty} u_\xi^{(n)} \cos n\theta$$

$$U_\theta = \frac{a\sigma_o}{E_o} \sum_{n=1}^{\infty} u_\theta^{(n)} \sin n\theta$$

$$W = \frac{a\sigma_o}{E_o} \sum_{n=0}^{\infty} w^{(n)} \cos n\theta$$

$$q = \frac{\sigma_o h_o}{a} \sum_{n=0}^{\infty} p^{(n)}(\xi) \cos n\theta$$

$$q_\xi = \frac{\sigma_o h_o}{a} \sum_{n=0}^{\infty} p_\xi^{(n)}(\xi) \cos n\theta$$

$$q_\theta = \frac{\sigma_o h_o}{a} \sum_{n=1}^{\infty} p_\theta^{(n)}(\xi) \sin n\theta$$

$$\Phi_\xi = \frac{\sigma_o}{E_o} \sum_{n=0}^{\infty} \phi_\xi^{(n)} \cos n\theta$$

$$Q_\xi = \sigma_o h_o \sum_{n=0}^{\infty} \hat{f}_\xi^{(n)} \cos n\theta$$

APPENDIX C

DEFINITION OF MATRICES

As shown in reference 2, the nonzero elements of the matrices E, F, G, and e are:

$$E_{11} = b$$

$$E_{22} = \frac{b(1 - \nu)}{2} + \frac{\lambda^2 d(1 - \nu)\left(3\omega_\theta - \omega_\xi\right)^2}{8}$$

$$E_{23} = \frac{\lambda^2 d(1 - \nu)\left(3\omega_\theta - \omega_\xi\right)n}{2\rho}$$

$$E_{32} = E_{23}$$

$$E_{33} = \lambda^2 d(1 - \nu)\left[\frac{2n^2}{\rho^2} + (1 + \nu)\gamma^2\right]$$

$$E_{34} = \lambda^2$$

$$E_{43} = -d$$

$$F_{11} = \gamma b + b'$$

$$F_{12} = \frac{(1 + \nu)bn}{2\rho} + \frac{\lambda^2 dn(1 - \nu)}{8\rho}\left(3\omega_\xi - \omega_\theta\right)\left(3\omega_\theta - \omega_\xi\right)$$

$$F_{13} = b\left(\omega_\xi + \nu\omega_\theta\right) + \lambda^2 d(1 - \nu)\left[(1 + \nu)\gamma^2\omega_\xi + \left(\frac{n^2}{2\rho^2}\right)\left(3\omega_\xi - \omega_\theta\right)\right]$$

$$F_{14} = \lambda^2\omega_\xi$$

$$F_{21} = -F_{12}$$

$$F_{22} = \frac{1 - \nu}{2}(\gamma b + b') - \frac{\lambda^2 d(1 - \nu)}{8}\left(3\omega_\theta - \omega_\xi\right)\left[2\omega_\xi' - \gamma\left(5\omega_\xi - 3\omega_\theta\right)\right]$$

$$+ \frac{\lambda^2 d'(1 - \nu)}{8}\left(3\omega_\theta - \omega_\xi\right)^2$$

$$F_{23} = \frac{\lambda^2 d(1 - \nu)n}{2\rho}\left[2(1 + \nu)\gamma\omega_\theta - \omega_\xi' + 3\gamma\left(\omega_\xi - \omega_\theta\right)\right] + \frac{\lambda^2 d'(1 - \nu)\left(3\omega_\theta - \omega_\xi\right)n}{2\rho}$$

$$F_{31} = -F_{13}$$

$$F_{32} = \frac{\lambda^2 d(1 - \nu)n}{2\rho}\left[3\gamma\omega_\xi - \gamma\omega_\theta(5 + 2\nu) - \omega_\xi'\right] + \frac{\lambda^2 d'(1 - \nu)n}{2\rho}\left(3\omega_\theta - \omega_\xi\right)$$

$$F_{33} = -\lambda^2 d(1 - \nu)\left[(1 + \nu)\left(2\gamma\omega_\xi\omega_\theta + \gamma^3\right) + \frac{2\gamma n^2}{\rho^2}\right] + \lambda^2 d'(1 - \nu)\left[(1 + \nu)\gamma^2 + \frac{2n^2}{\rho^2}\right]$$

$$F_{34} = \lambda^2\gamma(2 - \nu)$$

$$F_{41} = d\omega_\xi$$

$$F_{43} = -d\nu\gamma$$

$$G_{11} = \nu b'\gamma - \nu b\omega_\theta\omega_\xi - b\gamma^2 - \frac{(1 - \nu)bn^2}{2\rho^2}$$

$$- \lambda^2 d(1 - \nu)\left[(1 + \nu)\gamma^2\omega_\xi^2 + \frac{\left(3\omega_\xi - \omega_\theta\right)^2 n^2}{8\rho^2}\right]$$

$$G_{12} = \frac{\nu n b'}{\rho} - \frac{3 - \nu}{2\rho}\gamma b n - \frac{\lambda^2 d(1 - \nu)\gamma n}{\rho}\left[\frac{\left(3\omega_\xi - \omega_\theta\right)\left(3\omega_\theta - \omega_\xi\right)}{8} + (1 + \nu)\omega_\xi\omega_\theta\right]$$

$$G_{13} = b\left[\omega_\xi' + \gamma\left(\omega_\xi - \omega_\theta\right)\right] + b'\left(\omega_\xi + \nu\omega_\theta\right) - \frac{\lambda^2 d(1 - \nu)\gamma n^2}{\rho^2}\left[\frac{3\omega_\xi - \omega_\theta}{8} + (1 + \nu)\omega_\xi\right]$$

$$G_{14} = \lambda^2(1 - \nu)\gamma\omega_\xi$$

$$G_{21} = -\frac{b\gamma n}{2\rho}(3 - \nu) - \frac{(1 - \nu)nb'}{2\rho} + \frac{\lambda^2 d(1 - \nu)n}{\rho}\left[-(1 + \nu)\gamma\omega_\xi\omega_\theta + \frac{\gamma}{8}\left(6\omega_\xi\omega_\theta - 7\omega_\xi^2 - 3\omega_\theta^2\right) - \frac{\omega_\xi'}{4}\left(5\omega_\theta - 3\omega_\xi\right)\right] - \frac{\lambda^2 d'(1 - \nu)n}{8\rho}\left(3\omega_\xi - \omega_\theta\right)\left(3\omega_\theta - \omega_\xi\right)$$

$$G_{22} = \gamma F_{22} + \left(\frac{1 - \nu}{2}\right)b\omega_\xi\omega_\theta - \frac{bn^2}{\rho^2} - \lambda^2 d(1 - \nu)\left[\frac{1 + \nu}{\rho^2}\omega_\theta^2 n^2 - \frac{\omega_\xi\omega_\theta}{8}\left(3\omega_\theta - \omega_\xi\right)^2\right]$$

$$G_{23} = -\frac{bn\left(\omega_\theta + \nu\omega_\xi\right)}{\rho} + \frac{\lambda^2 dn(1 - \nu)}{2\rho}\left[\gamma\omega_\xi' - 2\gamma^2\omega_\xi - \frac{2(1 + \nu)}{\rho}\omega_\theta n^2 + \left(3\omega_\theta - \omega_\xi\right)\left(\gamma^2 + \omega_\xi\omega_\theta\right)\right] - \frac{\lambda^2 d'n(1 - \nu)\gamma}{2\rho}\left(3\omega_\theta - \omega_\xi\right)$$

$$G_{24} = -\frac{\nu\lambda^2\omega_\theta n}{\rho}$$

$$G_{31} = -b\gamma\left(\omega_\theta + \nu\omega_\xi\right) + \lambda^2 d(1 - \nu)\left[\gamma(1 + \nu)\left(-\gamma\omega_\xi' + \gamma^2\omega_\xi - \frac{n^2\omega_\xi}{\rho^2} + 2\omega_\xi^2\omega_\theta\right) + \frac{n^2}{2\rho^2}\left(\gamma\omega_\xi - \gamma\omega_\theta - 3\omega_\xi'\right)\right] - \lambda^2 d'(1 - \nu)\left[(1 + \nu)\gamma^2\omega_\xi + \frac{n^2}{2\rho^2}\left(3\omega_\xi - \omega_\theta\right)\right]$$

$$G_{32} = -\frac{bn\left(\omega_\theta + \nu\omega_\xi\right)}{\rho} + \frac{\lambda^2 d(1 - \nu)n}{2\rho}\left[2(1 + \nu)\left(\omega_\xi\omega_\theta^2 - \gamma^2\omega_\xi + 2\gamma^2\omega_\theta - \frac{n^2\omega_\theta}{\rho^2}\right) + \gamma\omega_\xi' + 3\gamma^2\left(\omega_\theta - \omega_\xi\right) + \omega_\xi\omega_\theta\left(3\omega_\theta - \omega_\xi\right)\right] - \frac{\lambda^2 d'(1 - \nu)n}{2\rho}\left[2(1 + \nu)\gamma\omega_\theta + \gamma\left(3\omega_\theta - \omega_\xi\right)\right]$$

$$G_{33} = -b\left(\omega_\xi^2 + 2\nu\omega_\xi\omega_\theta + \omega_\theta^2\right) + \frac{\lambda^2 d(1 - \nu)n^2}{\rho^2}\left[(1 + \nu)\left(\omega_\xi\omega_\theta - \frac{n^2}{\rho^2} + 2\gamma^2\right) + 2\left(\gamma^2 + \omega_\xi\omega_\theta\right)\right] - \frac{\lambda^2 d'(1 - \nu)n^2}{\rho^2}(3 + \nu)\gamma$$

$$G_{34} = -\lambda^2\left[(1 - \nu)\omega_\xi\omega_\theta + \frac{\nu n^2}{\rho^2}\right]$$

$$G_{41} = d\left(\omega_\xi' + \nu\gamma\omega_\xi\right)$$

$$G_{42} = \frac{d\nu n\omega_\theta}{\rho}$$

$$G_{43} = \frac{d\nu n^2}{\rho^2}$$

$$G_{44} = -1$$

$$e_1 = -p_\xi + t_T' - \lambda^2(1 - \nu)\gamma\omega_\xi M_T$$

$$e_2 = -p_\theta - \frac{n}{\rho}t_T - \lambda^2(1 - \nu)\frac{n}{\rho}\omega_\theta M_T$$

$$e_3 = -p - \left(\omega_\xi + \omega_\theta\right)t_T - \lambda^2(1 - \nu)\gamma M_T' + \lambda^2(1 - \nu)\left[\omega_\xi\omega_\theta - \frac{n^2}{\rho^2}\right]M_T$$

$$e_4 = M_T$$

where

$$b = \frac{\int E\, d\zeta}{E_o h_o\left(1 - \nu^2\right)}$$

$$d = \frac{\int E\zeta^2\, d\zeta}{E_o h_o^3\left(1 - \nu^2\right)}$$

$$t_T^{(n)} = \frac{\int E\alpha T^{(n)}\, d\zeta}{\sigma_o h_o(1 - \nu)}$$

$$M_T = a \int \frac{\zeta E \alpha T^{(n)} \, d\zeta}{\sigma_o h_o^3 (1 - \nu)}$$

The nonzero components of the H,J,F matrix (ref. 2) are

$$H_{11} = b$$

$$H_{22} = \frac{b(1 - \nu)}{2} + \frac{\lambda^2 d(1 - \nu)}{8}\left(3\omega_\theta - \omega_\xi\right)^2$$

$$H_{23} = \frac{\lambda^2 d(1 - \nu)n}{2\rho}\left(3\omega_\theta - \omega_\xi\right)$$

$$H_{32} = \frac{\lambda^2 d(1 - \nu)n}{2\rho}\left(3\omega_\theta - \omega_\xi\right)$$

$$H_{33} = \lambda^2 d(1 - \nu)\left[\frac{2n^2}{\rho^2} + (1 + \nu)\gamma^2\right]$$

$$H_{34} = \lambda^2$$

$$H_{43} = -1$$

$$J_{11} = \nu\gamma b$$

$$J_{12} = \frac{\nu n b}{\rho}$$

$$J_{13} = b\left(\omega_\xi + \nu\omega_\theta\right)$$

$$J_{21} = -\frac{b(1 - \nu)n}{2\rho} - \frac{d\lambda^2(1 - \nu)n}{8\rho}\left(3\omega_\xi - \omega_\theta\right)\left(3\omega_\theta - \omega_\xi\right)$$

$$J_{22} = -\gamma H_{22}$$

$$J_{23} = -\gamma H_{23}$$

$$J_{31} = -\lambda^2 d(1 - \nu)\left[(1 + \nu)\gamma^2 \omega_\xi + \frac{n^2}{2\rho^2}\left(3\omega_\xi - \omega_\theta\right)\right]$$

$$J_{32} = -\frac{\lambda^2 d(1 - \nu)\gamma n}{2\rho}\left[3\omega_\theta - \omega_\xi + 2(1 + \nu)\omega_\theta\right]$$

$$J_{33} = -\lambda^2 d(1 - \nu)(3 + \nu)\frac{\gamma n^2}{\rho^2}$$

$$J_{34} = \lambda^2 (1 - \nu)\gamma$$

$$J_{41} = \omega_\xi$$

$$F_1 = -t_T$$

$$F_3 = \lambda^2 \gamma (1 - \nu) M_T$$

APPENDIX D

PROGRAM LISTING

The complete listing of the program with both the HGS and SUMUP programs follow. The asterisks on the right-hand edge indicate those statements added to or modified from those given in the HGS program of reference 1.

```
      PROGRAM HGS (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE14)
C     MAIN PROGRAM
C     THIS PROGRAM CONSISTS OF THE MAIN PROGRAM TOGETHER WITH THE FOLLOWING
C     SUBROUTINES
C     MATINV
C     EFG
C     OUTPUT
C     STRESS
C     EQ73
C     INIT
C     FINAL
C     FORCE
C     PANDX
C     ABCG
C     HFJ
C     KLT
C     BDB
C     POLE
C     IN ADDITION TO THE ABOVE THE USER MUST SUPPLY THE FOLLOWING SUB
C     ROUTINES AND FUNCTIONS.
C     SUBROUTINE INPUT--CALCULATES THE SHELL GEOMETRY.
C     FUNCTION P-SPECIFIES THE NORMAL PRESSURE DISTRIBUTION.
C     FUNCTION PX-SPECIFIES THE MERIDIONAL PRESSURE DISTRIBUTION.
C     FUNCTION PT-SPECIFIES THE CIRCUMFERENTIAL PRESSURE DISTRIBUTION
C     FUNCTION TEMP-SPECIFIES THE MERIDIONAL TEMPERATURE DISTRIBUTION
C     FUNCTION DTEMP- SPECIFIES THE DERIVATIVE OF THE TEMPERATURE
C     FUNCTION DELT- SPECIFIES THE DISTRIBUTION OF THE TEMPERATURE
C                   VARIATION THROUGH THE THICKNESS.
C     FUNCTION DDELT- SPECIFIES RATE OF CHANGE OF DELT
C     FUNCTION HHT-SPECIFIES THE SHELL THICKNESS DISTRIBUTION
C     FUNCTION DHHT-SPECIFIES THE DERIVATIVE OF THE THICKNESS.
C     FUNCTION HRA-SPECIFIES THE DISTRIBUTION OF THE RATIO OF THE
C                 TOTAL THICKNESS- TO -COVER PLATE THICKNESS.
C     FUNCTION DHRA- THE DERIVATIVE OF THE ABOVE RATIO.
      INTEGER FREQ,DYNRESP,BOUNDRY, RUNTYPE                            *
      REAL NU,LAM,N,JAY
      COMMON
     3/BL3/NU,LAM,N,EALSIG,CHAR,DEL
     6/BL6/A(4,4),B(4,4),C(4,4),SMAG(4)
     7/BL7/PEE(4,4,102),X(4,102)                                       *
     9/BL9/D(4,4),ZDOT(4,102),ZDDOT(4,102),EPS                        *
     A/BL10/JTIME                                                      *
      COMMON/BL11/ZSAVE(4,4,102)                                       *
     C/BL12/INITZ                                                      *
     6/BL14/AC,TIM,CPS                                                 *
     G/BL16/TAU                                                        *
     H /BL17/KTHTIME                                                   *
     K/BL18/RUNTYPE, JUMP(5)                                           *
      DIMENSION Z(4,102),IPIVOT(4),INDEX(4,2)                          *
      EQUIVALENCE(X(1,1),Z(1,1))                                       *
      NAMELIST/INPUTC/NC,TKN,CHAR,   EALSIG,INC5,                      *
     1NFOUR,NMAX,FREQ,JUMP,NTIME,RUNTYPE,KTHTIME, AK                   *
C RUNTYPE = 1   STATIC CASE                                            *
C RUNTYPE = 2   DYNAMIC RESPONSE--INITIAL DEFLECTIONS                  *
C RUNTYPE = 3   DYNAMIC RESPONSE--INITIAL LOADS                        *
   12 READ (5,INPUTC)                                                  *
      PRINT INPUTC                                                     *
      RT=RUNTYPE                                                       *
      READ(5,31)
   31 FORMAT(72H
     1                )
      WRITE(6,31)
      JTIME=0                                                          *
      N=0.
      LAM=TKN/CHAR
      K=1
```

```
20 FORMAT(1H1,15H TIME ITERATION15/10H REAL TIME3XE15.8,5X*P(1)=*      *
  1    E15.7)                                                          *
30 CONTINUE                                                            *
   IF (JTIME.NE.0) GO TO 17                                            *
   IF (RUNTYPE.EQ.1) GO TO 16                                          *
   IF (RUNTYPE.EQ.3) GO TO 16                                          *
   CALL CALZ (NMAX,IND5)                                               *
   NPRINT=2
   CALL OUTPUT (FREQ,NMAX,DEL,NPRINT)
17 CONTINUE
   IF (JTIME.NE.JUMP(JJJ)) GO TO 16                                    *
   CALL CALZ (NMAX,IND5)                                               *
   JJJ=JJJ+1                                                           *
16 NPRINT=3                                                            *
   IF (RUNTYPE.EQ.2.AND.JTIME.EQ.0) GO TO 19                           *
   CALL HFJ (1,IND5,NMAX,2.)
   CALL EFG(2,IND5,NMAX)
   CALL FORCE(2,IND5,NMAX)
   K=2
   CALL ABCG(K)
   CALL INIT (IND5,BOUNDRY)
   NMAX1=NMAX-1
   DO 6 K=3,NMAX1
   CALL EFC(K,IND5,NMAX)
   CALL FORCE(K,IND5,NMAX)
   CALL ABCG(K)
 6 CALL PANDX(K)
   CALL HFJ(NMAX,IND5,NMAX,2.)
   CALL FINAL (NMAX,IND5,BOUNDRY)
   DO 7 L=2,NMAX1
   K=NMAX+1-L
 7 CALL EG73(K)
   IF(IND5.EQ.0.OR.IND5.EQ.3) GO TO 14
   CALL EG73(1)
   GO TO 15
14 CALL EFC(2,IND5,NMAX)
   CALL FORCE(2,IND5,NMAX)
   K=2
   CALL ABCC(K)
   DO 8 I=1,4
   S1=0.
   S2=0.
   DO 9 J=1,4
   S1=S1+A(I,J)*Z(J,3)
 9 S2=S2+B(I,J)*Z(J,2)
 8 SMAG(I)=SMAG(I)-S1-S2
   CALL MATINV(C,4,SMAG,1,DETERM,IPIVOT,INDEX,4,ISCALE)
   DO 10 I=1,4
10 Z(I,1)=SMAG(I)
15 CALL STRESS(FREQ,NMAX,IND5)
   CALL OUTPUT (FREQ,NMAX,DEL,NPRINT)
   IF (RUNTYPE .NE. 3 ) GO TO 21                                       *
   IF (JTIME .NE. JUMP(JJJ)) GO TO 21                                  *
   INITZ=1                                                             *
   CALL CALZ(NMAX, IND5)                                               *
   NPRINT=2                                                            *
   CALL OUTPUT(FREQ, NMAX, DEL, NPRINT)                                *
   JJJ=JJJ+1                                                           *
21 CONTINUE                                                            *
   BOUNDRY=1                                                           *
   IF ((RUNTYPE .EQ. 1) .AND. (N .EQ. FLOAT(NFOUR))) STOP              *
   IF (RUNTYPE.NE. 1) RUNTYPE=3                                        *
19 CONTINUE                                                            *
   IF (N .EQ. FLOAT(NFOUR)) STOP                                       *
 2 CONTINUE                                                            *
   GO TO 12
   END
```

```
      FHHT=HHT(K,DEL)
      FDHHT= DHHT(K,DEL)
      FHRA= HRA(K,DEL)
      FDHRA=DHRA(K,DEL)
      Q=P(1)
    4 FORMAT(5X, *H/HO= *E12.5,5X*D(H/HO)/DS=*E12.5,5X*H/T=*E12.5,5X    *
     1*D(H/T)/DS=*E12.5)                                                 *
      WRITE(6,4) FHHT, FDHHT, FHRA, FDHRA                                *
    5 FORMAT(5X*THIS RUN WILL SUM FOURIER TERMS FROM N=0 TO *I4)         *
      WRITE(6,5) NFOUR                                                   *
      WRITE(6,81) TIM,CPS                                                *
   81 FORMAT(5X*TIME INCREMENT=*E14.7,5X*CPS=*E14.7)                     *
      REWIND 14                                                          *
      CALL RECOUT(14, 1, 0, FREQ, NMAX, NTIME, NFOUR, AK, TIM, RUNTYPE)  *
      NF1= NFOUR+1                                                       *
      DO 2 IN =1, NF1                                                    *
      N=IN-1                                                             *
      RUNTYPE=RT                                                         *
      IF (RUNTYPE.EQ.1) WRITE(6,32)                                      *
   32 FORMAT(/15H STATIC PROBLEM)                                        *
      IF (RUNTYPE.EQ.2) WRITE(6,33)                                      *
   33 FORMAT(/56H DYNAMIC RESPONSE PROBLEM WITH INITIAL DEFLECTIONS GIVE *
     1N)                                                                 *
      IF (RUNTYPE.EQ.3) WRITE(6,34)                                      *
   34 FORMAT(/75H DYNAMIC RESPONSE PROBLEM - DEFLECTIONS CALCULATED FROM *
     1 INITIAL LOADS GIVEN)                                              *
      DO 3 I=1,4                                                         *
      DO 3 J=1,4                                                         *
    3 D(I,J)=0.                                                          *
      D(1,1)=1.                                                          *
      D(2,2)=1.                                                          *
      D(3,3)=1.                                                          *
      DO 101 K=1,NMAX                                                    *
      DO 101 J=1, 4                                                      *
      PEE(J,3,K)=0.                                                      *
      ZSAVE(J,1,K)=0.                                                    *
      ZSAVE(J,2,K)=0.                                                    *
      ZSAVE(J,3,K)=0.                                                    *
      ZSAVE(J,4,K)=0.                                                    *
      ZDOT(J,K)=0.                                                       *
  101 ZDDOT(J,K)=0.                                                      *
      CALL INPUT (NMAX,NTIME)
      WRITE (6,60)
   60 FORMAT(/11H INPUT DATA)
      WRITE(6,164) NU,TKN,CHAR,N,EALSIG,NMAX,FREQ
  164 FORMAT(/10X26HPOISSONS RATIO        (NU) =E16.8/
     15X31HREFERENCE THICKNESS      (TKN) =E16.8/
     28X28HREFERENCE LENGTH     (CHAR) =E16.8/
     311X25HFOURIER INDEX        (N) =E16.8/
     41X35HTEMPERATURE COEFFICIENT  (EALSIG) =E16.8/
     56X30HNUMBER OF STATIONS     (NMAX) =I16/
     66X30HPRINTING FREQUENCY     (FREQ) =I16)
      IF (RUNTYPE.NE.1) WRITE(6,35) NTIME
   35 FORMAT(4X32HNUMBER OF TIME STEPS    (NTIME) =I16)
      BOUNDRY=0
      IF ( IN .NE.  1) BOUNDRY  =1
      NPRINT=1
      IF (N .EQ. 0.)CALL OUTPUT(FREQ, NMAX, DEL, NPRINT)
      JJJ=1
      DO 19 II=1,NTIME                                                   *
      JTIME=II-1                                                         *
      INITZ=0                                                            *
      Q=P(1)                                                             *
      IF (MOD(JTIME,KTHTIME) .NE. 0) GO TO 30                            *
      WRITE(6,20) JTIME,TAU, Q                                           *
```

```
      SUBROUTINE CALZ (NMAX,IND5)                                          *
      REAL NU,N,LAM                                                        *
      COMMON                                                               *
     1/BL1/R(102),GAM(102),OMT(102),OMXI(102),DECMX(102)                   *
     3/BL3/NU,LAM,N,EALSIG,CHAR,DEL                                        *
     6/BL6/A(4,4),B(4,4),C(4,4),SMAG(4)                                    *
     7/BL7/PEE(4,4,102),X(4,102)                                           *
     9/BL9/D(4,4),ZDOT(4,102),ZDDOT(4,102),EPS                             *
     A/BL10/JTIME                                                          *
     B/BL11/ZSAVE(4,4,102)                                                 *
     C/BL12/INITZ                                                          *
     K/BL13/RUNTYPE, JUMP(5)                                               *
      INTEGER RUNTYPE                                                      *
      IF (RUNTYPE .NE. 2) GO TO 12                                         *
C PEE(1,3,K)=U XI                                                          *
C PEE(2,3,K)=U THETA                                                       *
C PEE(3,3,K)=W                                                             *
C ZDOT(1,K)=U XI DOT                                                       *
C ZDOT(2,K)=U THETA DOT                                                    *
C ZDOT(3,K)=W DOT                                                          *
      DO 1 K=1,NMAX                                                        *
      DO 1 J=1,3                                                           *
      PEE(J,3,K)=ZINIT(J,K)                                                *
    1 ZDOT(J,K)=ZDINIT(J,K)                                                *
      NMAX1=NMAX-1                                                         *
      PRO1=(PEE(1,3,1)+PEE(1,3,2))/2.                                      *
      PRO2=(PEE(2,3,1)+PEE(2,3,2))/2.                                      *
      PRO3=(PEE(3,3,1)+PEE(3,3,2))/2.                                      *
      PRL1=(PEE(1,3,NMAX)+PEE(1,3,NMAX1))/2.                               *
      PRL2=(PEE(2,3,NMAX)+PEE(2,3,NMAX1))/2.                               *
      PRL3=(PEE(3,3,NMAX)+PEE(3,3,NMAX1))/2.                               *
CALCULATION OF M XI                                                        *
      DO 2 K=1,NMAX                                                        *
      IF (K.EQ.1) GO TO 3                                                  *
      IF (K.EQ.NMAX) GO TO 4                                               *
      WDP=(PEE(3,3,K+1)-2.*PEE(3,3,K)+PEE(3,3,K-1))/DEL**2                 *
      WP=(PEE(3,3,K+1)-PEE(3,3,K-1))/(2.*DEL)                              *
      UXIP=(PEE(1,3,K+1)-PEE(1,3,K-1))/(2.*DEL)                            *
      GO TO 5                                                              *
    3 WP=(PEE(3,3,2)-PEE(3,3,1))/DEL                                       *
      UXIP=(PEE(1,3,2)-PEE(1,3,1))/DEL                                     *
      WDP=(3.*PEE(3,3,1)-7.*PEE(3,3,2)+5.*PEE(3,3,3)-PEE(3,3,4))/          *
```

```
      1(2.*DEL**2)                                                      *
       CALL BDB (K,DEL,NU,BLK1,BLK2,DM,BLK3)                            *
       PEE(4,3,1)=DM*(-WDP+OMXI(1)*UXIP+(DEOMX(1)+NU*GAM(1)*OMXI(1))   *
      1*PRO1-NU*GAM(1)*WP+NU*(N/R(1))**2*PRO3+NU*(N/R(1))*OMT(1)*PRO2) *
       PEE(4,3,1)=C.C                                                   *
       GO TO 2                                                          *
     4 WP=(PEE(3,3,NMAX)-PEE(3,3,NMAX-1))/DEL                          *
       UXIP=(PEE(1,3,NMAX)-PEE(1,3,NMAX-1))/DEL                        *
       WDP=(3.*PEE(3,3,NMAX)-7.*PEE(3,3,NMAX-1)+5.*PEE(3,3,NMAX-2)     *
      1 -PEE(3,3,NMAX-3))/(2.*DEL**2)                                   *
       CALL BDB (K,DEL,NU,BLK1,BLK2,DM,BLK3)                            *
       PEE(4,3,NMAX)=DM*(-WDP+OMXI(NMAX)*UXIP+(DEOMX(NMAX)+NU*GAM(NMAX) *
      1*OMXI(NMAX))*PRL1-NU*GAM(NMAX)*WP+NU*(N/R(NMAX))**2*PRL3+NU*    *
      2(N/R(NMAX))*OMT(NMAX)*PRL2)                                      *
       PEE(4,3,NMAX)=C.0                                                *
       GO TO 2                                                          *
     5 CALL BCB (K,DEL,NU,BLK1,BLK2,DM,BLK3)                            *
       PEE(4,3,K)=DM*(-WDP+OMXI(K)*UXIP+(DEOMX(K)+NU*GAM(K))*PEE(1,3,K)*
      1-NU*GAM(K)*WP+NU*(N/R(K))**2*PEE(3,3,K)+NU*(N/R(K))*OMT(K)      *
      2*PEE(2,3,K))                                                     *
     2 CONTINUE                                                         *
       PRO4=PEE(4,3,1)                                                  *
       PRL4=PEE(4,3,NMAX)                                               *
       PEE(4,3,1)=2.*PRO4-PEE(4,3,2)                                    *
       PEE(4,3,NMAX)=2.*PRL4-PEE(4,3,NMAX1)                             *
    12 NMAX1=NMAX-1                                                     *
       IF (RUNTYPE .EC. 2) GO TO 21                                     *
       DO 11 K=2,NMAX1                                                  *
       DO 11 J=1,4                                                      *
    11 ZDOT(J,K)=(11.*ZSAVE(J,4,K)-18.*ZSAVE(J,3,K)+9.*ZSAVE(J,2,K)-2.* *
      1ZSAVE(J,1,K))/(6.*EPS)                                           *
    21 DO 9 K=2,NMAX1                                                   *
       CALL EFG (K,IND5,NMAX)                                           *
       CALL FCRCE (K,IND5,NMAX)                                         *
       CALL ABCG(K)                                                     *
       DO 10 I=1,4                                                      *
       SUM=0.0                                                          *
       DO 20 J=1,4                                                      *
    20 SUM=SUM+A(I,J)*PEE(J,3,K+1)+B(I,J)*PEE(J,3,K)+C(I,J)*PEE(J,3,K-1)*
    10 ZDDOT(I,K)=(SUM-SMAG(I))/(2.*DEL)                                *
       ZDDOT(4,K)=0.0                                                   *
     9 CONTINUE                                                         *
       RETURN                                                           *
       END                                                              *
```

```
      SUBROUTINE FFJ(K,IND5,NMAX,YAH)
C SUBROUTINE FFJ     THIS SUBROUTINE CALCULATES THE ELEMENTS OF THE H,F,
C AND J MATRICES,AS DEFINED IN APPENDIX A OF REFERENCE(1),AT THE STATION
C SPECIFIED BY THE INDEX K.
      COMMON
     1/BL1/R(102),GAM(102),OMT(102),OMXI(102),DEOMX(102)
     3/BL3/NU,LAM,N,EALSIG,CHAR,DEL
     5/BL5/H(4,4),FF(4),JAY(4,4)
      REALNU,LAM,N,JAY,L2
      CALL BDB(K,DEL,NU,B,DB,D,DD)
      L2=LAM**2
      D1=(1.-NU)
      OX=OMXI(K)
      REG=0.
      IF(YAH.EQ.2.) REG=1.
      EAL=EALSIG
      T=TEMP(K,DEL)
      DLT=DELT(K,DEL)
      H1=HHT(K,DEL)
      HRB=HRA(K,DEL)
      FF(1)=-2.*H1*EAL/(D1*HRB)*T
      FF(2)=0.
      FF(3)=L2*CHAR*EAL*DLT*H1**3/3.*(1.5/HRB-3./HRB**2+2./HRB**3)
      FF(4)=0.
      IF(IND5)1,1,2
    2 IF(((IND5-2).LE.0).AND.(K.EQ.1)) GO TO 8
      IF(((IND5-2).GT.0).AND.(K.EQ.NMAX)) GO TO 8
    1 GA=GAM(K)
      FF(3)=FF(3)*GA
      RA=R(K)
      OT=OMT(K)
      ENR=N/RA
      OXT=3.*OMXI(K)-OMT(K)
      OTX=3.*OMT(K)-OMXI(K)
      DL=D*L2*D1*ENR
      H(1,1)=B
      H(1,2)=0.
      H(1,3)=0.
      H(1,4)=0.
      H(2,1)=0.
      H(2,2)=B*D1/2.+L2*D*D1/8.*OTX**2*REG
      H(2,3)=DL/2.*OTX*REG
```

```
      H(2,4)=C.
      H(3,1)=C.
      H(3,2)=DL*CTX*YAH/4.
      ENR2=ENR**2
      H(3,3)=L2*C*D1*(YAH*ENR2+(1.+NU)*GA**2)
      GA2=GA**2
      H(3,4)=L2
      H(4,1)=C.
      H(4,2)=C.
      H(4,3)=-1.
      H(4,4)=C.
      JAY(1,1)=NU*GA*B
      JAY(1,2)=NU*B*ENR
      JAY(1,3)=B*(CX+NU*OT)
      JAY(1,4)=C.
      JAY(2,1)=-B*C1*ENR/2.-DL/8.*CXT*OTX*REG
      JAY(2,2)=-GA*H(2,2)
      JAY(2,3)=-GA*H(2,3)
      JAY(2,4)=C.
      JAY(3,1)=-L2*D*C1*((1.+NU)*GA2*CX+ENR2/4.*OXT*YAH)
      JAY(3,2)=-GA*DL/2.*(     2.*OT*(1.+NU)+OTX/2.*YAH)
      JAY(3,3)=-L2*D*C1*(1.+NU+YAH)*GA*ENR2
      JAY(3,4)=L2*D1*GA
      JAY(4,1)=CX
      JAY(4,2)=C.
      JAY(4,3)=C.
      JAY(4,4)=C.
      GO TO 5
    6 DO 6I=1,4
      DO 6J=1,4
      H(I,J)=0.
    6 JAY(I,J)=0.
      H(1,1)=B*(1.+NU)
      H(1,2)=N*B*NU
      H(2,1)=B*D1*(   -N)/2.
      AWB=D1*(-3.*(1.+NU)+N**2*(3.+NU))
      AW=AWB*C
      C1=AW/(-2.+NU*(N**2-1.))
      ATHETA=1.5*C*N*CX*D1*(3.+NU)
      AXI=1.5*D*CX*C1*(2.*(1.+NU)-N)
      H(3,4)=L2*(2.-NU+2.*AWB)
      H(4,3)=-1.
      JAY(1,3)=B*(1.+NU)*CX
      JAY(4,1)=CX
      DH=CHHT(M,CEL)
      DTOH=DHRA(M,CEL)
      DDLT=DDELT(M,CEL)
      DTMCM=CHAR*EALSIG/3./D1*((1.5/HRB-3./HRB**2+2./HRB**3)*(H1**3*DDLT
     1+3.*DH*H1**2*CLT)+DLT*H1**3*DTOH/HRB**2*(-1.5+6./HRB-6./
     2HRB**2))
      FF(3)=DTMCM*(2.*AWB/(-2.+NU*(N**2-1.))+2.-NU)
    5 RETURN
      END
```

```
      SUBROUTINE OUTPUT (FREQ,NMAX,DEL,NPRINT)
C SUBROUTINE OUTPUT        THIS SUBROUTINE CONTROLS PROGRAM PRINTING AND
C PUNCHING.GEOMETRIC DATA IS PRINTED IF IND1 IS NOT EQUAL TO ZERO.ANY
C OR ALL OF THE ELEVEN OUTPUT QUANTITIES CAN BE PUNCHED BY SUITABLE
C SPECIFICATION OF THE FIELDS OF THE NOL CARD
      COMMON
     1/BL1/R(102),GAM(102),OMT(102),OMXI(102),DEOMX(102)
     7/BL7/PEE(4,4,102),X(4,102)
     9/BL9/D(4,4),ZDOT(4,102),ZDDOT(4,102),EPS
     A/BL10/JTIME
     B/BL11/ZSAVE(4,4,102)
     H /BL17/KTHTIME
      DIMENSION          JJ(11),KK(11),ESS(102),YORD(102)
      EQUIVALENCE   (X(1,1),ESS(1)),(X(1, 27),YORD(1))
      INTEGER FREQ
      GO TO (1000,1001,1001), NPRINT
 1000 WRITE(6,11)
   11 FORMAT(//7X4HR/RB,12X4HZ/RB,12X4HS/RB,8X11HOMEGA THETA,7X8HOMEGA X
     1I,7X10HDEOMEGA XI,8X5HGAMMA/)
      ZED=0.
      S=0.
      DO 8 I=1,NMAX
      IF(I-1)8,8,9
    9 DEM=DEL
      IF((I.EQ.2).OR.(I.EQ.NMAX))DEM=.5*DEL
      S=S+DEM
      ARGU=DEM**2-(R(I)-R(I-1))**2
      IF(ARGU.LE.O.) GO TO 8
      ZED=SQRT(ARGU)+ZED
    8 WRITE(6,12)R(I),ZED,S,OMT(I),OMXI(I),DEOMX(I),GAM(I)
   12 FORMAT(  7E16.8)
      RETURN
 1001 IF (NPRINT.EQ.3) GO TO 51                                          *
      NMAX1=NMAX-1                                                       *
      DO 6 J=1,4                                                         *
      PEE(J,3,1)=.5*(PEE(J,3,1)+PEE(J,3,2))                              *
      PEE(J,3,NMAX)=.5*(PEE(J,3,NMAX1)+PEE(J,3,NMAX))                    *
      ZDOT(J,1)=.5*(ZDOT(J,1)+ZDOT(J,2))                                 *
      ZDOT(J,NMAX)=.5*(ZDOT(J,NMAX1)+ZDOT(J,NMAX))                       *
      ZDDOT(J,1)=.5*(ZDDOT(J,1)+ZDDOT(J,2))                              *
    6 ZDDOT(J,NMAX)=.5*(ZDDOT(J,NMAX1)+ZDDOT(J,NMAX))                    *
      WRITE(6,602)                                                       *
  602 FORMAT(//26H INITIAL DISPLACEMENTS GIVEN//                         *
     14X1HS,8X4HL XI,11X7HU THETA,10X4HM XI,14X1HW/)                     *
      KOUNT=0                                                            *
      DO 603 I=1,NMAX                                                    *
      IF ((I.EQ.1).OR.(I.EQ.2).OR.(I.EQ.NMAX)) GO TO 604                 *
      KOUNT=KOUNT+1                                                      *
      IF (KOUNT-FREQ) 603,605,605                                        *
  605 KOUNT=0                                                            *
  604 WRITE(6,606) ESS(I),PEE(1,3,I),PEE(2,3,I),PEE(4,3,I),PEE(3,3,I)    *
  606 FORMAT(1XF6.3,4E16.8)                                              *
  603 CONTINUE                                                           *
      PRINT 2001                                                         *
 2001 FORMAT(1H1//*ZDOT*//)                                              *
      KOUNT=0                                                            *
      DO 3000 I=1,NMAX                                                   *
      IF ((I.EQ.1).OR.(I.EQ.2).OR.(I.EQ.NMAX)) GO TO 1004                *
      KOUNT=KOUNT+1                                                      *
      IF (KOUNT-FREQ) 3000,1005,1005                                     *
 1005 KOUNT=0                                                            *
 1004 PRINT 1007, (ZDOT(K,I),K=1,4)                                      *
 1007 FORMAT(4E20.8)                                                     *
 3000 CONTINUE                                                           *
      PRINT 1008                                                         *
 1008 FORMAT(///* ZDDOT*//)                                              *
      KOUNT=0                                                            *
      DO 2000 I=1,NMAX                                                   *
      IF ((I.EQ.1).OR.(I.EQ.2).OR.(I.EQ.NMAX)) GO TO 2004                *
```

```
      KOUNT=KOUNT+1                                                     *
      IF (KOUNT-FREQ) 2000,2005,2005                                    *
 2005 KOUNT=0                                                           *
 2004 PRINT 1007,(ZDDOT(K,I),K=1,4)                                     *
 2000 CONTINUE                                                          *
      GO TO 600
   51 KOUNT=0
      DO 33 I=1,NMAX
      IF((I.EQ.1).OR.(I.EQ.NMAX)) GO TO 103
   14 PEE(4,3,I)=X(4,I)
      PEE(1,3,I)=X(1,I)
      PEE(2,3,I)=X(2,I)
      PEE(3,3,I)=X(3,I)
      GO TO 33
  103 IF(I-1) 104,104,105
  104 K=2
      J=1
      GO TO 106
  105 K=NMAX
      J=NMAX
  106 PEE(4,3,J)=.5*(X(4,K)+X(4,K-1))
      PEE(1,3,J)=.5*(X(1,K)+X(1,K-1))
      PEE(2,3,J)=.5*(X(2,K)+X(2,K-1))
      PEE(3,3,J)=.5*(X(3,K)+X(3,K-1))
   33 CONTINUE
  600 JSAVE=JTIME+1                                                     *
      IF (JSAVE.GT.5) JSAVE=5                                           *
      GO TO (1,4,77,10,100), JSAVE                                      *
    1 DO 200 K=1,NMAX                                                   *
      DO 200 J=1,4                                                      *
  200 ZSAVE(J,1,K)=PEE(J,3,K)                                           *
      GO TO 501                                                         *
    4 DO 210 K=1,NMAX                                                   *
      DO 210 J=1,4                                                      *
  210 ZSAVE(J,2,K)=PEE(J,3,K)                                           *
      GO TO 501                                                         *
   77 DO 22 K=1,NMAX                                                    *
      DO 22 J=1,4                                                       *
   22 ZSAVE(J,3,K)=PEE(J,3,K)                                           *
      GO TO 501                                                         *
   10 DO 23 K=1,NMAX                                                    *
      DO 23 J=1,4                                                       *
   23 ZSAVE(J,4,K)=PEE(J,3,K)                                           *
      GO TO 501                                                         *
  100 DO 24 K=1,NMAX                                                    *
      DO 24 J=1,4                                                       *
      ZSAVE(J,1,K)=ZSAVE(J,2,K)                                         *
      ZSAVE(J,2,K)=ZSAVE(J,3,K)                                         *
      ZSAVE(J,3,K)=ZSAVE(J,4,K)                                         *
   24 ZSAVE(J,4,K)=PEE(J,3,K)                                           *
  501 IF (NPRINT.EQ.2) GO TO 13                                         *
      IF (MOD(JTIME,KTHTIME) .NE. 0) GO TO 1003                         *
      NOPTS=(NMAX-2)/FREQ+2
      SMAX=DEL*FLOAT(NMAX-2)
      WRITE(6,53)SMAX
   53 FORMAT(/9H SMAX/RB=E16.8)
      ESS(1)=0.
      DO 20 I=2,NMAX
      DEM=DEL
      IF((I.EQ.2).OR.(I.EQ.NMAX)) DEM=DEL/2.
   20 ESS(I)= ESS(I-1)+DEM /SMAX
  601 WRITE(6,2)
    2 FORMAT(/4X1HS,8X4HN XI,11X7HN THETA,10X4HN XT,12X4HQ XI,12X4HM XI,
     111X7HM THETA/)
      KOUNT=0
      DO 3 I=1,NMAX
      IF((I.EQ.1).OR.(I.EQ.2).OR.(I.EQ.NMAX)) GO TO 17
```

```
      KOUNT=KCUNT+1
      IF(KOUNT-FREQ) 3,18,18
   18 KOUNT=0
   17 WRITE(6,7) ESS(I),PEE(1,2,I),PEE(2,1,I),PEE(2,2,I),PEE(3,2,I),
     1PEE(4,3,I),PEE(1,1,I)
    7 FORMAT(1XF6.3,6E16.8)
    3 CONTINUE
      WRITE(6,201)
  201 FORMAT(/4X1HS,8X4HM XT,12X4HU XI,11X7HU THETA,12X1HW,12X6HPHI XI/)
      KOUNT=0
      DO 202 I=1,NMAX
      IF ((I.EQ.1).CR.(I.EQ.2).OR.(I.EQ.NMAX)) GO TO 203
      KOUNT=KCUNT+1
      IF (KOUNT-FREC) 202,204,204
  204 KOUNT=0
  203 WRITE(6,205) FSS(I),PEE(3,1,I),PEE(1,3,I),PEE(2,3,I),PEE(3,3,I),
     1PEE(4,2,I)
  205 FORMAT(1XF6.3,5E16.8)
  202 CONTINUE
 1003 CONTINUE
  500 M=0
      JJ(1) =1
      JJ(2) =2
      JJ(3) =2
      JJ(4) =3
      JJ(5) =4
      JJ(6) =1
      JJ(7) =3
      JJ(8) =1
      JJ(9) =2
      JJ(10)=3
      JJ(11)=4
      KK(1) =2
      KK(2) =1
      KK(3) =2
      KK(4) =2
      KK(5) =3
      KK(6) =1
      KK(7) =1
      KK(8) =3
      KK(9) =3
      KK(10)=3
      KK(11)=2
      KKK=11
      DC 50 L=1,KKK
      J=JJ(L)
      K=KK(L)
      KOUNT=0
      DO 19 I=1,NMAX
      M=I
      IF((I.EQ.1).CR.(I.EQ.2)) GO TO 19
      M=(NMAX-2)/FREC+2
      IF(I.EQ.NMAX) GC TO 19
      KOUNT=KCUNT+1                                                    *
      IF(KOUNT-FREQ)19,21,21                                           *
   21 KCUNT=0
      M=(I-2)/FREC+2
   19 YORD(M)=PEE(J,K,I)
      N1=(NMAX-2)/FREC+2
  101 FORMAT(5E12.5,4X2I4)
      DO 102 I=1,N1                                                    *
       ABC=YCRD(I)                                                     *
      CALL RECOUT(14, 1, C, ABC)                                       •
  102 CONTINUE                                                         *
   50 CONTINLE
   13 RETURN
      END
```

```
      SUBROUTINE EFG(K,IND5,NMAX)
C SUBROUTINE EFG    THIS SUBROUTINE CALCULATES THE ELEMENTS OF THE E,F,AND G
C MATRICES,AS DEFINED IN APPENDIX A OF REFERENCE (1),AT THE STATION SPECIFIED
C BY THE INDEX K.
      COMMON
     1/BL1/R(1C2),GAM(102),OMT(1C2),OMXI(102),DEOMX(102)
     2/BL2/E(4,4),G(4,4),F(4,4)
     3/BL3/NU,LAM,N,EALSIG,CHAR,DEL
      REAL  NU,LAM,N,LAM2
      CALL BDB(K,DEL,NU,B,DB,D,DD)
      E(1,1)=B
      E(1,2)=0.
      E(1,3)=C.
      E(1,4)=0.
      E(2,1)=C.
      D1=(1.-NU)
      LAM2=LAM**2
      RA=R(K)
      GA=GAM(K)
      OX=CMXI(K)
      OT=OMT(K)
      DEX=DECMX(K)
      GA2=GA**2
      REX=(3.*OT-CX)
      RXE=(3.*CX-CT)
      OTX=OT*OX
      DNLR=LAM2*D*N*D1/(2.*RA)
      DDNLR=DNLR*DD/D
      E(2,2)=B*D1/2.+LAM2*D*D1*REX**2/8.
      E(2,3)=DNLR*REX
      E(2,4)=C.
      E(3,1)=0.
      E(3,2)=E(2,3)
      RAN=(N/RA)**2
      E(3,3)=LAM2*D*D1*(2.*RAN+(1.+NU)*GA2)
      E(3,4)=LAM2
      E(4,1)=C.
      E(4,2)=C.
      E(4,3)=-D
      E(4,4)=C.
      F(1,1)=GA*B+DB
      F(1,2)=(1.+NU)*B*N/(2.*RA)+DNLR*REX*RXE/4.
      F(1,3)=B*(CX+NU*OT)+LAM2*D*D1*((1.+NU)*GA2*OX+RAN*RXE/2.)
      F(1,4)=LAM2*CX
      F(2,1)=-F(1,2)
      F(2,2)=(D1/2.)*(GA*B+DB)-(LAM2*D*D1*REX/8.)*(2.*DEX-GA*(5.*CX
     1-3.*OT))+LAM2*DD*D1*REX**2/8.
      F(2,3)=DNLR*(2.*(1.+NU)*GA*OT-DEX+3.*GA*(OX-OT))+DDNLR*REX
      F(2,4)=0.
      F(3,1)=-F(1,3)
      F(3,2)=DNLR*(3.*GA*OX-GA*OT*(5.+2.*NU)-DEX)+DDNLR*REX
      F(3,3)=-LAM2*D*D1*((1.+NU)*(2.*GA*OX*OT+GA**3)+2.*GA*RAN)
     1+LAM2*DD*D1*((1.+NU)*GA2+2.*RAN)
      F(3,4)=LAM2*GA*(2.-NU)
      F(4,1)=D*CX
      F(4,2)=0.
      F(4,3)=-D*NU*GA
      F(4,4)=C.
      G(1,1)=NU*DB*GA-NU*B*OTX-B*GA2-D1*B*RAN/2.-LAM2*D*D1*((1.+NU)*GA2*
     1OX**2+RXE**2*RAN/8.)
      G(1,2)=NU*N*DB/RA-(3.-NU)/(2.*RA)*GA*B*N-DNLR*2.*GA*(REX*RXE/8.
     1+(1.+NU)*OTX)
      G(1,3)=B*(DEX+GA*(OX-OT))+DB*(OX+NU*OT)-LAM2*D*D1*GA*RAN*(RXE/2.+(
     11.+NU)*CX)
      G(1,4)=LAM2*D1*GA*OX
      G(2,1)=-B*GA*N*(3.-NU)/(2.*RA)-D1*N*DB/(2.*RA)+DNLR*2.*(-1.*(1.+
     1NU)*GA*OTX+GA/8.*(6.*OTX-7.*OX**2-3.*OT**2)-DEX/4.*(5.*OT-3.*OX))
```

```
 2-DDNLR/4.*REX*RXE
  G(2,2)=-GA*F(2,2)+C1/2.*B*OTX-B*RAN-LAM2*D*D1*((1.+NU)*OT**2*RAN
 1-OTX/8.*REX**2)
  G(2,3)=-B*N*(CT+NU*CX)/RA+DNLR*(GA*DEX-2.*GA2*OX-2.*(1.+NU)*OT
 1*RAN+REX*(GA2+CTX))-DDNLR*REX*GA
  G(2,4)=-NU*LAM2*OT*N/RA
  G(3,1)=-B*GA*(CT+NU*OX)+LAM2*D*D1*(GA*(1.+NU)*(-GA*DEX+GA2*OX
 1-OX*RAN+2.*CTX*CX)+RAN/2.*(GA*OX-GA*OT-3.*DEX))
 2-LAM2*CD*D1*((1.+NU)*GA2*OX+RAN/2.*RXE)
  G(3,2)=-B*N*(CT+NU*OX)/RA+DNLR*(2.*(1.+NU)*(CTX*OT-GA2*OX+2.*GA2
 1*OT-OT*RAN)+GA*DEX+3.*GA2*(OT-OX)+OTX*REX)-DDNLR*(2.*(1.+NU)*GA
 2*CT+GA*REX)
  G(3,3)=-B*(CX**2+2.*NU*OTX+OT**2)+LAM2*D*D1*RAN*((1.+NU)*(OTX-RAN
 1+2.*GA2)+2.*(GA2+OTX))-LAM2*CD*D1*RAN*(3.+NU)*GA
  G(3,4)=-LAM2*(D1*CTX+NU*RAN)
  G(4,1)=C*(DEX+NU*GA*OX)
  G(4,2)=C*NL*N*CT/RA
  G(4,3)=C*NL*RAN
  G(4,4)=-1.
  RETURN
  END
```

```
      SUBROUTINE FORCE(K,IND5,NMAX)
C SUBROUTINE FORCE  THIS SUBROUTINE CALCULATES THE ELEMENTS OF THE LOWER CASE
C E-VECTOR AS DEFINED IN APPENDIX A OF REFERENCE(1),AT THE STATION SPECIFIED
C BY THE INDEX K.
      COMMON
     1/BL1/R(102),GAM(102),OMT(102),OMXI(102),DEOMX(102)
     3/BL3/NU,LAM,N,EALSIG,CHAR,DEL
     4/BL4/CEE(4)
      REAL NU,LAM,N,L2
      REAL MSTP
      RA=R(K)
      GA=GAM(K)
      OX=OMXI(K)
      OT=OMT(K)
      T=TEMP(K,CEL)
      DT=DTEMP(K,CEL)
      DELT1=DELT(K,CEL)
      DLT1=DDELT(K,CEL)
      PX1=PX(K,CEL)
      PT1=PT(K,CEL)
      P1=P(K)
      H=HHT(K,CEL)
      DH=DHHT(K,CEL)
      HRB=HRA(K,CEL)
      DHRB=DHRA(K,CEL)
      D1=1.-NU
      L2=LAM**2
      EAL=EALSIG
      TSUBT=2.*H*EAL/(D1*HRB)*T
      MSTP=CHAR*EAL/(3.*D1)*((1.5/HRB-3./HRB**2+2./HRB**3)*(DLT1*H**3+3.      *
     1*H**2*DH*DELT1)+DELT1*H**3*DHRB/HRB**2*(-1.5+6./HRB-6./HRB**2))          *
      CEE(4)=CHAR*EAL*DELT1*H**3/(3.*D1)*(1.5/HRB-3./HRB**2+2./HRB**3)          *
      CEE(1)=-PX1+2.*EAL/(D1*HRB)*(H*DT+T*DH)-DHRB/HRB*TSUBT-                   *
     1L2*D1*GA*CX*CEE(4)                                                        *
      CEE(2)=-PT1-N/RA*TSUBT-L2*D1*N/RA*OT*CEE(4)                               *
      CEE(3)=-P1-(CX+OT)*TSUBT-L2*D1*GA*MSTP+                                   *
     1L2*D1*CEE(4)*(CX*OT-(N/RA)**2)                                            *
      RETURN
      END
```

```
      SUBROUTINE ABCG (K)
C     SUBROUTINE ABCG-- THIS SUBROUTINE CALCULATES THE A,B,C, AND LOWER
C     CASE G MATRICES  USING THE CURRENT VALUES OF THE E,F,G, AND LOWER CASE
C     E MATRICES.
      COMMON
     2/BL2/E(4,4),G(4,4),F(4,4)
     3/BL3/NU,LAM,N,EALSIG,CHAR,DEL
     4/BL4/CEE(4)
     6/BL6/A(4,4),B(4,4),C(4,4),SMAG(4)
     9/BL9/D(4,4),ZDOT(4,102),ZDDOT(4,102),EPS
     A/BL10/JTIME
     B/BL11/ZSAVE(4,4,102)
      DIMENSION TERM(4)
      REAL N,NU,LAM
      D2=2./DEL
      D4=4./DEL
      DX=2.*DEL
      DO1I=1,4
      SMAG(I)=DX*CEE(I)
      DO1J=1,4
      B(I,J)=-D4*E(I,J)+DX*G(I,J)
      C(I,J)=D2*E(I,J)-F(I,J)
    1 A(I,J)=D2*E(I,J)+F(I,J)
      JJ=JTIME+1
      IF (JJ.GT.5) JJ=5
      GO TO (2,3,4,5,6), JJ
    3 DO 14 J=1,4
   14 TERM(J)=-6.*ZSAVE(J,1,K)-6.*EPS*ZDOT(J,K)-2.*EPS**2*ZDDOT(J,K)
      DO 15 I=1,4
      SUM=0.0
      DO 25 J=1,4
   25 SUM=SUM+D(I,J)*TERM(J)
   15 SMAG(I)=SMAG(I)+SUM*DX/EPS**2
      DO 16 I=1,4
      DO 16 J=1,4
   16 B(I,J)=B(I,J)-6.*DX*D(I,J)/EPS**2
    2 RETURN
    4 DO 17 J=1,4
   17 TERM(J)=-4.*ZSAVE(J,2,K)+2.*ZSAVE(J,1,K)-EPS**2*ZDDOT(J,K)
      DO 18 I=1,4
      SUM=0.0
      DO 26 J=1,4
   26 SUM=SUM+D(I,J)*TERM(J)
   18 SMAG(I)=SMAG(I)+SUM*DX/EPS**2
      GO TO 19
    5 DO 20 J=1,4
   20 TERM(J)=-5.*ZSAVE(J,3,K)+4.*ZSAVE(J,2,K)-ZSAVE(J,1,K)
      DO 21 I=1,4
      SUM=0.0
      DO 27 J=1,4
   27 SUM=SUM+D(I,J)*TERM(J)
   21 SMAG(I)=SMAG(I)+SUM*DX/EPS**2
      GO TO 19
    6 DO 22 J=1,4
   22 TERM(J)=-5.*ZSAVE(J,4,K)+4.*ZSAVE(J,3,K)-ZSAVE(J,2,K)
      DO 23 I=1,4
      SUM=0.0
      DO 28 J=1,4
   28 SUM=SUM+D(I,J)*TERM(J)
   23 SMAG(I)=SMAG(I)+SUM*DX/EPS**2
   19 DO 24 I=1,4
      DO 24 J=1,4
   24 B(I,J)=B(I,J)-2.*DX*D(I,J)/EPS**2
      RETURN
      END
```

```
      SUBROUTINE INIT (INC5,BCUNDRY)
C     CALCULATICN OF THE P-MATRIX AND THE X-VECTOR AT THE FIRST STATION.
      INTEGER BCUNCRY
      CCMMON
     2/BL2/E(4,4),G(4,4),F(4,4)
     3/BL3/NU,LAM,N,EALSIG,CHAR,DEL
     4/BL4/CEE(4)
     5/BL5/H(4,4),FF(4),JAY(4,4)
     6/BL6/A(4,4),B(4,4),C(4,4),SMAG(4)
     7/BL7/PEF(4,4,102),X(4,102)
      DIMENSICN CMEG1(4,4),CAPL1(4,4),EL1(4)
      DIMENSICN IPIVCT(4),INDEX(4,2),A0(4,4),B0(4,4),A3(4,4),A4(4,4),G0(
     14)
      NAMELIST/FIRST/CMEG1,CAPL1,EL1
      REAL JAY,N,NU,LAM
   90 DO 91 I=1,4
   91 G0(I)=0.
      IF (IND5) 13,13,10
   10 IF (INC5-2) 11,11,13
   11 CALL PCLE(N,DEL,F,G)
      IF (BOUNDRY.EC.1) GC TC 16
      WRITE(6,166)
  166 FCRMAT(/27H BCUNCARY CCNDITICNS AT S=0)
      WRITE(6,40)
   40 FORMAT(38H CCNDITICNS FCR A SHELL PCLE GENERATED)
      GO TO 16
   13 IF (BOUNDRY.EC.1) GO TO 70
      DO 41 I=1,4
      EL1(I)=C.C
      DO 41 J=1,4
      CMEG1(I,J)=C.C
   41 CAPL1(I,J)=0.C
      READ(5,FIRST)
   15 WRITE(6,166)
      WRITE(6,167)
  167 FORMAT(/21X5HCMEGA,49X5HLAMDA,34X3HELL/)
      DO 168 I=1,4
  168 WRITE(6,169)CMEG1(I,1),CMEG1(I,2),OMEG1(I,3),OMEG1(I,4),
     1            CAPL1(I,1),CAPL1(I,2),CAPL1(I,3),CAPL1(I,4),
     2            EL1(I)
  169 FORMAT(4E12.4,2(6X4E12.4))
   70 DO 1 I=1,4
      DO 1 J=1,4
      B0(I,J)=H(I,J)/DEL+JAY(I,J)/2.
    1 A0(I,J)=JAY(I,J)/2.-H(I,J)/DEL
      DC 2 I=1,4
      DO 2 J=1,4
      S1=0.
      S2=0.
      DO 3 L=1,4
      S1=S1+CMEG1(I,L)*A0(L,J)
```

```
  3 S2=S2+OMEG1(I,L)*BO(L,J)
    A3(I,J)=S1
  2 A4(I,J)=S2
    DO 5 I=1,4
    S1=0.
    DO 4 J=1,4
    S1=S1+OMEG1(I,J)*FF(J)
    BO(I,J)=A3(I,J)+CAPL1(I,J)/2.
  4 AO(I,J)=A4(I,J)+CAPL1(I,J)/2.
  5 GO(I)=EL1(I)-S1
    CALL MATINV (C,4,CEE,0,DETERM,IPIVOT,INDEX,4,ISCALE)
    DO 50 I=1,4
    DO 50 J=1,4
    S1=0.
    S2=0.
    S3=0.
    DO 51 K=1,4
    S1=S1+C(I,K)*B(K,J)
    S2=S2+C(I,K)*A(K,J)
 51 S3=S3+BC(I,K)*C(K,J)
    A3(I,J)=S1
    A4(I,J)=S2
 50 E(I,J)=S3
    DO 52 I=1,4
    DO 52 J=1,4
    S1=0.
    S2=0.
    DO 53 K=1,4
    S1=S1+BO(I,K)*A3(K,J)
 53 S2=S2+BO(I,K)*A4(K,J)
    F(I,J)=S1-AC(I,J)
 52 G(I,J)=S2
 16 CALL MATINV(F,4,G,4,DETERM,IPIVOT,INDEX,4,ISCALE)
    IF(IND5.EQ.0.OR.IND5.EQ.3) GO TO 59
    DO 60 I=1,4
    X(I,1)=0.
    DO 60 J=1,4
 60 PEE(I,J,1)=G(I,J)
    CALL PANDX(2)
    RETURN
 59 DO 54 I=1,4
    S1=0.
    DO 61 J=1,4
    PEE(I,J,2)=G(I,J)
 61 S1=E(I,J)*SMAG(J)+S1
 54 GO(I)=S1-GC(I)
    DO 56 I=1,4
    S1=0.
    DO 57 J=1,4
 57 S1=S1+F(I,J)*GC(J)
 56 X(I,2)=S1
    RETURN
    END
```

```
      SUBROUTINE PANCX(K)
C SUBROUTINE PANCX  THIS SUBROUTINE CALCULATES THE P MATRIX AND THE X-VECTOR
C AT THE STATION SPECIFIED BY THE INDEX K USING EQUATION (29) OF THE TEXT.
      COMMON
     6/BL6/A(4,4),B(4,4),C(4,4),SMAG(4)
     7/BL7/PEE(4,4,102),X(4,102)
      DIMENSIONP1(4,4),IPIVOT(4),INDEX(4,2),X1(4),X2(4)
      DO 1 I=1,4
      DO 1 J=1,4
      SUM=0.
      DO 2 L=1,4
    2 SUM=SUM+C(I,L)*PEE(L,J,K-1)
    1 P1(I,J)=B(I,J)-SUM
      CALLMATINV(P1,4,X2,0,DETERM,IPIVOT,INDEX,4,ISCALE)
      DO 4 I=1,4
      SUM=0.
      DO 3 J=1,4
    3 SUM=SUM+C(I,J)*X(J,K-1)
    4 X1(I)=SMAG(I)-SUM
      DO 5 I=1,4
      DO 5 J=1,4
      SUM=0.
      DO 6 L=1,4
    6 SUM=SUM+P1(I,L)*A(L,J)
    5 PEE(I,J,K)=SUM
      DO 7 I=1,4
      SUM=0.
      DO 8 J=1,4
    8 SUM=SUM+P1(I,J)*X1(J)
    7 X(I,K)=SUM
      RETURN
      END
```

```
      SUBROUTINE EQ73(K)
C SUBROUTINE EQ73   THIS SUBROUTINE CALCULATES THE SOLUTION VECTOR AT
C THE STATION(K),GIVEN THE SOLUTION AT K+1.
      COMMON
     7/BL7/PEE(4,4,102),X(4,102)
      DIMENSION Z(4,102)
      EQUIVALENCE (X(1,1),Z(1,1))
      DO 1 I=1,4
      SUM=0.
      DO 2 J=1,4
    2 SUM=SUM+PEE(I,J,K)*Z(J,K+1)
    1 Z(I,K)=X(I,K)-SUM
      RETURN
      END
```

```
      SUBROUTINE FINAL (NMAX,IND5,BOUNDRY)
C     CALCULATION OF SOLUTION VECTOR ASSOCIATED WITH LAST STATION
      INTEGER BOUNDRY
      COMMON
     3/BL3/NU,LAM,N,EALSIG,CHAR,DEL
     5/BL5/H(4,4),FF(4),JAY(4,4)
     7/BL7/PEE(4,4,1C2),X(4,1C2)
      DIMENSION OMEGL(4,4),CAPLL(4,4),ELL(4)
      DIMENSION IPIVOT(4),INDEX(4,2),A1(4,4),A2(4,4),A3(4,4),PSI(4,4),
     1GM(4,4),ETA(4),B(4,4)
      REAL JAY,N,NU,LAM
      NAMELIST/LAST/OMEGL,CAPLL,ELL
   90 IF (IND5) 13,13,10
   10 IF (IND5-2) 13,11,11
   11 CALL POLE (N,DEL,PSI,GM)
      IF (BOUNDRY.EQ.1) GO TO 16
      WRITE(6,17C)
  170 FORMAT(/30H BOUNDARY CONDITIONS AT S=SMAX)
      WRITE(6,4C)
   40 FORMAT(38H CONDITIONS FOR A SHELL POLE GENERATED)
      GO TO 16
   13 IF (BOUNDRY.EQ.1) GO TO 70
      DO 41 I=1,4
      ELL(I)=0.0
      DO 41 J=1,4
      OMEGL(I,J)=0.C
   41 CAPLL(I,J)=C.C
      READ (5,LAST)
   15 WRITE(6,17C)
      WRITE(6,167)
  167 FORMAT(21X5HOMEGA,49X5HLAMDA,34X3HELL/)
      DO 171 I=1,4
  171 WRITE(6,169)OMEGL(I,1),OMEGL(I,2),OMEGL(I,3),OMEGL(I,4),
     1              CAPLL(I,1),CAPLL(I,2),CAPLL(I,3),CAPLL(I,4),
     2              ELL(I)
  169 FORMAT(4E12.4,2(6X4E12.4))
   70 DO 1 I=1,4
      DO 1J=1,4
      A1(I,J)=JAY(I,J)/2.+H(I,J)/DEL
    1 A2(I,J)=JAY(I,J)/2.-H(I,J)/DEL
      DO 2 I=1,4
      DO 2 J=1,4
      S2=0.
      S3=0.
      DO 3 L=1,4
      S2=OMEGL(I,L)*A1(L,J)+S2
    3 S3=OMEGL(I,L)*A2(L,J)+S3
      PSI(I,J)=S3+CAPLL(I,J)/2.
    2 GM(I,J)=S2+CAPLL(I,J)/2.
   16 DO 4 I=1,4
      DO 4 J=1,4
      S1=0.
      DO 5 L=1,4
    5 S1=S1+PSI(I,L)*PEE(L,J,NMAX-1)
    4 B(I,J)=GM(I,J)-S1
      DO 6 I=1,4
      S1=0.
      S2=0.
      DO 7 J=1,4
      S1=S1+PSI(I,J)*X(J,NMAX-1)
    7 S2=S2+OMEGL(I,J)*FF(J)
    6 ETA(I)=ELL(I)-S1-S2
      CALL MATINV(B,4,ETA,1,DETERM,IPIVOT,INDEX,4,ISCALE)
      DO 12 I=1,4
   12 X(I,NMAX)=ETA(I)
      RETURN
      END
```

```
      SUBROUTINE STRESS(FREQ,NMAX,  IND5)
C     SUBROUTINE STRESS-- THIS SUBROUTINE CALCULATES THE SECONDARY QUANTITIES
C     N XI, N XI THETA, Q XI, PHI, M THETA, M XI THETA, M XI THETA, AND
C     N THETA AT EACH STATION ALONG THE SHELL.
      COMMON
     1/BL1/R(102),GAM(102),OMT(102),OMXI(102),DECMX(102)
     3/BL3/NU,LAM,N,EALSIG,CHAR,DEL
     5/BL5/H(4,4),FF(4),JAY(4,4)
     7/BL7/PEE(4,4,102),X(4,102)
     8/BL8/AK(3,4),ALL(3,4),STHER(3)
      DIMENSION Z(4,102),Y(4),DZ(4)
      EQUIVALENCE                         (Z(1,1),X(1,1))
      INTEGER FREQ
      REAL N,NU,JAY,LAM
      KOUNT=0
      DO 9 I=1,NMAX
      IF((I.EQ.1).OR.(I.EQ.NMAX)) GO TO 1
      IF(I-2) 13,13,14
   14 KOUNT=KOUNT+1
      IF(KOUNT-FREQ)9,12,12
   12 KOUNT=0
   13 DO 3 L=1,4
      Y(L)=Z(L,I)
    3 DZ(L)=(Z(L,I+1)-Z(L,I-1))/2./DEL
      GO TO 2
    1 IF(I-1)4,4,5
    4 K=2
      GO TO 6
    5 K=NMAX
    6 DO 11 L=1,4
      Y(L)=.5*(Z(L,K)+Z(L,K-1))
   11 DZ(L)=(Z(L,K)-Z(L,K-1))/DEL
    2 CALL HFJ(I,IND5,NMAX,1.)
      CALL KLT(I,IND5,NMAX)
      DO 7 L=1,4
      SUM1=0.
      SUM2=0.
      DO8M=1,4
      SUM1=SUM1+ H(L,M)*DZ(M)
    8 SUM2=SUM2+JAY(L,M)*Y(M)
    7 PEE(L,2,I)=SUM1+SUM2+FF(L)
      DO 9 L=1,3
      SUM3=0.
      SUM4=0.
      DO 10 M=1,4
      SUM3=SUM3+AK(L,M)*DZ(M)
   10 SUM4=SUM4+ALL(L,M)*Y(M)
      PEE(L,1,I)=SUM3+SUM4+STHER(L)
    9 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE KLT(K,IND5,NMAX)
C SUBROUTINE KLT     THIS SUBROUTINE CALCULATES THE ELEMENTS OF THE MATRICES
C WHICH ALLOW THE CALCULATION OF THE QUANTITIES M-THETA,N-THETA,AND M-XI THETA
C AT THE STATION SPECIFIED BY THE INDEX K.
      COMMON
     1/BL1/R(102),GAM(102),OMT(102),OMXI(102),DEOMX(102)
     3/BL3/NU,LAM,N,EALSIG,CHAR,DEL
     8/BL8/AK(3,4),ALL(3,4),STHER(3)
      REAL NU,LAM,N
      CALL BDB(K,DEL,NU,B,DB,D,DD)
      D1=D*(1.-NU)
      D2=D*(1.-NU**2)
      OX=OMXI(K)
      EAL=EALSIG
      H=HHT(K,DEL)
      HRB=HRA(K,DEL)
      TEMPER=TEMP(K,DEL)
      DELTAT=DELT(K,DEL)
      STHER(1)=-CHAR*EAL*DELTAT *H**3*(1./(2.*HRB)-1./HRB**2
     1+2./(3.*HRB**3))
      STHER(2)=-2.*H*EAL*TEMPER /((1.-NU)*HRB)
      STHER(3)=0.
      IF(IND5)1,1,2
    2 IF(((IND5-2).LE.0).AND.(K.EQ.1)) GO TO 8
      IF(((IND5-2).GT.0).AND.(K.EQ.NMAX)) GO TO 8
    1 RA=R(K)
      GA=GAM(K)
      OT=OMT(K)
      REX=3.*OT-OX
      RXE=3.*OX-OT
      RAN=N/RA
      RAN2=RAN**2
      AK(1,1)=0.
      AK(1,2)=0.
      AK(1,3)=-GA*D2
      AK(1,4)=0.
      AK(2,1)=B*NU
      AK(2,2)=0.
      AK(2,3)=0.
      AK(2,4)=0.
      AK(3,1)=0.
      AK(3,2)=D1*REX/4.
      AK(3,3)=RAN*D1
      AK(3,4)=0.
      ALL(1,1)=GA*OX*D2
      ALL(1,2)=D2*RAN*OT
      ALL(1,3)=D2*RAN2
      ALL(1,4)=NU
      ALL(2,1)=B*GA
      ALL(2,2)=B*RAN
      ALL(2,3)=B*(OT+NU*OX)
      ALL(2,4)=0.
      ALL(3,1)=-D1*RAN*RXE/4.
      ALL(3,2)=-GA*D1*REX/4.
      ALL(3,3)=-GA*RAN*D1
      ALL(3,4)=0.
      GO TO 6
    8 DO 3 I=1,3
      DO 3 J=1,4
      AK(I,J)=0.
    3 ALL(I,J)=0.
      C1=(N**2/2.-1.)/(1.+NU-N**2*NU/2.)
      AK(1,1)=D2*OX*((1.+NU)*C1+1.)
      AK(1,2)=D2*OX*N*(NU*C1+1.)
      ALL(1,4)=(NU-(1.-NU**2)*C1)
      AK(2,1)=B*(1.+NU)
      AK(2,2)=B*N
      ALL(2,3)=B*(1.+NU)*OX
      C2=2.+2.*NU-NU*N**2
      AK(3,1)=D1*N*(1./C2-OX/2.)
      ALL(3,4)=-N*(1.-NU)/C2
    6 RETURN
      END
```

```
      SUBROUTINE BDB(K,DEL,NU,B,DB,D,DD)
C     SUBROUTINE BDB-- THIS SUBROUTINE CALCULATES THE BENDING STIFFNESS
C     ,D, THE MEMBRANE STIFFNESS,B, AND THE DERIVATIVES OF D AND B,DD AND
C     DB, RESPECTIVELY, FOR A SHELL COMPOSED OF A CORE HAVING NO STIFFNESS
C     AND TWO SYMMETRICAL COVER PLATES
      REAL NU,N,LAM
      HRB=HRA(K,DEL)
      DHRB=DHRA(K,DEL)
      H=HHT(K,DEL)
      DH=DHHT(K,DEL)
      D2=1.-NU**2
      B=2.*H/(D2*HRB)
      D=H**3*(3./(2.*HRB)-3./HRB**2+2./HRB**3)/(D2*3.)
      DB=2.*DH/(D2*HRB)-   B*DHRB/HRB
      DD=3.*DH*D/H+H**3*DHRB/(D2*HRB**2)*(-.5+2./HRB-2./HRB**2)
      DD=DD/3.
      RETURN
      END
```

```
      SUBROUTINE POLE(N,DEL,A1,A2)
C SUBROUTINE POLE-  THIS SUBROUTINE CALCULATES THE FINITENESS CONDITIONS FOR
C A CLOSED SHELL
      DIMENSION A1(4,4),A2(4,4)
      REAL N
      DO 1 I=1,4
      DO 1 J=1,4
      A1(I,J)=0.
    1 A2(I,J)=C.
      IF(N.EQ.0.) GO TO 2
      IF((N.EQ.1.).OR.(N.EQ.-1.)) GO TO 3
      A1(1,1)=.5
      A1(2,2)=.5
      A2(1,1)=.5
      A2(2,2)=.5
      IF(N.NE.2.) GO TO 4
      A2(3,3)=1./DEL
      A1(3,3)=-1./DEL
      A1(4,4)=-1./DEL
      A2(4,4)=1./DEL
      RETURN
    4 A1(4,4)=.5
      A2(4,4)=.5
      A1(3,3)=.5
      A2(3,3)=.5
      RETURN
    2 A1(1,1)=.5
      A1(2,2)=.5
      A1(3,3)=-1./DEL
      A1(4,4)=-1./DEL
      A2(1,1)=.5
      A2(2,2)=.5
      A2(3,3)=1./DEL
      A2(4,4)=1./DEL
      RETURN
    3 A1(2,1)=.5
      A1(2,2)=.5
      A1(1,1)=-1./DEL
      A1(3,3)=.5
      A1(4,4)=.5
      A2(2,1)=.5
      A2(2,2)=.5
      A2(1,1)=1./DEL
      A2(3,3)=.5
      A2(4,4)=.5
      RETURN
      END
```

```
      SUBROUTINE INPUT (NMAX,IND6)
      COMMON
     1/BL1/R(102),GAM(102),OMT(102),OMXI(102),DEOMX(102)
     3/BL3/NU,LAM,N,EALSIG,CHAR,DEL
     9/BL9/D(4,4),ZDOT(4,102),ZDDOT(4,102),EPS                          *
     6/BL14/AO,TIM,CPS                                                   *
      REAL N,NU,LAM                                                      *
      PI=3.14159265358979                                                *
      RI= 40.1                                                           *
      RO=90.                                                             *
      ANG= PI/6.                                                         *
      DEL= (RO- RI)/(RO*COS(ANG)*FLOAT(NMAX-2))                          *
      R(NMAX)= 1.0                                                       *
      R(1)= RI/RO                                                        *
      DELR= (R(NMAX)-R(1))/FLOAT(NMAX-2)                                 *
      R(2)= R(1)+ DELR/2                                                 *
      NM1= NMAX-1                                                        *
      DO 1 I=3, NM1                                                      *
    1 R(I)= R(I-1)+ DELR                                                 *
      DO 2 I=1, NMAX                                                     *
      GAM(I)= COS(ANG)/R(I)                                              *
      OMT(I)= SIN(ANG)/R(I)                                              *
      OMXI(I)= 0.                                                        *
    2 DEOMX(I)=0.                                                        *
C  IF RUNTYPE = 2 OR 3 THEN SET EE AND RHO TO FIND TIME INCREMENT, EPS. HERE  *
C    EE IS THE MODULUS OF ELASTICITY AND RHO IS THE DENSITY(LBS/IN**3)   *
      GACC= 386.088527                                                   *
      EE= 10500000.                                                      *
      RHOA= 0.1                                                          *
      RHO= RHOA*2.75/27.                                                 *
      SS= EE/RHO                                                         *
      EDRHO= SS*GACC                                                     *
      EPS= SQRT(EDRHO/CHAR**2)*TIM                                       *
      RETURN
      END
```

```
      FUNCTION HHT(K,DEL)
      HHT=1.
      RETURN
      END
```

```
      FUNCTION TEMP(K,DEL)
      TEMP=0.
      RETURN
      END
```

```
FUNCTION DTEMP(K,DEL)
DTEMP=0.
RETURN
END
```

```
FUNCTION DELT(K,DEL)
DELT=0.
RETURN
END
```

```
FUNCTION DHHT(K,DEL)
DHHT=0.
RETURN
END
```

```
FUNCTION HRA(K,DEL)
HRA=27.
RETURN
END
```

```
FUNCTION DHRA(K,DEL)
DHRA=0.
RETURN
END
```

```
FUNCTION DDELT(K,DEL)
DDELT=0.
RETURN
END
```

```
      FUNCTION P(K)
      COMMON /BL3/NU, LAM, N, EALSIG, CHAR, DEL
      COMMON /BL9/ C(4,4), ZDOT(4,102), ZDDOT(4,102), EPS              *
      COMMON/BL1C/ JTIME                                               *
     C/BL12/INITZ                                                      *
     6/BL14/AO,TIM,CPS                                                 *
      COMMON /BL16/TAU                                                 *
     K/BL18/RUNTYPE, JUMP(5)                                           *
      INTEGER RUNTYPE                                                  *
      REAL N                                                           *
      REAL LAM                                                         *
      DATA PI/3.14159265358979/                                        *
      CPS=10.                                                          *
      TIM=.05/CPS                                                      *
      TAU= FLOAT(JTIME)*TIM                                            *
      PSIBAR = 5.*PI/180.                                              *
      FEE=PI/6.                                                        *
      PSI= PSIBAR*SIN(CPS*TAU*2.*PI)                                   *
      Q=1./LAM                                                         *
      SA=SIN(PSI)                                                      *
      CA=COS(PSI)                                                      *
      SF= SIN(FEE)                                                     *
      CF=COS(FEE)                                                      *
      PO= -Q*(SA*SA*SF*SF+ 2.*CA*CA*CF*CF)                             *
      P1= -4.*Q*CA*SA*SF*CF                                            *
      P2= -Q*SA*SA*SF*SF                                               *
      IF (N .EQ. 0.) P=PO                                              *
      IF (N .EQ. 1.) P=P1                                              *
      IF (N .EQ. 2.) P=P2                                              *
      RETURN
      END
```

```
      FUNCTION PT(K,DEL)
      PT=0.
      RETURN
      END
```

```
      FUNCTION PX(K,DEL)
      PX=0.
      RETURN
      END
```

```
      FUNCTION ZINIT(KK, I)                                            *
C     KK= 1 IS FOR ZINIT= U XI(I,0)                                    *
C     KK= 2 IS FOR ZINIT= U THETA(I,0)                                 *
C     KK= 3 IS FOR ZINIT= W(I,0)                                       *
      ZINIT=0.                                                         *
      RETURN                                                           *
      END                                                              *
```

```
      FUNCTION ZDINIT(KK, I)                                                  *
C     KK= 1 IS FOR ZDINIT= U XI DOT(I,0)                                      *
C     KK= 2 IS FOR ZDINIT= U THETA DOT(I,0)                                   *
C     KK= 3 IS FOR ZDINIT= W DOT(I,0)                                         *
      ZDINIT= 0.                                                              *
      RETURN                                                                  *
      END                                                                     *
```

```
      PROGRAM SUMUP(INPUT, OUTPUT, TAPE14)                                    *
      DIMENSION ANS(11, 39, 21)                                               *
      INTEGER FREQ                                                            *
      INTEGER RUNTYPE                                                         *
   11 FORMAT (/1X*STA*        2X*N XI*8X*N THETA*5X*N XITHETA*3X*SHEAR*7X     *
     1 *M XI*8X*M THETA*5X*M XITHETA*3X*U XI *6X* UTHETA*6X*W*11X*PHI*)       *
   12 FORMAT(1X,I2,11E12.3)                                                   *
   14 FORMAT (*1*5X*THE TIMESTEP IS*I6,5X*TIME=*E15.7)                         *
      DATA PI/3.14159265358979/                                               *
      REWIND 14                                                               *
      CALL RECIN(14, 1, IC, FREQ, NMAX, NTIME, NFOUR, AK, TIM, RUNTYPE)       *
      LL=1                                                                    *
      IF(RUNTYPE.EQ. 2) LL=2                                                  *
      THET=AK*PI                                                              *
      PRINT 5, FREQ, NMAX, NTIME, NFOUR, THET                                 *
    5 FORMAT(5X*FREQ=*I6,10X*NMAX=*I6,10X*NTIME=*I6,10X*NFOUR=*I6             *
     1 /5X*THETA=*E14.7)                                                      *
      KK =11                                                                  *
      N1=(NMAX-2)/FREQ+2                                                      *
      NFOUR=NFOUR+1                                                           *
      DO 20 J=1,N1                                                            *
      DO 20 I=1, KK                                                           *
      DO 20 K=1,NTIME                                                         *
   20 ANS(I,J,K)=0.                                                           *
      DO 3 M=1,NFOUR                                                          *
      AN=M-1                                                                  *
      ANG=AN*THET                                                             *
      DO 3 K=LL,NTIME
      DO   3 I=1,KK                                                           *
      DO   3 J=1,N1                                                           *
      CALL RECIN(14, 1, IC, ABC)                                              *
      IF ((I.EQ. 3) .OR.(I.EQ.7) .OR. (I.EQ.9).OR.(I.EQ.11))GO TO 2           *
      ANS(I,J,K)=ANS(I,J,K)+ABC*COS(ANG)                                      *
      GO TO 3                                                                 *
    2 ANS(I,J,K)=ANS(I,J,K)+ABC*SIN(ANG)                                      *
    3 CONTINUE                                                                *
      DO 13 K=LL,NTIME                                                        *
      NCYC= K-1                                                               *
      TIME=TIM*FLOAT(NCYC)                                                    *
      PRINT 14, NCYC, TIME                                                    *
      PRINT 11                                                                *
      DO 13 J=1,N1                                                            *
      L=J+(J-2)*(FREQ-1)                                                      *
      IF (J .EQ.1) L=1                                                        *
      IF (J .EQ.2) L=2                                                        *
      IF (J .EQ.NMAX) L=NMAX                                                  *
      PRINT 12, L, (ANS(I,J,K),I=1,KK)                                        *
   13 CONTINUE                                                                *
      STOP                                                                    *
      END OF SUMUP                                                            *
```

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. Sanders, J. Lyell, Jr.: An Improved First-Approximation Theory for Thin Shells. NASA TR R-24, 1959.
4. 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.
5. Houbolt, John C.: A Recurrence Matrix Solution for the Dynamic Response of Aircraft in Gusts. NACA Rep. 1010, 1951. (Supersedes NACA TN 2060.)
6. Anon.: Control Data 6400/6500/6600 Computer Systems FORTRAN Reference Manual. Publ. No. 60174900B, Control Data Corp., Nov. 1967.
7. Comm. on Metric Pract.: ASTM Metric Practice Guide. NBS Handbook 102, U.S. Dep. Com., Mar. 10, 1967.
8. Cohen, Gerald A.: The Effect of Edge Constraint on the Buckling of Sandwich and Ring-Stiffened 120 Degree Conical Shells Subjected to External Pressure. NASA CR-795, 1967.

NASA-Langley, 1971 —— 32 L-6298

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

OFFICIAL BUSINESS

FIRST CLASS MAIL

POSTAGE AND FEES PAID
NATIONAL AERONAUTICS AND
SPACE ADMINISTRATION

05U 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 OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546