NASA CONTRACTOR REPORT

NASA CR-1987

A COMPUTER PROGRAM FOR THE GEOMETRICALLY NONLINEAR STATIC AND DYNAMIC ANALYSIS OF ARBITRARILY LOADED SHELLS OF REVOLUTION, THEORY AND USERS MANUAL

by Robert E. Ball

Prepared by
NAVAL POSTGRADUATE SCHOOL
Monterey, Calif. 93940
for Langley Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • APRIL 1972

1. Report No. NASA CR-1987	2. Government Accession No.	3.
4. Title and Subtitle A COMPUTER PROGRAM FOR THE GEOMETRICALLY NONLINEAR STATIC AND DYNAMIC ANALYSIS OF ARBITRARILY LOADED SHELLS OF REVOLUTION, THEORY AND USERS MANUAL		5. Report Date April 1972
		6. Performing Organization Code
7. Author(s) Robert E. Ball		8. Performing Organization Report No.
		10. Work Unit No.
9. Performing Organization Name and Address Naval Postgraduate School Monterey, CA 93940		11. Contract or Grant No. L-16438
		13. Type of Report and Period Covered Contractor Report
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 digital computer program known as SATANS - Static and Transient Analysis, Nonlinear, Shells, for the geometrically nonlinear static and dynamic response of arbitrarily loaded shells of revolution is presented. Instructions for the preparation of the input data cards and other information necessary for the operation of the program are described in detail and two sample problems are included. The governing partial differential equations are based upon Sanders' nonlinear thin shell theory for the conditions of small strains and moderately small rotations. The governing equations are reduced to uncoupled sets of four linear, second order, partial differential equations in the meridional and time coordinates by expanding the dependent variables in a Fourier sine or cosine series in the circumferential coordinate and treating the nonlinear modal coupling terms as pseudo loads. The derivatives with respect to the meridional coordinate are approximated by central finite differences, and the displacement accelerations are approximated by the implicit Houbolt backward difference scheme with a constant time interval. At every load step or time step each set of difference equations is repeatedly solved, using an elimination method, until all solutions have converged. All geometric and material properties of the shell are axisymmetric, but may vary along the shell meridian. The applied load may consist of any combination of pressure loads, temperature distributions and initial conditions that are symmetric about a datum meridional plane. The shell material is isotropic, but the elastic modulus may vary through the thickness. The boundaries of the shell may be closed, free, fixed, or elastically restrained. The program is coded in the FORTRAN IV language and is dimensioned to allow a maximum of 10 arbitrary Fourier harmonics and a maximum product of the total number of meridional stations and the total number of Fourier harmonics of 200. The program requires 155,000 bytes of core storage.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Shell analysis Numerical methods Shells of revolution Structural dynamics	Unclassified - Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	103	$3.00

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

1. Shells (structural forms)
2. Bodies of revolution--mechanical properties
3. Computer programs
4. Modulus of elasticity

CONTENTS

<u>TABLES</u>

A COMPUTER PROGRAM FOR THE GEOMETRICALLY NONLINEAR STATIC AND DYNAMIC ANALYSIS OF ARBITRARILY LOADED SHELLS OF REVOLUTION, THEORY AND USERS MANUAL

by Robert E. Ball

SUMMARY

A digital computer program known as SATANS - Static and Transient Analysis, Nonlinear, Shells, for the geometrically nonlinear static and dynamic response of arbitrarily loaded shells of revolution is presented. Instructions for the preparation of the input data cards and other information necessary for the operation of the program are described in detail and two sample problems are included. The governing partial differential equations are based upon Sanders' nonlinear thin shell theory for the conditions of small strains and moderately small rotations. The governing equations are reduced to uncoupled sets of four linear, second order, partial differential equations in the meridional and time coordinates by expanding the dependent cariables in a Fourier sine or cosine series in the circumferential coordinate and treating the nonlinear modal coupling terms as pseudo loads. The derivatives with respect to the meridional coordinate are approximated by central finite differences, and the displacement accelerations are approximated by the implicit Houbolt backward difference scheme with a constant time interval. At every load step or time step each set of difference equations is repeatedly solved, using an elimination method, until all solutions have converged. All geometric and material properties of the shell are axisymmetric, but may vary along the shell meridian. The applied load may consist of any combination of pressure loads, temperature distributions and initial conditions that are symmetric about a datum meridional plane. The shell material is isotropic, but the elastic modulus may vary through the thickness. The boundaries of the shell may be closed, free, fixed, or elastically restrained. The program is coded in the FORTRAN IV language and is dimensioned to allow a maximum of 10 arbitrary Fourier harmonics and a maximum product of the total number of meridional stations and the total number of Fourier harmonics of 200. The program requires 155,000 bytes of core storage.

INTRODUCTION

The design of many shell structures is influenced by the geometrically nonlinear response of the shell when subjected to static and/or dynamic loads. As a consequence, a number of investigations have been devoted to the study of the buckling phenomenon exhibited by shells. Most of the early works examine the behavior of the shallow spherical cap, the truncated cone, and the cylinder under axisymmetric loads. As a consequence of the lack of information on the axisymmetric response of shells with other meridional geometries and on the response of shells subjected to asymmetric loads, a computer program for the geometrically nonlinear static and dynamic response of arbitrarily loaded shells of revolution has been developed. The dynamic analysis capability is a recent extension of the program developed by the author for the nonlinear static analysis of arbitrarily loaded shells of revolution[1]. The program can be used to analyze any shell of revolution for which the following conditions hold:

1) The geometric and material properties of the shell are axisymmetric, but may vary along the shell meridian.

2) The applied pressure and temperature distributions and initial conditions are symmetric about a datum meridional plane.

3) The shell material is isotropic, but the modulus of elasticity may vary through the thickness. Poisson's ratio is constant.

4) The boundaries of the shell may be closed, free, fixed, or elastically restrained.

The governing partial differential equations are based upon Sanders' nonlinear thin shell theory for the condition of small strains and moderately small rotations[2]. The inplane and normal inertial forces are accounted for, but the rotary inertial terms are neglected. The set of governing nonlinear partial differential equations is reduced to an infinite number of sets of four second-order differential equations in the meridional and time coordinates by expanding all dependent variables in a sine or cosine series in terms of the circumferential coordinate. The sets are uncoupled by utilizing appropriate trigonometric identities and by treating the nonlinear coupling terms as pseudo loads. The meridional derivatives are replaced by the conventional central finite difference approximations, and the displacement accelerations are approximated by the implicit Houbolt backward differencing scheme[3]. This leads to sets of algebraic equations in terms of the dependent variables and the Fourier index. At each load or time step, an estimate of the solution is obtained by extrapolation from the solutions at the previous load or time steps. The sets of algebraic equations are repeatedly solved using Potters'[4] form of Gaussian elimination, and the pseudo loads are recomputed, until the solution converges.

An automatic variable load incrementing routine is included in the program for the static analysis. When the number of iterations are small the load is incremented in equal steps. As the nonlinear terms become large, and the number of iterations exceeds a prescribed maximum, the incremental load is reduced by a factor of five. Any number of increment reductions can be made. The load is continually increased until either the prescribed maximum number of load steps or increment reductions have been taken. Post-buckling behavior cannot be determined in the static analysis because of the method of solution employed.

This report contains a description of the theory, the method of solution, instructions for the preparation of the input data cards, and other information necessary for the operation of the program. Two sample problems are included to illustrate the data preparation and output format. For additional information concerning the accuracy and applicability of the program refer to references [5] and [6].

SYMBOLS

a = reference length

B = inplane stiffness, $\int E d_\zeta/(1-\nu^2)$

b = nondimensional inplane stiffness, $B/(E_o h_o)$

D = bending stiffness, $\int \zeta^2 \, E d_\zeta/(1-\nu^2)$

d = nondimensional bending stiffness, $D/(E_o h_o^3)$

E = elastic modulus

E_o = reference elastic modulus

$e_s, e_\theta, e_{s\theta}$ = nondimensional Fourier coefficients for the reference surface strains, equations (32)

f_s = nondimensional Fourier coefficient for the transverse force, equations (32)

$\hat{f}_s$ = nondimensional Fourier coefficient for the effective transverse force, $\hat{Q}_s/(\sigma_o h_o)$

h = thickness

h_o = reference thickness

K = last meridian station on the shell

$k_s, k_\theta, k_{s\theta}$ = nondimensional Fourier coefficients for the bending strains, equations (32)

$M_s, M_\theta, M_{s\theta}$ = bending and twisting moments per unit length

m = mass density of the shell material

$m_s, m_\theta, m_{s\theta}$ = nondimensional Fourier coefficients for bending and twisting moments, equations (32)

m_T = nondimensional Fourier coefficient for the thermal bending moment, equation (32)

$N_s, N_\theta, N_{s\theta}$ = membrane forces per unit length

$\hat{N}_{s\theta}$ = effective shear force, equation (14)

n = Fourier index

p, p_s, p_θ = nondimensional Fourier coefficients for the components of the pressure load, equations (32)

Q_s, Q_θ = transverse forces per unit length

$\hat{Q}_s$ = effective transverse force, equation (15b)

q_s, q_θ, q = meridional, circumferential, and normal components of applied pressure load

R_s, R_θ = principal radii of curvature

r = normal distance from the axis of the shell

s = meridional shell coordinate

T = time

T_o = reference time

t = nondimensional time, T/T_o

$t_s, t_\theta, t_{s\theta}$ = nondimensional Fourier coefficients for membrane forces, equations (32)

$\hat{t}_{s\theta}$ = nondimensional Fourier coefficient for the effective shear force, $\hat{N}_{s\theta}/(\sigma_o h_o)$

t_T = nondimensional Fourier coefficient for the thermal membrane force, equations (32)

U, V = displacements tangent to the meridian and to the parallel circle respectively.

u, v = nondimensional Fourier coefficients for the displacements tangent to the meridian and to the parallel circle respectively, equations (32)

W = displacement normal to the reference surface

w = nondimensional Fourier coefficient for the displacement normal to the reference surface, equations (32)

α = coefficient of thermal expansion

$\beta_s, \beta_\theta, \beta, \beta_{s\theta}$ = nondimensional coefficients for the nonlinear terms in the strain-displacement relations, equations (20a) and (20c)

γ = ρ'/ρ

Δ = nondimensional distance between stations, meridian length/a/(K_{max}-1)

δt = nondimensional time interval

$\epsilon_s, \epsilon_\theta, \epsilon_{s\theta}$ = reference surface strains, equations (5)

ϵ_T = thermal membrane force, equation (12c)

ζ = coordinate normal to the reference surface

$\eta_{ss}, \eta_\theta, \eta_s$, $\eta_{\theta s}, \eta_{\theta\theta}, \eta_{s\theta}$ = nondimensional coefficients for the nonlinear terms in the equilibrium equations, equations (20c)

θ = circumferential angle

$\varkappa_s, \varkappa_\theta, \varkappa_{s\theta}$ = bending strains, equations (6)

$\varkappa_T$ = thermal bending moment, equation (12d)

μ = nondimensional mass, $a^2 \int m d_\zeta / (h_o E_o T_o{}^2)$

ν = Poisson's ratio

ξ = nondimensional meridional coordinate, s/a

ρ = nondimensional radius, r/a

σ_o = reference stress level

ω_s, ω_θ = nondimensional curvatures, a/R_s, a/R_θ

$\Phi_s, \Phi_\theta, \Phi$ = reference surface rotations, equations (7)

$\varphi_s, \varphi_\theta, \varphi$ = nondimensional Fourier coefficients for the rotations, equations (32)

τ = local temperature change

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

$(\ ')$ = $\frac{\partial}{\partial s}$ or $\frac{\partial}{\partial\xi}$

$\left(^{(n)}\right)$ = Fourier series coefficient

MATRICES

$A,B,\overline{B},C$	=	4x4 matrices, equations (25b) and (28b)
E,F,G,H,J	=	4x4 matrices, reference [1]
e,f	=	1x4 column matrices, reference [1]
$g,\overline{g}$	=	1x4 column matrices, equations (25b) and (28b)
ℓ	=	1x4 boundary condition matrix
$P,\overline{P},\hat{P}$	=	4x4 matrices, equations (30)
x	=	1x4 column matrix, equations (29a) and (31a)
z	=	1x4 column matrix containing the unknown variables $u,v,w,$ and m_s
Λ,Ω	=	4x4 nondimensional boundary condition matrices
$\overline{\Lambda},\overline{\Omega}$	=	4x4 boundary condition matrices
$\overline{\mu}$	=	mass matrix

THEORY

Shell Geometry

Consider the general shell of revolution shown in figure 1. Located within this shell is a reference surface. All material points of the shell can be located using the orthogonal coordinate system s, θ, ζ, where s is the meridional distance along the reference surface measured from one boundary, θ is the circumferential angle measured from a datum meridian plane, and ζ is the normal distance from the reference surface. The positive direction of each coordinate is indicated in figure 1. For convenience, let the reference surface be positioned so that

$$\int \zeta E d\zeta = 0 \tag{1}$$

where E is the elastic modulus and the integration is carried out over h, the thickness of the shell. Thus, when E is independent of ζ the reference surface coincides with the middle surface of the shell. Further, let the location of the reference surface be described by the dependent variable r, the normal distance from the axis of the shell. Accordingly, the principal radii of curvature of the reference surface are

$$\begin{aligned} R_\theta &= r/[1 - (r')^2]^{\frac{1}{2}} \\ R_s &= - [1 - (r')^2]^{\frac{1}{2}}/r'' \end{aligned} \tag{2}$$

where a prime denotes differentiation with respect to s. Further, note the Codazzi identity

$$\left(\frac{1}{R_\theta}\right)' = r' \, (R_s^{-1} - R_\theta^{-1})/r \tag{3}$$

and the relation

$$r'' = - r/R_s \, R_\theta \tag{4}$$

Strain-displacement Relations

For a shell of revolution, the strain-displacement relations derived by Sanders take the form

$$\left.\begin{aligned} \epsilon_s &= U' + W/R_s + (\Phi_s^2 + \Phi^2)/2 \\ \epsilon_\theta &= V^{\cdot}/r + r' \, U/r + W/R_\theta + (\Phi_\theta^2 + \Phi^2)/2 \\ \epsilon_{s\theta} &= (V' + U^{\cdot}/r - r'V/r + \Phi_s \, \Phi_\theta)/2 \end{aligned}\right\} \tag{5}$$

and

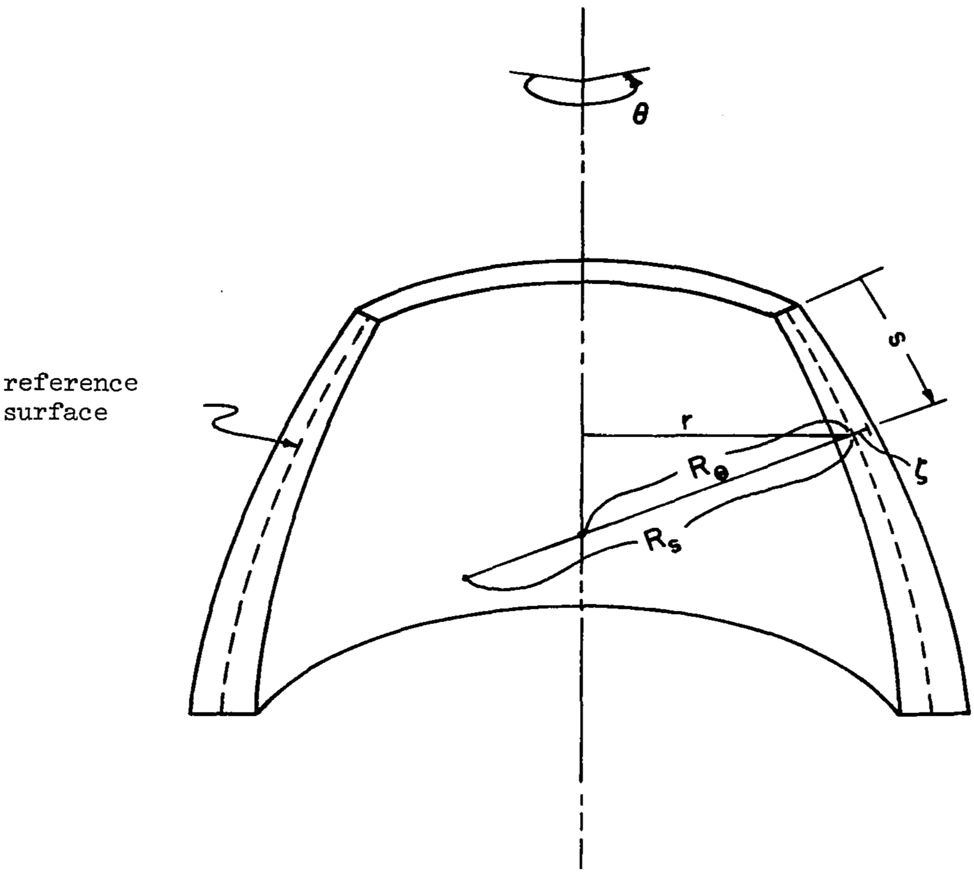

Figure 1. Shell Geometry and Coordinates

$$
\left.
\begin{aligned}
\kappa_s &= \Phi_s' \\
\kappa_\theta &= \Phi_\theta^{\cdot}/r + r' \; \Phi_s/r \\
\kappa_{s\theta} &= [\Phi_\theta' + \Phi_s^{\cdot}/r - r' \; \Phi_\theta/r + (R_\theta^{-1} - R_s^{-1}) \; \Phi]/2
\end{aligned}
\right\} \quad (6)
$$

where ϵ_s, ϵ_θ, and $\epsilon_{s\theta}$ are the reference surface strains, κ_s, κ_θ, and $\kappa_{s\theta}$ are the bending strains, U and V are displacements in the directions tangent to the meridian and to the parallel circle respectively, W is the displacement normal to the reference surface, and Φ_s, Φ_θ, and Φ are rotations defined by

$$
\left.
\begin{aligned}
\Phi_s &= - W' + U/R_s \\
\Phi_\theta &= - W^{\cdot}/r + V/R_\theta \\
\Phi \; &= (V' + r'V/r - U^{\cdot}/r)/2
\end{aligned}
\right\} \quad (7)
$$

In these equations, and henceforth, a superscript dot denotes differentiation with respect to θ. The positive direction of each displacement and rotation variable is indicated in figure 2.

Equations of Motion

Converting Sanders' equilibrium equations to the equations of motion for a shell of revolution leads to

$$
\left.
\begin{aligned}
&(rN_s)' + N_{s\theta}^{\cdot} - r'N_\theta + rQ_s/R_s + (R_s^{-1} - R_\theta^{-1}) \; M_{s\theta}^{\cdot}/2 = r(\int m d\zeta)\partial^2 U/\partial T^2 \\
&- rq_s + r(\Phi_s \; N_s + \Phi_\theta \; N_{s\theta})/R_s + [\Phi(N_s + N_\theta)]^{\cdot}/2 \\
&N_\theta^{\cdot} + (rN_{s\theta})' + r'N_{s\theta} + rQ_\theta/R_\theta + r[(R_\theta^{-1} - R_s^{-1})M_{s\theta}]'/2 = r(\int m d\zeta)\partial^2 V/\partial T^2 \\
&- rq_\theta + r(\Phi_\theta N_\theta + \Phi_s \; N_{s\theta})/R_\theta - r \; [\Phi(N_s + N_\theta)]'/2 \\
&(rQ_s)' + Q_\theta^{\cdot} - rN_s/R_s - rN_\theta/R_\theta = r(\int m d\zeta)\partial^2 W/\partial T^2 \\
&- rq + (r\Phi_s \; N_s + r\Phi_\theta N_{s\theta})' + (\Phi_s \; N_{s\theta} + \Phi_\theta \; N_\theta)^{\cdot}
\end{aligned}
\right\} (8)
$$

and

$$
(rM_s)' + M_{s\theta}^{\cdot} - r' \; M_\theta - rQ_s = 0 \qquad (9)
$$

$$
M_\theta^{\cdot} + (rM_{s\theta})' + r' M_{s\theta} - rQ_\theta = 0 \qquad (10)
$$

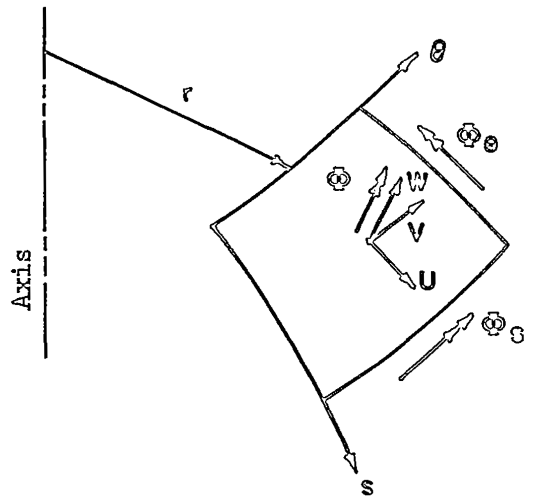

Figure 2. Positive Directions for Displacements and Rotations

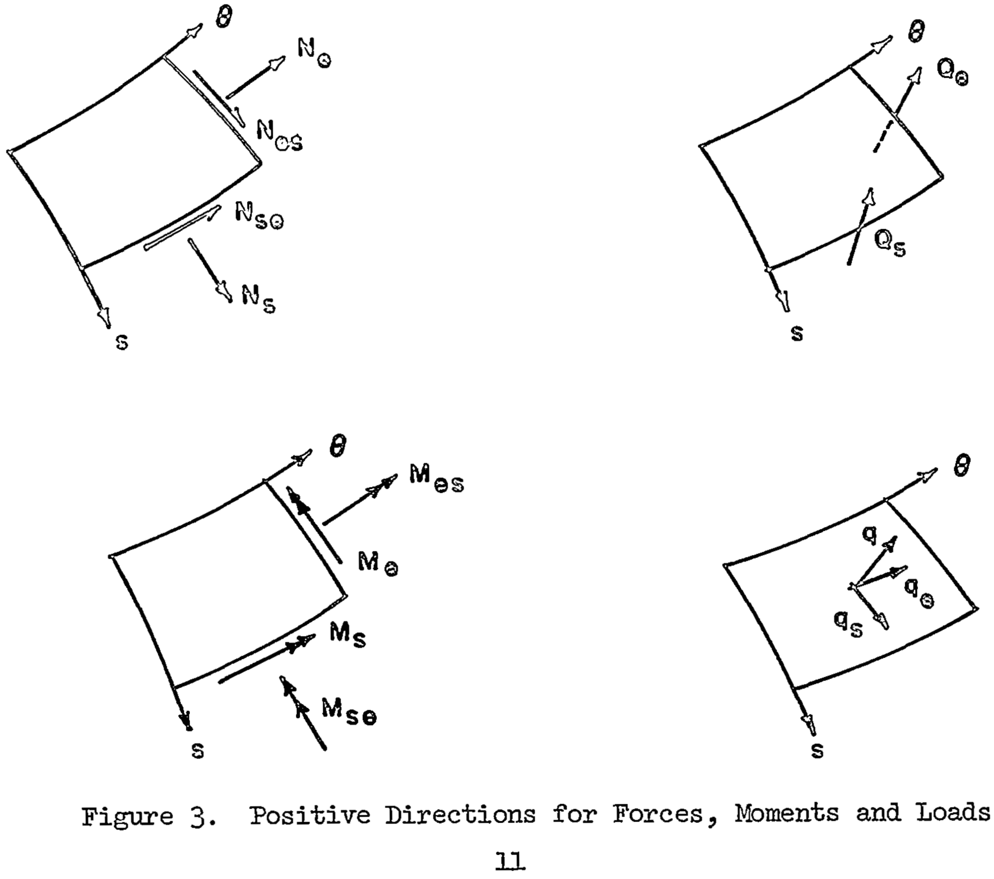

Figure 3. Positive Directions for Forces, Moments and Loads

when the effects of rotary inertia are neglected. In equations (8) - (10), m is the mass density of the shell material, T is time, q_s, q_θ, and q are the meridional, circumferential, and normal components of the applied pressure load, Q_s and Q_θ are the transverse forces per unit length, N_s, N_θ, and $N_{s\theta}$ are the membrane forces per unit length, and M_s, M_θ, and $M_{s\theta}$ are the bending and twisting moments per unit length. Refer to figure 3 for the positive directions of the pressure components, forces, and moments.

Constituitive Relations

The constituitive relations used in Sanders' nonlinear theory are the same as those proposed by Love in his first approximation to the linear, small strain theory of thin elastic shells. Noting equation (1), these can be given in the form

$$N_s = B\ (\epsilon_s + \nu\epsilon_\theta) - \epsilon_T \tag{11a}$$

$$N_\theta = B\ (\epsilon_\theta + \nu\epsilon_s) - \epsilon_T \tag{11b}$$

$$N_{s\theta} = B\ (1 - \nu)\ \epsilon_{s\theta} \tag{11c}$$

$$M_s = D\ (\kappa_s + \nu\kappa_\theta) - \kappa_T \tag{11d}$$

$$M_\theta = D\ (\kappa_\theta + \nu\kappa_s) - \kappa_T \tag{11e}$$

$$M_{s\theta} = D\ (1 - \nu)\ \kappa_{s\theta} \tag{11f}$$

where ν is Poisson's ratio, assumed constant through the thickness, and

$$B = \int E\ d\ \zeta\ /\ (1 - \nu^2) \tag{12a}$$

$$D = \int \zeta^2\ E\ d\ \zeta\ /\ (1 - \nu^2) \tag{12b}$$

$$\epsilon_T = \int \alpha\ \tau\ E\ d\ \zeta\ /\ (1 - \nu) \tag{12c}$$

$$\kappa_T = \int \zeta\ \alpha\ \tau\ E\ d\ \zeta\ /\ (1 - \nu) \tag{12d}$$

In equations (12c) and (12d), τ is the local temperature change and α is the coefficient of thermal expansion.

Boundary Conditions

In Sanders' nonlinear theory, the conditions to prescribe on the edge of a shell of revolution are

$$\left.\begin{array}{ll} N_s \text{ or } U & \hat{N}_{s\theta} \text{ or } V \\ \hat{Q}_s \text{ or } W & M_s \text{ or } \Phi_s \end{array}\right\} \qquad (13)$$

where $\hat{N}_{s\theta}$ and $\hat{Q}_s$ are the effective shear and transverse forces per unit length defined by

$$\hat{N}_{s\theta} = N_{s\theta} + (3\,R_\theta^{-1} - R_s^{-1})M_{s\theta}/2 + (N_s + N_\theta)\,\Phi\,/2 \qquad (14)$$

$$\hat{Q}_s = Q_s + M^{\bullet}_{s\theta}/r - \Phi_s\,N_s - \Phi_\theta\,N_{s\theta} \qquad (15a)$$

Using the equilibrium equation (9) to eliminate Q_s from equation (15a) leads to

$$\hat{Q}_s = [(rM_s)' + 2M^{\bullet}_{s\theta} - r'M_\theta]/r - \Phi_s N_s - \Phi_\theta N_{s\theta} \qquad (15b)$$

Elastic restraints at the edge of a shell can be provided for by linearly relating the forces or moment to the appropriate displacements or rotation. Consequently, the boundary conditions may be given in the matrix form

$$\bar{\Omega}\begin{bmatrix} N_s \\ \hat{N}_{s\theta} \\ \hat{Q}_s \\ \Phi_s \end{bmatrix} + \bar{\Lambda}\begin{bmatrix} U \\ V \\ W \\ M_s \end{bmatrix} = \ell \qquad (16)$$

where $\bar{\Omega}$ and $\bar{\Lambda}$ are 4x4 matrices and ℓ is a column matrix. The values of the elements of these matrices are determined by the conditions prescribed at the shell boundary.

METHOD OF SOLUTION

Fourier Expansions

The crux of the method used here to solve the nonlinear field equations is the elimination of the independent variable θ by expanding all dependent variables into sine or cosine series in the circumferential direction. Only loading and initial conditions that are symmetric about a datum meridian plane will be considered. Thus, the variable Φ_s can be expressed in the form*

$$\Phi_s = \frac{\sigma_o}{E_o} \sum_{n=0}^{\infty} \varphi_s^{(n)} \cos n\theta \tag{17}$$

where σ_o is a reference stress level, E_o is a reference elastic modulus, and the nondimensional series coefficient $\varphi_s^{(n)}$ is a function of the independent variables s and T. Similar series expansions can be made for the remaining dependent variables.

Modal Uncoupling

In order to eliminate the independent variable θ from the problem, and convert the partial differential equations to sets of uncoupled partial differential equations, the nonlinear terms are treated as known quantities or pseudo loads. Since every nonlinear term is the product of two Fourier series, each product can be reduced to a single trigonometric series wherein the coefficient is itself a series. For example, using equation (17), Φ_s^2 can be expressed as

$$\Phi_s^2 = \left(\frac{\sigma_o}{E_o}\right)^2 \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \varphi_s^{(m)} \varphi_s^{(n)} \cos m\theta \cos n\theta \tag{18}$$

Since

$$\cos m\theta \cos n\theta = \tfrac{1}{2} [\cos (m-n) \theta + \cos (m+n) \theta] \tag{19}$$

equation (18) can be given in the form

---

*Theoretically, the complete Fourier series including both the sine and cosine expansions should be used because of the possibility of "odd" displacements occurring under "even" loads, i.e. a bifurcation phenomenon. This aspect is not considered here.

$$\Phi_s^2 = \frac{\sigma_o}{E_o} \sum_{n=0}^{\infty} \beta_s^{(n)} \cos n\theta \tag{20a}$$

where

$$\beta_s^{(n)} = \frac{\sigma_o}{2E_o} \sum_{i=0}^{\infty} \varphi_s^{(i)} [\eta \, \varphi_s^{(i+n)} + \mu \, \varphi_s^{|i-n|}] \tag{20b}$$

with

$$\eta = \begin{cases} 0 \text{ for } n = 0 \\ 1 \text{ for } n > 0 \end{cases}, \quad \mu = \begin{cases} 1 \text{ for } i \neq n \\ 2 \text{ for } i = n \end{cases}$$

Similar series expressions can be derived for the other nonlinear terms in equations (5), (8), (14), and (15b). They are

$$\left.\begin{aligned}
\Phi_\theta^2 &= \frac{\sigma_o}{E_o} \sum_{n=0}^{\infty} \beta_\theta \cos n\theta \\
\Phi^2 &= \frac{\sigma_o}{E_o} \sum_{n=0}^{\infty} \beta \cos n\theta \\
\Phi_s \Phi_\theta &= \frac{\sigma_o}{E_o} \sum_{n=1}^{\infty} \beta_{s\theta} \sin n\theta \\
\Phi_s N_s &= \sigma_o h_o \sum_{n=0}^{\infty} \eta_{ss} \cos n\theta \\
\Phi_\theta N_{s\theta} &= \sigma_o h_o \sum_{n=0}^{\infty} \eta_{\theta s} \cos n\theta \\
\Phi \, N_s &= \sigma_o h_o \sum_{n=1}^{\infty} \eta_s \sin n\theta \\
\Phi \, N_\theta &= \sigma_o h_o \sum_{n=1}^{\infty} \eta_\theta \sin n\theta
\end{aligned}\right\} \tag{20c}$$

$$\left.\begin{aligned}\Phi_\theta N_\theta &= \sigma_o h_o \sum_{n=1}^{\infty} \eta_{\theta\theta} \sin n\theta \\ \Phi_s N_{s\theta} &= \sigma_o h_o \sum_{n=1}^{\infty} \eta_{s\theta} \sin n\theta\end{aligned}\right\} .$$

where h_o is a reference thickness.

As a result of the trigonometric series expansions, there is one set of governing equations for each value of n considered; when only the linear terms are considered the sets are uncoupled. The presence of the nonlinear terms couples the sets through terms like $\beta_s^{(n)}$ as given by equation (20b). However, by treating the nonlinear terms as known quantities and grouping them with the load terms, the sets of equations become uncoupled.

Final Equations

Budiansky and Radkowski[7] have shown that for the linear shell problem each set of Sanders' uncoupled field equations can be reduced to four second order differential equations provided M_θ is replaced by the equality obtained from the constituitive relations (11d) and (11e)

$$M_\theta = \nu M_s + D\ (1-\nu^2)\kappa_\theta - (1-\nu)\kappa_T \tag{21}$$

to prevent derivatives of W higher than two from appearing. The same procedure is used here. The four unknown dependent variables are the nondimensional series coefficients $u^{(n)}$, $v^{(n)}$, $w^{(n)}$, and $m_s^{(n)}$ corresponding to U, V, W, and M_s respectively. Three of the final four equations are derived from the equations of motion (8) by applying the rotational equilibrium equations (9) and (10), the constituitive relations (11) and (21), and the strain-displacement relations (5), (6), and (7). The fourth equation is derived from the meridional bending moment-curvature relationship given by (11d) with κ_s and κ_θ expressed in terms of the displacements.

A convenient representation of these four equations is the non-dimensional matrix form

$$E^{(n)}\ z^{(n)''} + F^{(n)}\ z^{(n)'} + G^{(n)}\ z^{(n)} = e^{(n)} + \bar{\mu}\ \partial^2\ z^{(n)}/\partial t^2 \tag{22}$$

where

$$z^{(n)} = \begin{bmatrix} u^{(n)} \\ v^{(n)} \\ w^{(n)} \\ m_s^{(n)} \end{bmatrix}$$

t is the nondimensional time T/T_o, T_o is a reference time, and $\overline{\mu}$ is the mass matrix given by

$$\overline{\mu} = \mu \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

The nondimensional scalar mass μ is defined by

$$\mu = \frac{a^2}{h_o E_o T_o^2} \int m d\ \zeta$$

where a is a reference length. Henceforth, the superscript n will be dropped for convenience.

The E, F, G, and e in equation (22) are matrices defined in reference [1]. The elements of E, F, and G are identical with those given in reference [7] for the linear shell analysis, but the e matrix as defined in reference [1] contains both the load and thermal terms and the nonlinear terms.

The boundary conditions on z are obtained by applying the constituitive relations (11) and (21), and the strain-displacement relations (5), (6), and (7) to equation (16). This leads to the matrix equation

$$\Omega H z' + (\Lambda + \Omega J) z = \ell - \Omega f \tag{23}$$

where Ω and Λ are the nondimensional forms of $\overline{\Omega}$ and $\overline{\Lambda}$. Matrices H and J are identical with those given in reference [7] for the linear shell problem, and matrix f, as defined in reference [1], contains the thermal and nonlinear terms. In this formulation, Ω, Λ, and ℓ are not functions of n, and hence, the same set of boundary conditions applies for each value of n considered. An example of the modifications required to allow different values of ℓ for each mode is given in reference [5].

Spatial Finite Difference Formulation

Let the shell meridian be divided into K - 1 equal increments, and denote the end of each increment or station by the index i. Thus, i = 1 corresponds to the initial edge of the shell and i = K corresponds to the final edge. One fictitious station is introduced off each end of the shell at i = 0 and i = K + 1.

Let the first and second derivatives of z at station i be approximated by

$$z_i' = (z_{i+1} - z_{i-1})/2\Delta \tag{24a}$$

$$z_i'' = (z_{i+1} - 2z_i + z_{i-1})/\Delta^2 \tag{24b}$$

where Δ is the nondimensional distance between stations. Substituting equations (24a) and (24b) into equation (22) leads to

$$A_i\ z_{i+1} + B_i\ z_i + C_i\ z_{i-1} = g_i + 2\Delta\bar{\mu}_i\ (\partial^2 z/\partial t^2)_i \tag{25a}$$

where

$$\left.\begin{aligned} A_i &= 2E_i/\Delta + F_i \\ B_i &= -\ 4E_i/\Delta + 2\ \Delta\ G_i \\ C_i &= 2E_i/\Delta - F_i \\ g_i &= 2\ \Delta\ e_i \end{aligned}\right\} \tag{25b}$$

and i = 1, 2 . . . K to insure equilibrium over the total length of the shell.

At the boundaries equation (23) must be satisfied. Thus, substituting equation (24a) into equation (23) leads to

$$\frac{1}{2\Delta}\Omega_1 H_1 z_2 + (\Omega_1 J_1 + \Lambda_1)z_1 - \frac{1}{2\Delta}\Omega_1 H_1 z_0 = \ell_1 - \Omega_1 f_1 \tag{26a}$$

at the initial edge, and

$$\frac{1}{2\Delta}\Omega_K H_K z_{K+1} + (\Omega_K J_K + \Lambda_K)z_K - \frac{1}{2\Delta}\Omega_K H_K z_{K-1} = \ell_K - \Omega_K f_K \tag{26b}$$

at the final edge.

Timewise Differencing Scheme

The inertial terms that appear in equations (25) can be approximated by Houbolt's backward differencing scheme. Accordingly,

$$\left(\frac{\partial^2 z}{\partial t^2}\right)_{i,j} = (2z_{i,j} - 5z_{i,j-1} + 4z_{i,j-2} - z_{i,j-3})/(\delta t)^2 \tag{27}$$

where j denotes the time step and δt is the nondimensional time interval. Thus, substituting equation (27) into equation (25) yields

$$A_i \, z_{i+1,j} + \overline{B}_i z_{i,j} + C_i \, z_{i-1,j} = \overline{g}_{i,j} \tag{28a}$$

where

$$\left.\begin{aligned} \overline{B}_i &= B_i - \frac{4\Delta\overline{\mu}_i}{(\delta t)^2} \\ \overline{g}_{i,j} &= g_{i,j} + \frac{2\Delta\overline{\mu}_i}{(\delta t)^2} (-5 \, z_{i,j-1} + 4 \, z_{i,j-2} - z_{i,j-3}) \end{aligned}\right\} \tag{28b}$$

and $i = 1, 2, \ldots K$.

Solution by Elimination

Eqautions (26) and (28) constitute a set of simultaneous algebraic equations in the unknowns $z_{i,j}$ provided $g_{i,j}$, $z_{i,j-1}$, $z_{i,j-2}$, and $z_{i,j-3}$ are known. There is one such set for each value of n considered. The equations can be arranged in the form shown in figure 4. Since these equations are tridiagonal in the matrix sense, Potters' form of Gaussian elimination can be used to solve for the $z_{i,j}$. In this method, recursion relationships of the form

$$x_{i,j} = \overline{P}_i \, g_{i,j} - P_i \, x_{i-1,j} \tag{29a}$$

$$z_{i,j} = - P_i \, z_{i+1,j} + x_{i,j} \tag{29b}$$

are developed. A forward pass from the initial edge to the final edge computes the $x_{i,j}$, and a back substitution determines the $z_{i,j}$. The

$$
\begin{bmatrix}
\frac{-\Omega_1 H_1^{(N)}}{2\Delta} & \Omega_1 J_1^{(N)} + \Lambda_1 & \frac{\Omega_1 H_1^{(N)}}{2\Delta} & & & & & \\
C_1^{(N)} & \bar{B}_1^{(N)} & A_1^{(N)} & & & & & \\
 & C_2^{(N)} & \bar{B}_2^{(N)} & A_2^{(N)} & & & & \\
 & & \cdot & \cdot & \cdot & & & \\
 & & & \cdot & \cdot & \cdot & & \\
 & & & & & C_K^{(N)} & B_K^{(N)} & A_K^{(N)} \\
 & & & & & \frac{-\Omega_K H_K^{(N)}}{2\Delta} & \Omega_K J_K^{(N)} + \Lambda_K & \frac{\Omega_K H_K^{(N)}}{2\Delta}
\end{bmatrix}
\begin{bmatrix}
Z_0^{(N)} \\ Z_1^{(N)} \\ Z_2^{(N)} \\ \cdot \\ \cdot \\ Z_K^{(N)} \\ Z_{K+1}^{(N)}
\end{bmatrix}
=
\begin{bmatrix}
\ell_1 - \Omega_1 f_1^{(N)} \\ \bar{g}_1^{(N)} \\ \bar{g}_2^{(N)} \\ \cdot \\ \cdot \\ \bar{g}_K^{(N)} \\ \ell_K - \Omega_K f_K^{(N)}
\end{bmatrix}
$$

Figure 4. Matrix Equation for $n = N$

matrices P_i, $\overline{P}_i$, and $\hat{P}_i$ are independent of the load and solution. Hence, they are computed only once. They are

$$\left.\begin{aligned}
P_1 &= \left[\Omega_1 \left\{\frac{H_1}{2\Delta} C_1^{-1} B_1 + J_1\right\} + \Lambda_1\right]^{-1} \left[\Omega_1 \frac{H_1}{2\Delta} (I + C_1^{-1} A_1)\right] \\
\overline{P}_i &= [B_i - C_i P_{i-1}]^{-1} \\
P_i &= \overline{P}_i A_i \\
\hat{P}_i &= \overline{P}_i C_i
\end{aligned}\right\} \tag{30}$$

The initial value of x is

$$x_1 = \left[\Omega_1 \left\{\frac{H_1}{2\Delta} C_1^{-1} B_1 + J_1\right\} + \Lambda_1\right]^{-1} \left\{\ell_1 + \Omega_1 \left[\frac{H_1}{2\Delta} C_1^{-1} \overline{g}_1 - f_1\right]\right\} \tag{31a}$$

and the value of z at station K + 1 is

$$\begin{aligned}
z_{K+1} = &\left[\Omega_K \frac{H_K}{2\Delta} (1 - P_{K-1} P_K) - (\Omega_K J_K + \Lambda_K) P_K\right]^{-1} \\
&\left\{\ell_K + \Omega_K \frac{H_K}{2\Delta} (-P_{K-1} x_K + x_{K-1}) - (\Omega_K J_K \right. \\
&\left. + \Lambda_K) x_K - \Omega_K f_K\right\}
\end{aligned} \tag{31b}$$

Poles

The equations (26a) and (26b) are applicable when the shell has edges. If the shell has a pole, r=0, and special "boundary" conditions are required to assure finite stresses and strains at the pole. These conditions are derived in reference [1].

SOLUTION PROCEDURE

As a consequence of the selection of the Houbolt timewise differencing scheme, both static and dynamic analyses can be carried out using essentially the same set of equations and solution procedure.

Static Analysis

For a static analysis, $\mu=0$, and the applied load is increased monotonically. Thus, the index j denotes the load step.

The procedure used to determine z for the monotonically increasing load is illustrated in figure 5 and described below:

1) The matrices P_i, $\overline{P}_i$, and $\hat{P}_i$ are computed.

2) A solution is obtained for a specified increment (DELØAD) of each Fourier coefficient of the design load. All pseudo loads are taken as zero.

3) The new solution is used to calculate the nonlinear terms, and a new value of the load vector $\overline{g}$ is obtained for each n. Additional values of n may be introduced by the nonlinear terms.

4) A solution is obtained for the new value of $\overline{g}$ for each n and is compared with the previous solution.

5) If the difference between two consecutive solutions, at any station and for any n, is greater than a specified percentage (EPS) of the maximum solution in that mode then step #3 is repeated. However, if the number of iterations has exceeded a specified maximum (ITRMAX), the total load (ALØAD) is reduced by one load increment, the increment (DELØAD) is reduced by a factor of 5, and this new increment is added to the load. If a specified number of load reductions (LCHMAX) have been made, the program ends.

6) If the two consecutive solutions have converged, another load increment is added, provided the number of load steps is less than a specified maximum (LSMAX). An estimate of the solution for this new load is made by linear extrapolation using the two preceeding converged solutions, and step #3 is repeated.

Since the method of solution is based on a nonlinear pseudo load approach, the shell reacts equally, in a linear fashion, to any change in either the applied load or the pseudo load. Thus, failure of the solution to converge in any mode can be attributed to two types of nonlinear behavior. Both types are illustrated in figure 5. The existance of a maximum or an inflection point on the softening load-deflection curve A represents a type of behavior for which a solution can be obtained only below the points of zero or nearly zero slope. On the other hand, the existence of a stiffening nonlinearity, as illustrated by curve B of figure 5, can also cause a convergence failure when ever the slope becomes too steep. Thus, in general, it is necessary to examine the load-displacement behavior of the shell in order to determine the cause of the convergence failure.

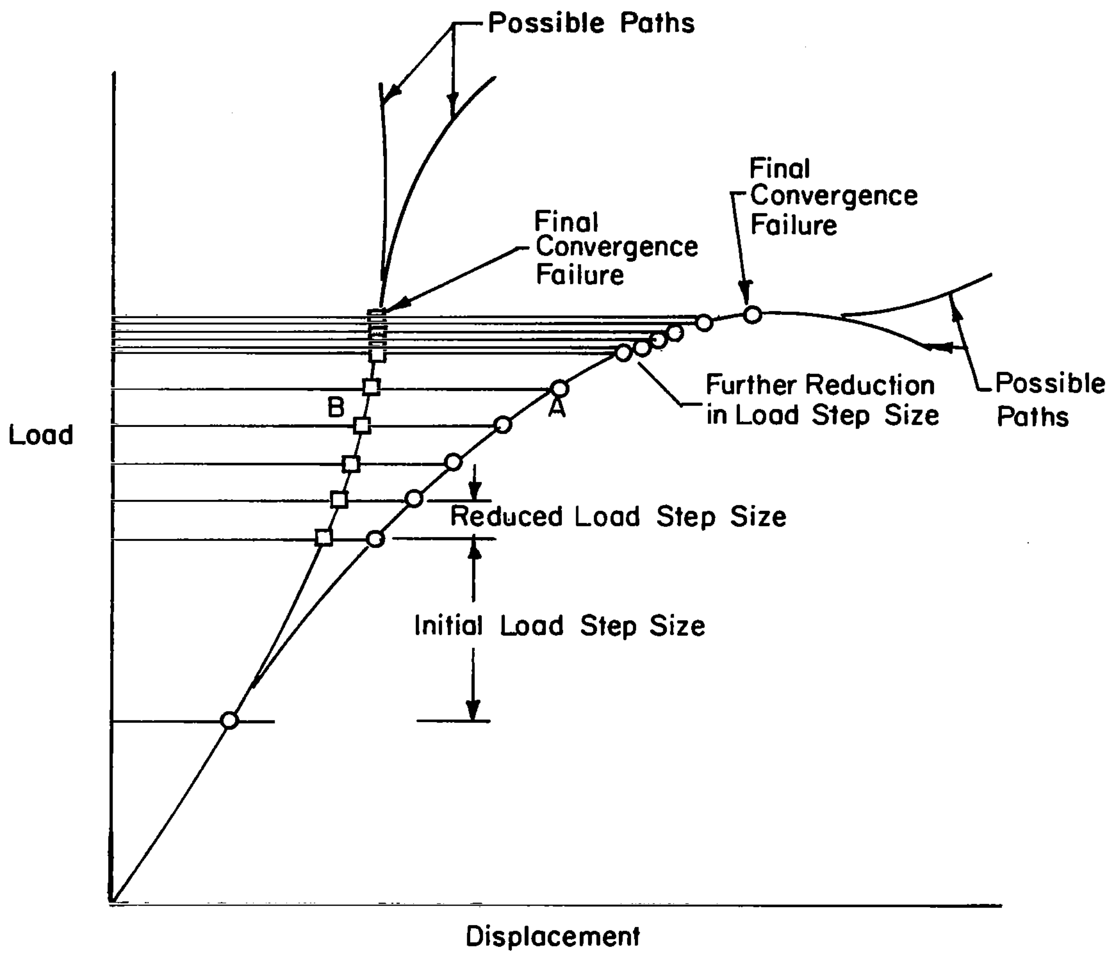

Figure 5. Typical Load-Displacement Curves from a Static Analysis

Dynamic Analysis

The dynamic analysis proceeds in essentially the same manner as the static analysis. The only differences are due to the fact that; (1) the applied load is not monotonically increased, but instead is a function of the time step j; and (2) initial conditions on z and $\partial z/\partial t$ are required to start the procedure. A brief description of the procedure used to obtain the response of the shell for a specified period of time and time increment (DELØAD) is given below:

1) The matrices P_i, $\overline{P}_i$, and $\widehat{P}_i$ are computed.

2) The solutions at $j = 0$, -1 and -2 are computed for each n from the specified initial conditions using the expressions

$z_{i,0}$ = initial condition on z supplied by user

$(\partial z/\partial t)_{i,0}$ = initial condition on $\partial z/\partial t$ supplied by user

$$z_{i,-1} = z_{i,0} - \delta t\ (\partial z/\partial t)_{i,0}$$

$$z_{i,-2} = z_{i,0} - 2\delta t\ (\partial z/\partial t)_{i,0}$$

for $i = 0,1,\ldots K+1$. An estimate of the solution at j=1 is obtained for each n from

$$z_{i,1} = z_{i,0} + \delta t\ (\partial z/\partial t)_{i,0}$$

for $i = 0, 1, 2, \ldots K+1$.

3) This new solution is used to calculate the nonlinear terms, and a new value of $\overline{g}$ is obtained for each n using the estimated nonlinear terms and applied loads at j and the solution at j-1, j-2, and j-3.

4) A solution is obtained for the new value of $\overline{g}$ for each n and is compared with the previous solution at j.

5) If the difference between two consecutive solutions, at any station and for any n, is greater than a specified percentage (EPS) of the maximum solution in that mode then step #3 is repeated. However, if the number of iterations has exceeded a specified maximum (ITRAMAX) the program ends.

6) If the two consecutive solutions are sufficiently close, an estimate of the solution at j+1 is obtained by quadratic extrapolation from the solution at j, j-1, and j-2. The preceeding solutions are updated, and step #3 is repeated for the new time step j=j+1, provided the number of time steps is less than a specified maximum (LSMAX).

Two comments are in order here. First, the approximations used to obtain the solutions at $j = -1$ and -2 are not the ones suggested by Houbolt. Houbolt's approximations require a change in the $\overline{B}$ matrix at the first time step. This is time consuming since it necessitates the recomputation of the P_i, $\overline{P}_i$, and P_i matrices, and does not appear to be worth the extra effort. Second, the time interval is usually so small no iteration is required since the difference between the estimated solution and computed solution is generally negligible. However, when the shell becomes dynamically unstable, the solution may not converge, even with iteration. Thus, the maximum number of iterations allowed should be small.

COMPUTER PROGRAM

Brief Description

The program described in this report - SATANS - Static and Transient Analysis, Nonlinear, Shells,- is a modified version of the program described in reference (1). The revisions were made by personnel at the NASA Langley Research Center and by the original author. The main difference between the two versions is the addition of the capability for dynamic analysis. Another difference is in the manner in which core storage is allocated for the solution vector z. The solution vector is now handled as a two dimensional array instead of a three dimensional array, allowing the user the freedom of prescribing almost any combination of meridional and circumferential unknowns within the dimensions of the array. In the modified program up to 200 unknowns may be specified so that the product of the total number of meridional stations and the total number of Fourier harmonics must be less than 201. However, the maximum number of Fourier harmonics that can be considered is still 10. Any combination of harmonics may be used. For example, n = 5, 0, 22, and 91 is allowed; there is no restriction on the order nor on the number.

A change was also made in the test for convergence. The original program required two consecutive solutions to differ by less than a specified percentage of the latest solution. This test was made at every station, for every mode, except when the solution was less than 10^{-6}. Experience with this routine showed it to be too restrictive. Consequently, it was replaced with the requirement that for each harmonic the difference between two consecutive solutions at each station must be less than a specified percentage of the maximum solution in that harmonic, considering all the stations, except when the solution is less than 10^{-5}. This new test for convergence appears to provide converged, accurate solutions in fewer iterations than the original scheme.

The output subroutine was also modified in order to present the data in more compact form; the COMMON and DIMENSION statements were changed to allow the compilation of the program in any order; and several bugs were detected and eliminated. The operational parameters of the program and the boundary conditions are still read in on cards, but the geometry and mass of the shell, the inplane and bending stiffnesses, the pressure and thermal loads, and the initial conditions are introduced through user-prepared subroutines. The input and output data may be in either dimensional form or non-dimensional form, and no special tapes, discs, or routines are required for execution. However, a tape is required if the dynamic analysis is to be restarted. All of these changes have enlarged the program to the extent that it now requires a core space of approximately 150,000 bytes on an IBM 360/67 digital computer and can no longer be executed on a 32,000 word computer. The compilation time using the FORTRAN IV Compiler, Level H, is slightly less than 2 minutes on the NPS IBM 360/67. The static version of this program has been available from COSMIC as M70-10098, LAR-10736.

The computer program has been used to solve a number of static and dynamic problems for both axisymmetric and asymmetric loads [5,6]. Two of these problems are presented here to illustrate the input and output features of the program.

Nondimensionalization

The input and output data may be in either dimensional or non-dimensional form. The dimensional parameters are E_o, a reference elastic modulus, σ_o, a reference stress, a, a reference length, and h_o, a reference thickness. The variables are made nondimensional as follows:

$$\left.\begin{array}{llll}
\rho = r/a & & t_s^{(n)} = N_s^{(n)}/(\sigma_o h_o) \\
\xi = s/a & & f_s^{(n)} = Q_s^{(n)}/(\sigma_o h_o) \\
\gamma = \frac{dr}{ds}/r/a & & m_s^{(n)} = M_s^{(n)}(a/\sigma_o h_o^3) \\
\omega_s = a/R_s & & u^{(n)} = U^{(n)}(E_o/a\sigma_o) \\
\omega_\theta = a/R_\theta & & \varphi_s^{(n)} = \Phi_s^{(n)}(E_o/\sigma_o) \\
b = B/(E_o h_o) & & e_s^{(n)} = \epsilon_s^{(n)}(E_o/\sigma_o) \\
d = D/(E_o h_o^3) & & k_s^{(n)} = \kappa_s^{(n)}(aE_o/\sigma_o) \\
\mu = \int m d\zeta (a^2/h_o E_o T_o^2) & & p_s^{(n)} = q_s^{(n)}(a/\sigma_o h_o) \\
t_T^{(n)} = \epsilon_T^{(n)}/(\sigma_o h_o) & & m_T^{(n)} = \kappa_T^{(n)}(a/\sigma_o h_o^3)
\end{array}\right\} \quad (32)$$

Similar expressions hold for $t_\theta^{(n)}$, $m_\theta^{(n)}$, etc.

Example of a Static Analysis

The first problem is the static analysis of a clamped, shallow spherical cap of constant thickness and uniformly loaded over one-half of the shell from $\theta = -90^\circ$ to $\theta = 90^\circ$. This problem was first considered by Famili and Archer [8]. The geometry of the spherical cap can be specified by the single nondimensional parameter λ, where

$$\lambda = 2[3(1-\nu^2)]^{\frac{1}{4}}(H/h)^{\frac{1}{2}}$$

H is the rise of the shell and h is its thickness. The classical buckling pressure of a complete sphere is denoted by q_o, where

$$q_o = 2Eh^2/R_s{}^2/[3(1-\nu^2)]^{\frac{1}{2}}$$

For this analysis,

$\lambda = 6$

$\nu = .3$

meridian length = 105 in.

$R_s = R_\theta = 1000$ in.

$E = 27.3 \times 10^6$ lb/in.^{2}

$h = 1$ in.

$B = 30.0 \times 10^6$ lb/in.

$D = 2.5 \times 10^6$ lb - in.

$q = -30$ lb/in.^{2} $\quad\quad -\pi/2 \le \theta \le \pi/2$

$q_o = -33.1$ lb/in.^{2}

The reference parameters for nondimensionalization are taken as

$E_o = 30 \times 10^6$ lb/in.^{2}

$\sigma_o = 1000$ lb/in.^{2}

$a = 1000$ in.

$h_o = 1$ in.

Seven stations over the length of the meridian and four modes are used for the solution. For the purpose of illustration, only the first three Fourier harmonics of the applied load are used. Thus,

$$q^{(0)} = -15.0 \text{ lb/in.}^2$$

$$q^{(1)} = -19.1 \text{ lb/in.}^2$$

$$q^{(3)} = 6.37 \text{ lb/in.}^2$$

The boundary conditions are

$$U = V = W = \Phi_s = 0$$

This problem took .33 minutes of execution time on the NPS IBM 360/67 Computer using the FORTRAN IV, Level H, Compiler with OPT=2.

Example of a Dynamic Analysis

The second example is the dynamic analysis of a clamped truncated cone subjected to an impulsive loading which is uniform along the meridian and varies in a cosine distribution over one-half of the circumference. This problem is Sample Problem No. 3 in the series of sample problems suggested by the Lockheed Missiles and Space Co.. The initial conditions are

$$W = 0. \qquad 0 \le \theta \le 2\pi$$

$$dW/dT = -4440.8 \cos\theta \text{ (in./sec)} \qquad -\pi/2 \le \theta \le \pi/2$$

$$dW/dT = 0. \qquad \pi/2 \le \theta \le 3\pi/2$$

The physical parameters are

$\nu = .286$

meridian length = 15.004 in.

$r_{min} = 7.9499$ in.

$r_{max} = 10.2300$ in.

$R_s = \infty$

$R_\theta = r/\cos\omega$

$\omega = \sin^{-1}[(10.2300 - 7.9499)/15.004]$

$E = 3.52 \times 10^6$ lb/in.^{2}

$h = .543$ in.

$B = 2.0816 \times 10^6$ lb/in.

$D = 5.114 \times 10^4$ lb - in.

The time step is

$\Delta T = 2 \times 10^{-6}$ sec

The reference parameters for nondimensionalization are taken as

$E_o = 3.52 \times 10^6$ lb/in.^{2}

$\sigma_o = 1000$ lb/in.^{2}

$a = 15.004$ in.

$h_o = .543$ in.

$T_o = 10.965 \times 10^{-5}$ sec

Thirty-one stations over the length of the meridian and four modes are used for the solution. The first four Fourier harmonics of the initial conditions are

$dW^{(0)}/dT = -4440.8/\pi$ in./sec

$dW^{(1)}/dT = -4440.8/2$ in./sec

$dW^{(2)}/dT = -2 \times 4440.8/(3\pi)$ in./sec

$dW^{(4)}/dT = 2 \times 4440.8/(15\pi)$ in./sec

The boundary conditions are

$$U = V = W = \Phi_s = 0$$

This problem took approximately 8 minutes of execution time for 750 time steps on the NPS IBM 360/67 computer using the FORTRAN IV, Level H, Compiler with OPT=2.

Input Data Cards

Card	Columns	Format	Item	Static Example	Dynamic Example	Interpretation
1	2-72	18A4	TITLE			Problem description.
2	1-5	I5	NØ	10	108	The problem number.
2	6-10	I5	SØRD	0	1	Set to: 1. = 0 for static analysis; 2. ≥ 1 for dynamic analysis.
2	11-15	I5	IMØDE	1	0	Set to: 1. > 0 if modal data are desired for each harmonic; 2. ≤ 0 if modal data are not desired.
2	16-20	I5	NDIMEN	0	0	Set to: 1. = 0 if dimensional form of output data is desired; 2. ≥ 1 if nondimensional form of output data is desired.
2	21-25	I5	NTHMAX	1	2	The summed solution will be printed at NTHMAX meridians, 0 ≤ NTHMAX ≤ 6.
2	26-30	I5	IFREQ	1	13	The solution will be printed at meridional stations 1, IFREQ + 1, 2 * IFREQ + 1,...., and the final station.
2	31-35	I5	IPRINT	1	2	Every IPRINTth converged solution will be printed.
2	36-40	I5	IBCINL	-1	0	Set to: 1 < 0 if the shell has a pole at the first station; 2. ≥ 0 if the shell does not have a pole at the first station.
2	41-45	I5	IBCFNL	0	0	Set to: 1. < 0 if the shell has a pole at the final station; 2. ≥ 0 if the shell does not have a pole at the final station.
2	46-50	I5	KMAX	7	31	The number of meridional stations. The product of KMAX and MAXM (the number of Fourier terms in the solution) must be less than 201.

2	51-55	I5	MNMAX	3	4	Number of Fourier terms used to describe the initial conditions, the pressure loads, and the thermal loads, MNMAX ≤ MAXM.
2	56-60	I5	MAXM	4	4	Number of Fourier terms in the solution, MAXM ≤ 10 and (KMAX)*(MAXM) ≤ 201.
2	61-65	I5	LSMAX	99		<u>Static analysis</u> Maximum number of load intensities to be considered. For a nonlinear analysis this number should be large. For a linear analysis set LSMAX = 1.
					750	<u>Dynamic analysis</u> Maximum number of time increments to be considered, LSMAX=$T_{max}/\Delta T$.
2	66-70	I5	LCHMAX	2		<u>Static analysis</u> Maximum number of load increment reductions. Recommend value, 2-4.
					0	<u>Dynamic analysis</u> LCHMAX = 0
2	71-75	I5	ITRMAX	50	20	Maximum number of iterations at any load intensity or time step. Recommended value, 10-30. For a linear analysis set ITRMAX = 1.
2	76-80	I5	IC	0		<u>Static analysis</u> IC = 0
					1	<u>Dynamic analysis</u> Set to: 1. ≤ 0 if shell at rest at t = 0, or if restarting solution at t>0 , i.e. ITAPE = 2 or 3; 2. > 0 for non-zero initial conditions at t=0.
3	1-12	E12.3	NU	.3	.286	Poisson's ratio, ν.
3	13-24	E12.3	SIGØ	1000.	1000.	Reference stress level, σ_o. When the data is to be input in dimensional form set SIGØ = 1..

3	25-36	E12.3	ELAST	.3E8	.352E7	Reference modulus of elasticity, E_o. When the data is to be input in dimensional form set ELAST = 1..
3	37-48	E12.3	TKN	1.	.543	Reference thickness, h_o. When the data is to be input in dimensional form set TKN = 1..
3	49-60	E12.3	CHAR	1000.	15.004	Characteristic shell dimension, a. When the data is to be input in dimensional form set CHAR = 1..
3	61-72	E12.3	TEEØ	0.		<u>Static analysis</u> TEEØ = 0.
					10.965E-5	<u>Dynamic analysis</u> Reference time T_o.
4	1-12	E12.3	DELOAD	.2		<u>Static analysis</u> The load increment. DELOAD remains unchanged until the solution fails to converge in ITRMAX iterations. Then it is automatically reduced by a factor of 5. A maximum of LCHMAX reductions will occur provided the number of load intensities considered is less than LSMAX.
					1.82396E-2	<u>Dynamic analysis</u> The nondimensional time increment δt.
4	13-24	E12.3	EPS	.01	.01	The convergence criterion. Recommended value, .01.
5	1-5	I5	ITAPE	0		<u>Static analysis</u> ITAPE = 0 <u>Dynamic analysis</u> The parameter for obtaining the data to restart the solution at $t > 0$: 1. no read or write on tape, ITAPE = 0; 2. write Z, ZØ, Z2, and Z3 after final time step, ITAPE = 1; 3. read Z,ZØ,Z2, and Z3 before initial time step, ITAPE = 2; 4. read Z,ZØ,Z2, and Z3 before initial time step, and write Z,ZØ,Z2, and Z3 after final time step, ITAPE = 3.

6	1-72	6E12.3	0.	0. 3.14159	A list of NTHMAX circumferential coordinates θ, in radians, where print-out of the solution is desired. <u>This card is omitted if NTHMAX = 0.</u>

For a static analysis, the execution for each case terminates and the program transfers to the first read statement when either the number of load intensities considered equals LSMAX or the number of iterations equals ITRMAX and the number of previous DELØAD reductions equals LCHMAX.

For a dynamic analysis, the execution terminates when either LSMAX time increments have been taken or when the solution does not converge after ITRMAX iterations.

Restart option. When Z, ZØ, Z2 and Z3 have been put on tape unit #8 after the final time step (ITAPE = 1), the response computation can be restarted by mounting the recorded tape on unit #8, specifying ITAPE = 2 or 3, and inputting the identical data except for IC which must be zero. The following two cards are required for the NPS IBM 360/67:

```
//GØ.FT08F001 DD DSN=NØNLIN,UNIT=2400,VØL=SER=NPS104,

// DCB=CRECFM=VS,LRECL=3204,BLKSIZE=3208 ,DISP=(NEW,KEEP),LABEL=(,SL)
```

The boundary conditions are read in on cards. If the shell does not have a pole at the first station, IBCINL ≥ 0, and cards 7-15 describe the boundary conditions at the first station. <u>However, if the shell does have a pole at the first station, IBCINL < 0, and cards 7-15 are omitted</u>. Cards 7-15 have the format 4E16.8 and correspond to the boundary conditions as follows:

$$\begin{array}{cccc} \text{Card 7,16} & \text{Card 8,17} & \text{Card 9,18} & \text{Card 10,19} \end{array}$$
$$\begin{bmatrix} \bar{\Omega}(1,1) & \bar{\Omega}(1,2) & \bar{\Omega}(1,3) & \bar{\Omega}(1,4) \\ \bar{\Omega}(2,1) & \bar{\Omega}(2,2) & \bar{\Omega}(2,3) & \bar{\Omega}(2,4) \\ \bar{\Omega}(3,1) & \bar{\Omega}(3,2) & \bar{\Omega}(3,3) & \bar{\Omega}(3,4) \\ \bar{\Omega}(4,1) & \bar{\Omega}(4,2) & \bar{\Omega}(4,3) & \bar{\Omega}(4,4) \end{bmatrix} \begin{bmatrix} N_s \\ \hat{N}_{s\theta} \\ \hat{Q}_s \\ \Phi_s \end{bmatrix} +$$

$$
\begin{array}{ccccccc}
\text{Card 11,20} & \text{Card 12,21} & \text{Card 13,22} & \text{Card 14,23} & & & \text{Card 15,24} \\
\end{array}
$$

$$
\begin{bmatrix}
\bar{\Lambda}(1,1) & \bar{\Lambda}(1,2) & \bar{\Lambda}(1,3) & \bar{\Lambda}(1,4) \\
\bar{\Lambda}(2,1) & \bar{\Lambda}(2,2) & \bar{\Lambda}(2,3) & \bar{\Lambda}(2,4) \\
\bar{\Lambda}(3,1) & \bar{\Lambda}(3,2) & \bar{\Lambda}(3,3) & \bar{\Lambda}(3,4) \\
\bar{\Lambda}(4,1) & \bar{\Lambda}(4,2) & \bar{\Lambda}(4,3) & \bar{\Lambda}(4,4)
\end{bmatrix}
\begin{bmatrix} U \\ V \\ W \\ M_s \end{bmatrix}
=
\begin{bmatrix} \ell(1) \\ \ell(2) \\ \ell(3) \\ \ell(4) \end{bmatrix}
$$

If the shell does not have a pole at the final station, IBCFNL ≥ 0, and cards 16-24 describe the boundary conditions at the final station. The format and correspondence are the same as for the boundary conditions at the first station given above. <u>However, if the shell does have a pole at the final station, IBCFNL < 0, and cards 16-24 are omitted.</u> Note that the boundary conditions are on the total variables and not on the individual modes. Thus, it is not possible to have different boundary conditions for each mode without modifying the program. An example of the modifications required to change ℓ is given in reference 5. Furthermore, note that the boundary conditions are input in dimensional form.

User-Prepared Subroutines

The geometry of the shell, the inplane and bending stiffnesses of the shell, the applied pressure and thermal loads, and the initial conditions are introduced to the program through the use of the five subroutines GEØM, BDB (K, B, DB, D, DD), PLØAD(K), TLØAD(K) and INITL. This section describes each of these subroutines.

1. <u>GEØM</u>

The nondimensional quantities Δ, ρ, γ, ω_θ, ω_s, $d\omega_s/d\xi$, and μ are defined in GEØM as a function of the meridional station number K. The correspondence between the nondimensional variables and the FØRTRAN variables is as follows:

$$\text{DEL} = \Delta = (\text{meridian length})/[a(\text{KMAX} - 1)]$$

$$
\left.
\begin{aligned}
&\text{R(K)} = (\rho)_K = (r/a)_K \\
&\text{GAM(K)} = (\gamma)_K = \left(\frac{d\rho/d\xi}{\rho}\right)_K = \left(\frac{dr/ds}{r/a}\right)_K \\
&\text{ØMT(K)} = (\omega_\theta)_K = (a/R_\theta)_K \\
&\text{ØMXI(K)} = (\omega_s)_K = (a/R_s)_K \\
&\text{DEØMX(K)} = \left(\frac{d\omega_s}{d\xi}\right)_K = \left[a^2\,\frac{d(1/Rs)}{ds}\right]_K \\
&\text{MASS(K)} = (\mu)_K = \frac{a^2}{h_o E_o T_o^{\,2}}\left(\int m\, d\zeta\right)_K
\end{aligned}
\right\}
\quad K = 1,2,\ \ldots\ \text{KMAX}
$$

The statements for the static example are

```
      DEL=(105./6.)/1000.
      DØ 4 K=2,KMAX
      RK = K
      THET = (RK-1.)*DEL
      R(K) = SIN(THET)
      GAM(K) = CØS(THET)/R(K)
      ØMT(K)=1.
      ØMXI(K)=1.
    4 DEØMX(K) = 0.
      R(1) = 0.
      GAM(1) = 0.
      ØMT(1)=1.
      ØMXI(1)=1.
      DEØMX(1) = 0.
```

The statements for the dynamic example are

```
      AKX=KMAX-1
      DEL=1./AKX
      THET=ARSIN(2.2801/CHAR)
      DØ 11 K=1,KMAX
      AK=K
      R(K)=(7.9499+(AK-1.)*(2.2801)/AKX)/CHAR
      GAM(K)=(2.2801/CHAR)/R(K)
      ØMXI(K)=0.
      DEØMX(K)=0.
      ØMT(K)=CØS(THET)/R(K)
      MASS(K)=1.
   11 CØNTINUE
```

2. BDB (K, B, DB, D, DD)

The nondimensional stiffness quantities b, $db/d\xi$, d, and $dd/d\xi$ are defined in BDB for each meridional station. The correspondence between the stiffness quantities at the Kth station and the FØRTRAN variables is as follows:

$$B = (b)_K = (B)_K/(E_o h_o)$$

$$DB = \left(\frac{db}{d\xi}\right)_K = \left(\frac{dB}{ds}\right)_K \left(\frac{a}{E_o h_o}\right)$$

$$D = (d)_K = (D)_K/(E_o h_o^3)$$

$$DD = \left(\frac{dd}{d\xi}\right)_K = \left(\frac{dD}{ds}\right)_K \left(\frac{a}{E_o h_o^3}\right)$$

where

$$B = \int E d\zeta/(1-\nu^2)$$

$$D = \int \zeta^2 E d\zeta/(1-\nu^2)$$

The statements for the static example are

```
B=27.3E+06/(30.E+06*1.*(1.-.09))
D=B/12.
DB=0.
DD=0.
```

The statements for the dynamic example are

```
B=1.089082
D=.09075683
DB=0.
DD=0.
```

3. PLØAD(K)

The nondimensional Fourier coefficients of the meridional, circumferential and normal components of the pressure load, $p_s^{(n)}$, $p_\theta^{(n)}$, and $p^{(n)}$ respectively, are defined in PLØAD for each meridional station as a function of the Fourier index. In addition, the array of Fourier integer numbers n is defined here. The relationship between these quantities at the Kth station and FØRTRAN variables is as follows:

$$NN(M) = n$$

$$PX(M) = (p_s^{(n)})_K = (q_s^{(n)})_K \, (a/\sigma_o h_o)$$

$$PT(M) = (p_\theta^{(n)})_K = (q_\theta^{(n)})_K \, (a/\sigma_o h_o)$$

M = 1, 2, ... MNMAX

$$PR(M) = (p^{(n)})_K = (q^{(n)})_K \; (a/\sigma_o h_o)$$

Note that these are stored as functions of M only.

The statements for the static example are

```
NN(1)=0
NN(2)=1
NN(3)=3
PR(1)=-15.
PR(2)=-19.1
PR(3)=6.37
```

No statements are required for the dynamic example. The array of mode numbers is included in the subroutine INITL.

4. <u>TLØAD(K)</u>

The nondimensional Fourier coefficients of the thermal loads $t_T^{(n)}$, $m_T^{(n)}$, $\frac{d}{d\xi}\,(t_T^{(n)})$ and $\frac{d}{d\xi}\;(m_T^{(n)})$ are defined in TLØAD(K) for each meridional station as a function of the Fourier index. The FØRTRAN variables are defined as follows:

$$TT(M) = (t_T^{(n)})_K = (\epsilon_T^{(n)})_K/(\sigma_o h_o)$$

$$DT(M) = (\frac{d}{d\xi}\,(t_T^{(n)}))_K = (\frac{d\epsilon_T^{(n)}}{ds})_K(a/\sigma_o h_o)$$

$$EMT(M) = (m_T^{(n)})_K = (\kappa_T^{(n)})_K\;(a/\sigma_o h_o^3)$$

$$DMT(M) = (\frac{d}{d\xi}(m_T^{(n)}))_K = (\frac{dm_T^{(n)}}{ds})\;(a^2/\sigma_o h_o^3)$$

Note that these are stored as functions of M only. If only thermal loads are applied the array of Fourier interger numbers can be introduced in TLØAD(K) instead of PLØAD(K).

5. <u>INITL</u>

This subroutine introduces the initial conditions of the nondimensional solution vector z for all the stations, including the ficticious stations off the ends of the shell, and all the modes. The FØRTRAN variables are defined as follows:

$$
\left.
\begin{aligned}
Z(I,L) &= (z^{(n)})_K = \begin{bmatrix} U^{(n)}(E_o/a\sigma_o) \\ V^{(n)}(E_o/a\sigma_o) \\ W^{(n)}(E_o/a\sigma_o) \\ M_s^{(n)}(a/\sigma_o h_o{}^3) \end{bmatrix}_K \\
Z\text{Ø}T(I,L) &= \left(\frac{dz^{(n)}}{dt}\right)_K = T_o \frac{d}{dT} \begin{bmatrix} U^{(n)}(E_o/a\sigma_o) \\ V^{(n)}(E_o/a\sigma_o) \\ W^{(n)}(E_o/a\sigma_o) \\ M_s^{(n)}(a/\sigma_o h_o{}^3) \end{bmatrix}_K
\end{aligned}
\right\}
\begin{aligned}
&I = 1,2,3,4 \\
&L = 1,2,\ \ldots \\
&\quad (KMAX+2)*(MNMAX)
\end{aligned}
$$

The index L runs from 1 to KMAX+2 for NN(1), and from 1+KMAX+2 to 2*(KMAX+2) for NN(2), etc. The first element for each value of n corresponds to the initial ficticious station, the next element corresponds to the first station on the shell, etc.

The statements for the dynamic example are

```
      NN(1)=0
      NN(2)=1
      NN(3)=2
      NN(4)=4
      PI=3.14159
      DØ 2 M=1,MAXM
      IF(M.EQ.1) VEL=-444.08/PI
      IF(M.EQ.2) VEL=-444.08/2.
      IF(M.EQ.3) VEL=-444.08*2./(3.*PI)
      IF(M.EQ.4) VEL=444.08*2./(15.*PI)
      DØ 2 K=2,KL
      I=K+1+(M-1)*KMAX2
    2 ZDØT(3,I)=VEL*ELAST*TEEØ/(CHAR*SIGØ)*10.
```

in which KL = KMAX-1 and KMAX2 = KMAX+2.

Output Format

The output from the program consists of the boundary conditions $\overline{\Omega}$, $\overline{\Lambda}$, and ℓ at each end of the shell; the input parameters, such as the number of stations, the number of modes, etc.; and the circumferential coordinates where a summed solution is desired. The remainder of the output can appear in either dimensional or nondimensional form. The correspondence between the printed FØRTRAN variables and the dimensional and nondimensional dependent variables is given below. For the shell geometry, the following are printed at each station:

RADIUS	-	r	or	ρ
GAMMA	-	$\dfrac{dr/ds}{r}$	or	$\dfrac{d\rho/\rho\xi}{\rho}$
ØMEGA S	-	$1/R_s$	or	ω_s
ØMEGA THETA	-	$1/R_\theta$	or	ω_θ
DEØMEGA S	-	$\dfrac{d}{ds}(1/R_s)$	or	$\dfrac{d\omega_s}{d\xi}$
MASS	-	$\int m d\zeta$	or	μ

For the inplane and bending stiffnesses, the following are printed at each station:

B STIFFNESS	-	B	or	b
D STIFFNESS	-	D	or	d
B PRIME	-	$\dfrac{dB}{ds}$	or	$\dfrac{db}{d\xi}$
D PRIME	-	$\dfrac{dD}{ds}$	or	$\dfrac{dd}{d\xi}$

For the pressure and thermal loads, the following are printed at each station for each value of n for the static analysis:

N	-	Fourier index n		
PR	-	$q^{(n)}$	or	$p^{(n)}$
PX	-	$q_s^{(n)}$	or	$p_s^{(n)}$
PT	-	$q_\theta^{(n)}$	or	$p_\theta^{(n)}$
TT	-	$e_T^{(n)}$	or	$t_T^{(n)}$

MT - $\varkappa_T^{(n)}$ or $m_T^{(n)}$

DTT - $\dfrac{d\epsilon_T^{(n)}}{ds}$ or $\dfrac{dt_T^{(n)}}{d\xi}$

DMT - $\dfrac{d\varkappa_T^{(n)}}{ds}$ or $\dfrac{dm_T^{(n)}}{d\xi}$

The following initial conditions are printed at each station for each value of n for non zero initial conditions in a dynamic analysis:

U - $U^{(n)}$ or $u^{(n)}$
V - $V^{(n)}$ or $v^{(n)}$
W - $W^{(n)}$ or $w^{(n)}$
M S - $M_s^{(n)}$ or $m_s^{(n)}$

U DØT - $dU^{(n)}/dT$ or $du^{(n)}/dt$
V DØT - $dV^{(n)}/dT$ or $dv^{(n)}/dt$
W DØT - $dW^{(n)}/dT$ or $dw^{(n)}/dt$
M S DØT - $dM_s^{(n)}/dT$ or $dm_s^{(n)}/dt$

Note that station 1 refers to the station on the initial boundary and not the ficticious station.

Every IPRINTth solution is printed with the corresponding load factor and the number of iterations. The definitions of the printed quantities preceeding the solution are:

LØAD STEP NUMBER - the number of load intensities considered.

TIME STEP NUMBER - the number of time steps taken.

LØAD FACTØR - the proportion of the loads given in PLØAD, TLØAD and ℓ currently on the shell.

TIME - both nondimensional and dimensional time are given.

ITERATIØNS - the number of iterations required for convergence.

The correspondence between the printed terms and the dimensional and nondimensional forces, moments, displacements and rotations is as follows:

N S - N_s or t_s
N THETA - N_θ or t_θ
N STHETA - $N_{s\theta}$ or $t_{s\theta}$

Q S	-	Q_s	or	f_s
M S	-	M_s	or	m_s
M THETA	-	M_θ	or	m_θ
M STHETA	-	$M_{s\theta}$	or	$m_{s\theta}$
U	-	U	or	u
V	-	V	or	v
W	-	W	or	w
PHI S	-	Φ_s	or	φ_s
PHI THETA	-	Φ_θ	or	φ_θ
PHI	-	Φ	or	φ

Sample Solutions

The printed input data and solution for the static analysis example is given in figure 6 for the load factor .744. The load-displacement plot is given in figure 7 for the displacement at the pole (station 1) and station 3 for $\theta = 0^\circ$. The printed input data and solution for the dynamic analysis example is given in figure 8 for T = 500 μsec. The time history of the normal displacement at $s = 6.5$ in. and $\theta = 0^\circ$ is given in figure 9.

--PROBLEM NUMBER 10--

SAMPLE PROBLEM 10 - UNSYMMETRICALLY LOADED SPHERICAL CAP, TEST CASE.

--INPUT DATA RECORD--

THE BOUNDARY CONDITIONS ARE:

THE SHELL HAS AN INITIAL POLE

AT THE FINAL EDGE

```
 ---------------OMEGA BAR----------------                        ---------------LAMBDA BAR---------------           ---EL-----
( 0.0        0.0        0.0        0.0         )  N S        ( 0.100E 01  0.0        0.0        0.0   )  U           0.0
( 0.0        0.0        0.0        0.0         )  N ST   +   ( 0.0        0.100E 01  0.0        0.0   )  V     =     0.0
( 0.0        0.0        0.0        0.0         )  Q S        ( 0.0        0.0        0.100E 01  0.0   )  W           0.0
( 0.0        0.0        0.0        0.100E 01)     PHI S      ( 0.0        0.0        0.0        0.0   )  M S         0.0
```



```
NUMBER OF STATIONS----------------------  7
NUMBER OF MODES-------------------------  4
INCREMENTAL LOAD FACTOR-----------------  0.200
MAXIMUM NUMBER OF LOAD STEPS------------  99
MAXIMUM NUMBER OF ITERATIONS------------  50
MAXIMUM NUMBER OF LOAD FACTOR CHANGES---  2
CONVERGENCE CRITERION-------------------  0.010

CHARACTERISTIC SHELL DIMENSION---   0.1000E 04
REFERENCE THICKNESS--------------   0.1000E 01
REFERENCE ELASTICITY-------------   0.3000E 08
REFERENCE STRESS-----------------   0.1000E 04
POISSON'S RATIO------------------   0.3000E 00
```

CIRCUMFERENTIAL COORDINATES FOR PRINT RECORD, IN RADIANS, ARE:

0.0 ,

THE DATA IS IN DIMENSIONAL FORM

Figure 6. Input Data and Solution for the Static Analysis Example

STATION	RADIUS	GAMMA	OMEGA S	OMEGA THETA	DEOMEGA S
1	0.0	0.0	0.1000E-02	0.1000E-02	0.0
2	0.1750E 02	0.5714E-01	0.1000E-02	0.1000E-02	0.0
3	0.3499E 02	0.2856E-01	0.1000E-02	0.1000E-02	0.0
4	0.5248E 02	0.1903E-01	0.1000E-02	0.1000E-02	0.0
5	0.6994E 02	0.1426E-01	0.1000E-02	0.1000E-02	0.0
6	0.8739E 02	0.1140E-01	0.1000E-02	0.1000E-02	0.0
7	0.1048E 03	0.9489E-02	0.1000E-02	0.1000E-02	0.0

STATION	B STIFFNESS	D STIFFNESS	B PRIME	D PRIME
1	0.300000E 08	0.250000E 07	0.0	0.0
2	0.300000E 08	0.250000E 07	0.0	0.0
3	0.300000E 08	0.250000E 07	0.0	0.0
4	0.300000E 08	0.250000E 07	0.0	0.0
5	0.300000E 08	0.250000E 07	0.0	0.0
6	0.300000E 08	0.250000E 07	0.0	0.0
7	0.300000E 08	0.250000E 07	0.0	0.0

PRESSURE AND TEMPERATURE COEFFICIENTS FOR N= 0 FOLLOW

STATION	PR	PX	PT	TT	MT	DTT	DMT
1	-0.1500E 02	0.0	0.0	0.0	0.0	0.0	0.0
2	-0.1500E 02	0.0	0.0	0.0	0.0	0.0	0.0
3	-0.1500E 02	0.0	0.0	0.0	0.0	0.0	0.0
4	-0.1500E 02	0.0	0.0	0.0	0.0	0.0	0.0
5	-0.1500E 02	0.0	0.0	0.0	0.0	0.0	0.0
6	-0.1500E 02	0.0	0.0	0.0	0.0	0.0	0.0
7	-0.1500E 02	0.0	0.0	0.0	0.0	0.0	0.0

PRESSURE AND TEMPERATURE COEFFICIENTS FOR N= 1 FOLLOW

STATION	PR	PX	PT	TT	MT	DTT	DMT
1	-0.1910E 02	0.0	0.0	0.0	0.0	0.0	0.0
2	-0.1910E 02	0.0	0.0	0.0	0.0	0.0	0.0
3	-0.1910E 02	0.0	0.0	0.0	0.0	0.0	0.0
4	-0.1910E 02	0.0	0.0	0.0	0.0	0.0	0.0
5	-0.1910E 02	0.0	0.0	0.0	0.0	0.0	0.0
6	-0.1910E 02	0.0	0.0	0.0	0.0	0.0	0.0
7	-0.1910E 02	0.0	0.0	0.0	0.0	0.0	0.0

PRESSURE AND TEMPERATURE COEFFICIENTS FOR N= 3 FOLLOW

STATION	PR	PX	PT	TT	MT	DTT	DMT
1	0.6370E 01	0.0	0.0	0.0	0.0	0.0	0.0
2	0.6370E 01	0.0	0.0	0.0	0.0	0.0	0.0
3	0.6370E 01	0.0	0.0	0.0	0.0	0.0	0.0
4	0.6370E 01	0.0	0.0	0.0	0.0	0.0	0.0
5	0.6370E 01	0.0	0.0	0.0	0.0	0.0	0.0
6	0.6370E 01	0.0	0.0	0.0	0.0	0.0	0.0
7	0.6370E 01	0.0	0.0	0.0	0.0	0.0	0.0

Figure 6. Continued

THE LOAD STEP NUMBER IS 9 THE LOAD FACTOR IS 0.7440E 00 THE SOLUTION CONVERGED IN 10 ITERATIONS

THE SUMMED FORCES, MOMENTS, DISPLACEMENTS AND ROTATIONS FOLLOW FOR THETA = 0.0

STATION	N S	N THETA	N STHETA	Q S	M S	M THETA	M STHETA
1	-0.7200E 04	-0.4576E 04	0.0	-0.1038E 03	0.2211E 03	0.1303E 03	0.0
2	-0.1384E 05	-0.1361E 05	0.0	-0.6181E 02	-0.1626E 04	-0.1565E 04	0.0
3	-0.1517E 05	-0.2011E 05	0.0	-0.3293E 02	-0.2913E 04	-0.1930E 04	0.0
4	-0.1369E 05	-0.1995E 05	0.0	0.5189E 02	-0.2613E 04	-0.1828E 04	0.0
5	-0.9044E 04	-0.1389E 05	0.0	0.1127E 03	-0.1166E 04	-0.1205E 04	0.0
6	-0.5843E 04	-0.5893E 04	0.0	0.1765E 03	0.9917E 03	-0.2303E 03	0.0
7	-0.1257E 04	-0.3771E 03	0.0	0.2868E 03	0.4425E 04	0.1328E 04	0.0

STATION	U	V	W	PHI S	PHI THETA	PHI
1	-0.2584E-01	0.0	-0.2880E 00	0.2900E-01	0.0	0.0
2	-0.2947E-01	0.0	-0.7308E 00	0.2082E-01	0.0	0.0
3	-0.2054E-01	0.0	-0.1018E 01	0.7398E-02	0.0	0.0
4	-0.6615E-02	0.0	-0.9904E 00	-0.9517E-02	0.0	0.0
5	0.2747E-02	0.0	-0.6849E 00	-0.2055E-01	0.0	0.0
6	0.3677E-02	0.0	-0.2711E 00	-0.1957E-01	0.0	0.0
7	-0.5960E-08	0.0	-0.3517E-06	-0.5058E-08	0.0	0.0

MODAL OUTPUT FOR MODE N = 0 FOLLOWS

STATION	N S	N THETA	N STHETA	Q S	M S	M THETA	M STHETA
1	-0.5888E 04	-0.5888E 04	0.0	0.0	0.1757E 03	0.1757E 03	0.0
2	-0.6410E 04	-0.6353E 04	0.0	-0.1433E 02	0.1757E 03	0.1222E 03	0.0
3	-0.7064E 04	-0.6914E 04	0.0	-0.2719E 02	-0.4328E 03	-0.1567E 03	0.0
4	-0.6727E 04	-0.6393E 04	0.0	0.1607E 01	-0.4999E 03	-0.3003E 03	0.0
5	-0.5578E 04	-0.4661E 04	0.0	0.2172E 02	-0.2436E 03	-0.2577E 03	0.0
6	-0.4604E 04	-0.2505E 04	0.0	0.4516E 02	0.2534E 03	-0.6102E 02	0.0
7	-0.3185E 04	-0.9554E 03	0.0	0.8682E 02	0.1211E 04	0.3634E 03	0.0

STATION	U	V	W	PHI S	PHI THETA	PHI
1	0.0	0.0	-0.2880E 00	0.0	0.0	0.0
2	-0.2918E-03	0.0	-0.2880E 00	0.5343E-03	0.0	0.0
3	0.1396E-02	0.0	-0.3067E 00	-0.4142E-03	0.0	0.0
4	0.3637E-02	0.0	-0.2735E 00	-0.3473E-02	0.0	0.0
5	0.4338E-02	0.0	-0.1850E 00	-0.5689E-02	0.0	0.0
6	0.2750E-02	0.0	-0.7421E-01	-0.5284E-02	0.0	0.0
7	0.6209E-09	0.0	-0.6358E-07	0.1817E-08	0.0	0.0

Figure 6. Continued

MODAL OUTPUT FOR MODE N = 1 FOLLOWS

STATION	N S	N THETA	N STHETA	Q S	M S	M THETA	M STHETA
1	0.0	0.0	0.0	-0.1038E 03	0.0	0.0	0.0
2	-0.6742E 04	-0.7480E 04	-0.1838E 04	-0.5941E 02	-0.1816E 04	-0.1264E 04	0.5525E 03
3	-0.7326E 04	-0.1222E 05	-0.4592E 04	0.1858E 01	-0.2080E 04	-0.1399E 04	0.5952E 03
4	-0.6906E 04	-0.1234E 05	-0.5566E 04	0.3830E 02	-0.1665E 04	-0.1244E 04	0.5720E 03
5	-0.4790E 04	-0.8735E 04	-0.5549E 04	0.6816E 02	-0.8401E 03	-0.8640E 03	0.4709E 03
6	-0.2723E 04	-0.3607E 04	-0.4701E 04	0.1218E 03	0.4724E 03	-0.2416E 03	0.2954E 03
7	0.6538E 03	0.1962E 03	-0.3387E 04	0.2270E 03	0.3019E 04	0.9058E 03	-0.2821E 00

STATION	U	V	W	PHI S	PHI THETA	PHI
1	-0.2584E-01	0.2584E-01	0.0	0.2900E-01	-0.2900E-01	0.0
2	-0.2584E-01	0.2870E-01	-0.4113E 00	0.1795E-01	-0.2347E-01	0.8296E-04
3	-0.1848E-01	0.2593E-01	-0.6290E 00	0.6036E-02	-0.1795E-01	-0.1298E-04
4	-0.8680E-02	0.2035E-01	-0.6232E 00	-0.5318E-02	-0.1185E-01	-0.7002E-04
5	-0.1416E-02	0.1327E-01	-0.4432E 00	-0.1252E-01	-0.6323E-02	-0.1175E-03
6	0.9771E-03	0.6213E-02	-0.1849E 00	-0.1266E-01	-0.2110E-02	-0.1485E-03
7	-0.5588E-08	0.1192E-07	-0.3497E-06	-0.8180E-08	-0.3324E-08	-0.1613E-03

MODAL OUTPUT FOR MODE N = 3 FOLLOWS

STATION	N S	N THETA	N STHETA	Q S	M S	M THETA	M STHETA
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.8360E 03	-0.6996E 03	-0.7687E 03	0.8236E 01	-0.3132E 02	0.2614E 03	0.1267E 03
3	0.1211E 04	-0.4578E 03	-0.8223E 03	0.4883E 01	0.1137E 03	0.3885E 03	0.6502E 02
4	0.1499E 04	0.2354E 03	-0.3732E 03	0.1461E 01	0.2192E 03	0.3771E 03	-0.1111E 02
5	0.1669E 04	0.7190E 03	0.1172E 03	-0.6189E 01	0.2923E 03	0.3210E 03	-0.8573E 02
6	0.1137E 04	0.5551E 03	0.4111E 03	-0.2778E 02	0.1456E 03	0.1705E 03	-0.1129E 03
7	0.1267E 03	0.3796E 02	0.3961E 03	-0.6937E 02	-0.5344E 03	-0.1603E 03	0.3286E-01

STATION	U	V	W	PHI S	PHI THETA	PHI
1	0.0	0.0	0.0	0.0	0.0	0.0
2	-0.3575E-04	0.6604E-04	0.5479E-02	-0.7353E-03	0.9394E-03	-0.2211E-05
3	0.1968E-03	-0.7231E-04	0.2573E-01	-0.1168E-02	0.2206E-02	-0.1396E-05
4	0.4647E-03	-0.5498E-03	0.4638E-01	-0.7714E-03	0.2651E-02	-0.4023E-05
5	0.3153E-03	-0.9175E-03	0.5275E-01	0.3901E-03	0.2262E-02	-0.1653E-05
6	-0.3925E-04	-0.6808E-03	0.3274E-01	0.1507E-02	0.1123E-02	0.8553E-05
7	-0.1242E-09	0.2111E-08	0.6358E-07	0.2214E-08	0.1822E-08	0.1886E-04

Figure 6. Continued

MODAL OUTPUT FOR MODE N =   2 FOLLOWS

STATION	N S	N THETA	N STHETA	Q S	M S	M THETA	M STHETA
1	-0.1312E 04	0.1312E 04	0.1312E 04	0.0	0.4539E 02	-0.4539E 02	-0.4539E 02
2	-0.1521E 04	0.9194E 03	0.1148E 04	0.3691E 01	0.4539E 02	-0.6848E 03	-0.1929E 03
3	-0.1994E 04	-0.5224E 03	0.3649E 03	-0.1249E 02	-0.5140E 03	-0.7631E 03	0.1307E 02
4	-0.1552E 04	-0.1455E 04	-0.4760E 03	0.1053E 02	-0.6669E 03	-0.6609E 03	0.1746E 03
5	-0.3456E 03	-0.1214E 04	-0.9906E 03	0.2905E 02	-0.3745E 03	-0.4043E 03	0.2146E 03
6	0.3471E 03	-0.3367E 03	-0.9952E 03	0.3735E 02	0.1203E 03	-0.9817E 02	0.1456E 03
7	0.1148E 04	0.3441E 03	-0.6130E 03	0.4230E 02	0.7292E 03	0.2188E 03	-0.5105E-01

STATION	U	V	W	PHI S	PHI THETA	PHI
1	0.0	0.0	0.0	0.0	0.0	0.0
2	-0.3300E-02	0.3716E-02	-0.3699E-01	0.3076E-02	-0.4224E-02	-0.3635E-05
3	-0.3653E-02	0.5516E-02	-0.1078E 00	0.2945E-02	-0.6154E-02	-0.6177E-05
4	-0.2036E-02	0.5078E-02	-0.1402E 00	0.4624E-04	-0.5337E-02	-0.2285E-04
5	-0.4905E-03	0.3250E-02	-0.1095E 00	-0.2730E-02	-0.3127E-02	-0.3707E-04
6	-0.1098E-04	0.1352E-02	-0.4466E-01	-0.3128E-02	-0.1021E-02	-0.3884E-04
7	-0.8692E-09	0.1614E-08	-0.1987E-08	-0.9091E-09	-0.3630E-10	-0.2919E-04

Figure 6. Continued

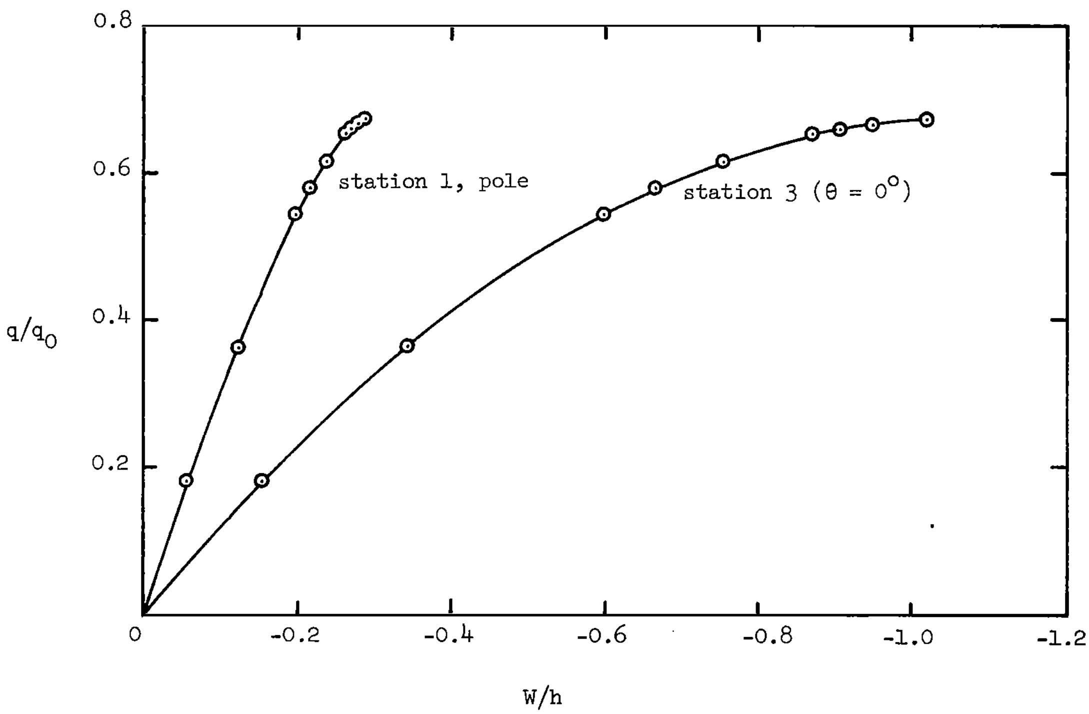

Figure 7. Load-Displacement Curves for Static Analysis Example Problem

```
                                        --PROBLEM NUMBER 108--

                                  LMSC TEST CASE, IMPULSIVELY LOADED CONE

                                        --INPUT DATA RECORD--
     THE BOUNDARY CONDITIONS ARE:

          AT THE INITIAL EDGE
   ---------------OMEGA BAR----------------             ---------------LAMBDA BAR---------------           ---EL-----

  ( 0.0       0.0       0.0       0.0        )  N S        ( 0.100E 01 0.0       0.0        0.0       )  U          0.0
  ( 0.0       0.0       0.0       0.0        )  N ST   +   ( 0.0       0.100E 01 0.0        0.0       )  V     =    0.0
  ( 0.0       0.0       0.0       0.0        )  Q S        ( C.0       0.C       0.100E 01  0.0       )  W          0.0
  ( 0.0       0.0       C.0       0.100E 01)    PHI S      ( 0.0       0.0       0.0        0.0       )  M S        0.0
          AT THE FINAL EDGE
   ---------------OMEGA BAR----------------             ---------------LAMBDA BAR---------------           ---EL-----

  ( 0.0       0.0       0.0       0.0        )  N S        ( 0.100E 01 0.0       0.0        0.0       )  U          0.0
  ( 0.0       0.0       0.0       0.0        )  N ST   +   ( 0.0       0.100E 01 0.0        0.0       )  V     =    0.0
  ( 0.0       0.0       0.0       0.0        )  Q S        ( 0.0       0.0       0.100E 01  0.0       )  W          0.0
  ( 0.0       0.0       0.0       0.100E 01)    PHI S      ( 0.0       0.C       0.0        0.0       )  M S        0.0


     NUMBER OF STATIONS---------------------- 31
     NUMBER OF MODES------------------------- 4
     INCREMENTAL TIME------------------------0.182E-01
     MAXIMUM NUMBER OF TIME STEPS------------ 750
     MAXIMUM NUMBER OF ITERATIONS------------ 10
     CONVERGENCE CRITERION------------------- 0.010

     CHARACTERISTIC SHELL DIMENSION---  0.1500E 02
     REFERENCE THICKNESS--------------  0.5430E 00
     REFERENCE ELASTICITY-------------  0.3520E 07
     REFERENCE STRESS-----------------  0.1000E C4
     REFERENCE TIME-------------------  0.1097E-03
     POISSON'S RATIO------------------  0.2860E 00

     CIRCUMFERENTIAL COORDINATES FOR PRINT RECORD, IN RADIANS, ARE:

              0.0          ,     0.314159E 01 ,



                                   THE DATA IS IN DIMENSIONAL FORM
                                   -------------------------------
```

Figure 8. Input Data and Solution for the Dynamic Analysis Example

STATION	RADIUS	GAMMA	OMEGA S	OMEGA THETA	DEOMEGA S	MASS
1	0.7950E 01	0.1912E-01	0.0	0.1243E 00	0.0	0.1021E-03
2	0.8026E 01	0.1893E-01	0.0	0.1231E 00	0.0	0.1021E-03
3	0.8102E 01	0.1876E-01	0.0	0.1220E 00	0.0	0.1021E-03
4	0.8178E 01	0.1858E-01	0.0	0.1209E 00	0.0	0.1021E-03
5	0.8254E 01	0.1841E-01	0.0	0.1197E 00	0.0	0.1021E-03
6	0.8330E 01	0.1824E-01	0.0	0.1187E 00	0.0	0.1021E-03
7	0.8406E 01	0.1808E-01	0.0	0.1176E 00	0.0	0.1021E-03
8	0.8482E 01	0.1792E-01	0.0	0.1165E 00	0.0	0.1021E-03
9	0.8558E 01	0.1776E-01	0.0	0.1155E 00	0.0	0.1021E-03
10	0.8634E 01	0.1760E-01	0.0	0.1145E 00	0.0	0.1021E-03
11	0.8710E 01	0.1745E-01	0.0	0.1135E 00	0.0	0.1021E-03
12	0.8786E 01	0.1730E-01	0.0	0.1125E 00	0.0	0.1021E-03
13	0.8862E 01	0.1715E-01	0.0	0.1115E 00	0.0	0.1021E-03
14	0.8938E 01	0.1700E-01	0.0	0.1106E 00	0.0	0.1021E-03
15	0.9014E 01	0.1686E-01	0.0	0.1097E 00	0.0	0.1021E-03
16	0.9090E 01	0.1672E-01	0.0	0.1087E 00	0.0	0.1021E-03
17	0.9166E 01	0.1658E-01	0.0	0.1078E 00	0.0	0.1021E-03
18	0.9242E 01	0.1644E-01	0.0	0.1069E 00	0.0	0.1021E-03
19	0.9318E 01	0.1631E-01	0.0	0.1061E 00	0.0	0.1021E-03
20	0.9394E 01	0.1618E-01	0.0	0.1052E 00	0.0	0.1021E-03
21	0.9470E 01	0.1605E-01	0.0	0.1044E 00	0.0	0.1021E-03
22	0.9546E 01	0.1592E-01	0.0	0.1035E 00	0.0	0.1021E-03
23	0.9622E 01	0.1579E-01	0.0	0.1027E 00	0.0	0.1021E-03
24	0.9698E 01	0.1567E-01	0.0	0.1019E 00	0.0	0.1021E-03
25	0.9774E 01	0.1555E-01	0.0	0.1011E 00	0.0	0.1021E-03
26	0.9850E 01	0.1543E-01	0.0	0.1003E 00	0.0	0.1021E-03
27	0.9926E 01	0.1531E-01	0.0	0.9958E-01	0.0	0.1021E-03
28	0.1000E 02	0.1519E-01	0.0	0.9882E-01	0.0	0.1021E-03
29	0.1008E 02	0.1508E-01	0.0	0.9807E-01	0.0	0.1021E-03
30	0.1015E 02	0.1497E-01	0.0	0.9734E-01	0.0	0.1021E-03
31	0.1023E 02	0.1485E-01	0.0	0.9662E-01	0.0	0.1021E-03

STATION	B STIFFNESS	D STIFFNESS	B PRIME	D PRIME
1	0.208163E 07	0.511471E 05	0.0	0.0
2	0.208163E 07	0.511471E 05	0.0	0.0
3	0.208163E 07	0.511471E 05	0.0	0.0
4	0.208163E 07	0.511471E 05	0.0	0.0
5	0.208163E 07	0.511471E 05	0.0	0.0
6	0.208163E 07	0.511471E 05	0.0	0.0
7	0.208163E 07	0.511471E 05	0.0	0.0
8	0.208163E 07	0.511471E 05	0.0	0.0
9	0.208163E 07	0.511471E 05	0.0	0.0
10	0.208163E 07	0.511471E 05	0.0	0.0
11	0.208163E 07	0.511471E 05	0.0	0.0
12	0.208163E 07	0.511471E 05	0.0	0.0
13	0.208163E 07	0.511471E 05	0.0	0.0
14	0.208163E 07	0.511471E 05	0.0	0.0
15	0.208163E 07	0.511471E 05	0.0	0.0
16	0.208163E 07	0.511471E 05	0.0	0.0
17	0.208163E 07	0.511471E 05	0.0	0.0
18	0.208163E 07	0.511471E 05	0.0	0.0
19	0.208163E 07	0.511471E 05	0.0	0.0
20	0.208163E 07	0.511471E 05	0.0	0.0
21	0.208163E 07	0.511471E 05	0.0	0.0
22	0.208163E 07	0.511471E 05	0.0	0.0
23	0.208163E 07	0.511471E 05	0.0	0.0
24	0.208163E 07	0.511471E 05	0.0	0.0
25	0.208163E 07	0.511471E 05	0.0	0.0
26	0.208163E 07	0.511471E 05	0.0	0.0
27	0.208163E 07	0.511471E 05	0.0	0.0
28	0.208163E 07	0.511471E 05	0.0	0.0
29	0.208163E 07	0.511471E 05	0.0	0.0
30	0.208163E 07	0.511471E 05	0.0	0.0
31	0.208163E 07	0.511471E 05	0.0	0.0

Figure 8. Continued

THE INITIAL CONDITIONS FOR N= 0 FOLLOW

STATION	U	V	W	M S
1	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0
13	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0
15	0.0	0.0	0.0	0.0
16	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0
24	0.0	0.0	0.0	0.0
25	0.0	0.0	0.0	0.0
26	0.0	0.0	0.0	0.0
27	0.0	0.0	0.0	0.0
28	0.0	0.0	0.0	0.0
29	0.0	0.0	0.0	0.0
30	0.0	0.0	0.0	0.0
31	0.0	0.0	0.0	0.0

STATION	U DOT	V DOT	W DOT	M S DOT
1	0.0	0.0	0.0	0.0
2	0.0	0.0	-0.141355E 04	0.0
3	0.0	0.0	-0.141355E 04	0.0
4	0.0	0.0	-0.141355E 04	0.0
5	0.0	0.0	-0.141355E 04	0.0
6	0.0	0.0	-0.141355E 04	0.0
7	0.0	0.0	-0.141355E 04	0.0
8	0.0	0.0	-0.141355E 04	0.0
9	0.0	0.0	-0.141355E 04	0.0
10	0.0	0.0	-0.141355E 04	0.0
11	0.0	0.0	-0.141355E 04	0.0
12	0.0	0.0	-0.141355E 04	0.0
13	0.0	0.0	-0.141355E 04	0.0
14	0.0	0.0	-0.141355E 04	0.0
15	0.0	0.0	-0.141355E 04	0.0
16	0.0	0.0	-0.141355E 04	0.0
17	0.0	0.0	-0.141355E 04	0.0
18	0.0	0.0	-0.141355E 04	0.0
19	0.0	0.0	-0.141355E 04	0.0
20	0.0	0.0	-0.141355E 04	0.0
21	0.0	0.0	-0.141355E 04	0.0
22	0.0	0.0	-0.141355E 04	0.0
23	0.0	0.0	-0.141355E 04	0.0
24	0.0	0.0	-0.141355E 04	0.0
25	0.0	0.0	-0.141355E 04	0.0
26	0.0	0.0	-0.141355E 04	0.0
27	0.0	0.0	-0.141355E 04	0.0
28	0.0	0.0	-0.141355E 04	0.0
29	0.0	0.0	-0.141355E 04	0.0
30	0.0	0.0	-0.141355E 04	0.0
31	0.0	0.0	0.0	0.0

Figure 8. Continued

THE INITIAL CONDITIONS FOR N= 1 FOLLOW

STATION	U	V	W	M S
1	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0
13	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0
15	0.0	0.0	0.0	0.0
16	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0
24	0.0	0.0	0.0	0.0
25	0.0	0.0	0.0	0.0
26	0.0	0.0	0.0	0.0
27	0.0	0.0	0.0	0.0
28	0.0	0.0	0.0	0.0
29	0.0	0.0	0.0	0.0
30	0.0	0.0	0.0	0.0
31	0.0	0.0	0.0	0.0

STATION	U DOT	V DOT	W DOT	M S DOT
1	0.0	0.0	0.0	0.0
2	0.0	0.0	-0.222040E 04	0.0
3	0.0	0.0	-0.222040E 04	0.0
4	0.0	0.0	-0.222040E 04	0.0
5	0.0	0.0	-0.222040E 04	0.0
6	0.0	0.0	-0.222040E 04	0.0
7	0.0	0.0	-0.222040E 04	0.0
8	0.0	0.0	-0.222040E 04	0.0
9	0.0	0.0	-0.222040E 04	0.0
10	0.0	0.0	-0.222040E 04	0.0
11	0.0	0.0	-0.222040E 04	0.0
12	0.0	0.0	-0.222040E 04	0.0
13	0.0	0.0	-0.222040E 04	0.0
14	0.0	0.0	-0.222040E 04	0.0
15	0.0	0.0	-0.222040E 04	0.0
16	0.0	0.0	-0.222040E 04	0.0
17	0.0	0.0	-0.222040E 04	0.0
18	0.0	0.0	-0.222040E 04	0.0
19	0.0	0.0	-0.222040E 04	0.0
20	0.0	0.0	-0.222040E 04	0.0
21	0.0	0.0	-0.222040E 04	0.0
22	0.0	0.0	-0.222040E 04	0.0
23	0.0	0.0	-0.222040E 04	0.0
24	0.0	0.0	-0.222040E 04	0.0
25	0.0	0.0	-0.222040E 04	0.0
26	0.0	0.0	-0.222040E 04	0.0
27	0.0	0.0	-0.222040E 04	0.0
28	0.0	0.0	-0.222040E 04	0.0
29	0.0	0.0	-0.222040E 04	0.0
30	0.0	0.0	-0.222040E 04	0.0
31	0.0	0.0	0.0	0.0

Figure 8. Continued

THE INITIAL CONDITIONS FOR N= 2 FOLLOW

STATION	U	V	W	M S
1	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0
13	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0
15	0.0	0.0	0.0	0.0
16	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0
24	0.0	0.0	0.0	0.0
25	0.0	0.0	0.0	0.0
26	0.0	0.0	0.0	0.0
27	0.0	0.0	0.0	0.0
28	0.0	0.0	0.0	0.0
29	0.0	0.0	0.0	0.0
30	0.0	0.0	0.0	0.0
31	0.0	0.0	0.0	0.0

STATION	U DOT	V DOT	W DOT	M S DOT
1	0.0	0.0	0.0	0.0
2	0.0	0.0	-0.942367E 03	0.0
3	0.0	0.0	-0.942367E 03	0.0
4	0.0	0.0	-0.942367E 03	0.0
5	0.0	0.0	-0.942367E 03	0.0
6	0.0	0.0	-0.942367E 03	0.0
7	0.0	0.0	-0.942367E 03	0.0
8	0.0	0.0	-0.942367E 03	0.0
9	0.0	0.0	-0.942367E 03	0.0
10	0.0	0.0	-0.942367E 03	0.0
11	0.0	0.0	-0.942367E 03	0.0
12	0.0	0.0	-0.942367E 03	0.0
13	0.0	0.0	-0.942367E 03	0.0
14	0.0	0.0	-0.942367E 03	0.0
15	0.0	0.0	-0.942367E 03	0.0
16	0.0	0.0	-0.942367E 03	0.0
17	0.0	0.0	-0.942367E 03	0.0
18	0.0	0.0	-0.942367E 03	0.0
19	0.0	0.0	-0.942367E 03	0.0
20	0.0	0.0	-0.942367E 03	0.0
21	0.0	0.0	-0.942367E 03	0.0
22	0.0	0.0	-0.942367E 03	0.0
23	0.0	0.0	-0.942367E 03	0.0
24	0.0	0.0	-0.942367E 03	0.0
25	0.0	0.0	-0.942367E 03	0.0
26	0.0	0.0	-0.942367E 03	0.0
27	0.0	0.0	-0.942367E 03	0.0
28	0.0	0.0	-0.942367E 03	0.0
29	0.0	0.0	-0.942367E 03	0.0
30	0.0	0.0	-0.942367E 03	0.0
31	0.0	0.0	0.0	0.0

Figure 8. Continued

THE INITIAL CONDITIONS FOR N= 4 FOLLOW

STATION	U	V	W	M S
1	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0
6	0.0	0.0	0.0	0.0
7	0.0	0.0	0.0	0.0
8	0.0	0.0	0.0	0.0
9	0.0	0.0	0.0	0.0
10	0.0	0.0	0.0	0.0
11	0.0	0.0	0.0	0.0
12	0.0	0.0	0.0	0.0
13	0.0	0.0	0.0	0.0
14	0.0	0.0	0.0	0.0
15	0.0	0.0	0.0	0.0
16	0.0	0.0	0.0	0.0
17	0.0	0.0	0.0	0.0
18	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0
21	0.0	0.0	0.0	0.0
22	0.0	0.0	0.0	0.0
23	0.0	0.0	0.0	0.0
24	0.0	0.0	0.0	0.0
25	0.0	0.0	0.0	0.0
26	0.0	0.0	0.0	0.0
27	0.0	0.0	0.0	0.0
28	0.0	0.0	0.0	0.0
29	0.0	0.0	0.0	0.0
30	0.0	0.0	0.0	0.0
31	0.0	0.0	0.0	0.0

STATION	U DOT	V DOT	W DOT	M S DOT
1	0.0	0.0	0.0	0.0
2	0.0	0.0	0.188473E 03	0.0
3	0.0	0.0	0.188473E 03	0.0
4	0.0	0.0	0.188473E 03	0.0
5	0.0	0.0	0.188473E 03	0.0
6	0.0	0.0	0.188473E 03	0.0
7	0.0	0.0	0.188473E 03	0.0
8	0.0	0.0	0.188473E 03	0.0
9	0.0	0.0	0.188473E 03	0.0
10	0.0	0.0	0.188473E 03	0.0
11	0.0	0.0	0.188473E 03	0.0
12	0.0	0.0	0.188473E 03	0.0
13	0.0	0.0	0.188473E 03	0.0
14	0.0	0.0	0.188473E 03	0.0
15	0.0	0.0	0.188473E 03	0.0
16	0.0	0.0	0.188473E 03	0.0
17	0.0	0.0	0.188473E 03	0.0
18	0.0	0.0	0.188473E 03	0.0
19	0.0	0.0	0.188473E 03	0.0
20	0.0	0.0	0.188473E 03	0.0
21	0.0	0.0	0.188473E 03	0.0
22	0.0	0.0	0.188473E 03	0.0
23	0.0	0.0	0.188473E 03	0.0
24	0.0	0.0	0.188473E 03	0.0
25	0.0	0.0	0.188473E 03	0.0
26	0.0	0.0	0.188473E 03	0.0
27	0.0	0.0	0.188473E 03	0.0
28	0.0	0.0	0.188473E 03	0.0
29	0.0	0.0	0.188473E 03	0.0
30	0.0	0.0	0.188473E 03	0.0
31	0.0	0.0	0.0	0.0

Figure 8. Continued

THE TIME STEP NUMBER IS 250 THE TIME IS 4.56 OR 0.500E-03 SECONDS THE SOLUTION CONVERGED IN 2 ITERATIONS

THE SUMMED FORCES, MOMENTS, DISPLACEMENTS AND ROTATIONS FOLLOW FOR THETA = 0.0

STATION	N S	N THETA	N STHETA	Q S	M S	M THETA	M STHETA
1	-0.3495E 04	-0.9659E 03	0.0	-0.2371E 04	0.2368E 03	0.6771E 02	0.0
14	0.7610E 03	0.2495E 04	0.0	0.2499E 04	0.1838E 04	0.4449E 03	0.0
27	0.3077E 04	-0.3228E 04	0.0	-0.2860E 04	-0.6161E 03	-0.1717E 03	0.0
31	0.2633E 04	0.7879E 03	0.0	0.1825E 04	-0.9602E 02	-0.2746E 02	0.0

STATION	U	V	W	PHI S	PHI THETA	PHI
1	0.0	0.0	0.2250E-09	-0.2540E-09	0.0	0.0
14	-0.6019E-02	0.0	0.1312E 00	-0.3548E-01	0.0	0.0
27	-0.4080E-02	0.0	0.3925E-01	0.4481E-01	0.0	0.0
31	-0.2044E-09	0.0	-0.7940E-10	-0.5080E-09	0.0	0.0

THE SUMMED FORCES, MOMENTS, DISPLACEMENTS AND ROTATIONS FOLLOW FOR THETA = 0.314159E 01

STATION	N S	N THETA	N STHETA	Q S	M S	M THETA	M STHETA
1	0.4581E 04	0.1276E 04	-0.2199E-01	-0.1034E 04	0.1454E 04	0.4158E 03	-0.1008E-03
14	-0.6556E 04	-0.3293E 05	-0.8604E-02	-0.2622E 03	-0.8920E 03	-0.3827E 03	0.5865E-03
27	-0.3589E 04	-0.6110E 04	0.2565E-01	0.3636E 03	0.5828E 03	0.1070E 03	-0.4031E-03
31	-0.1640E 04	-0.5054E 03	0.1780E-01	0.5451E 03	0.1496E 04	0.4279E 03	0.6339E-04

STATION	U	V	W	PHI S	PHI THETA	PHI
1	0.0	0.0	-0.4845E-09	0.3302E-08	0.6000E-15	-0.1480E-07
14	0.5235E-02	-0.2114E-06	-0.2075E 00	0.2528E-01	0.9815E-07	-0.1485E-07
27	0.2690E-02	-0.6559E-07	-0.4600E-01	-0.3938E-01	0.2316E-07	0.2331E-07
31	0.8138E-10	-0.2562E-14	-0.4764E-10	0.2540E-08	-0.2530E-15	0.1198E-07

Figure 8. Continued

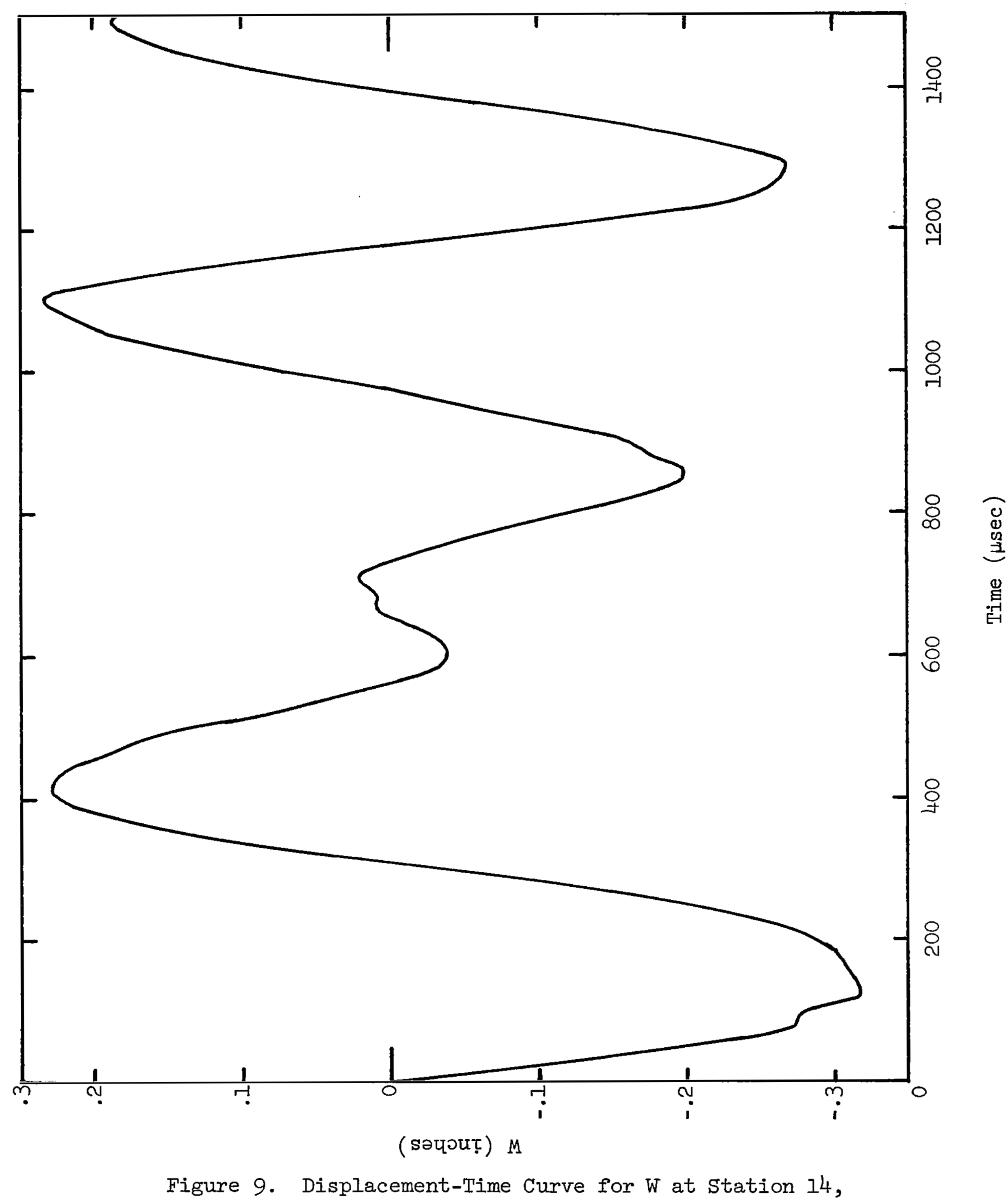

Figure 9. Displacement-Time Curve for W at Station 14,
$\theta = 0^{\circ}$, from Dynamic Analysis Example Problem

Subroutine Descriptions

MAIN

This program controls the logical connections between the subroutines. The case description, control parameters, physical constants, and boundary conditions are both read and printed out in this routine. The boundary conditions are nondimensionalized and many of the common indices and coefficients are determined here. The iteration procedure, the load incrementing procedure, and the calculation for the estimate of the next solution are all carried out here. The data for restarting the computation is written on tape and read from tape here.

Subroutine GEØM

This subroutine computes the nondimensional geometry functions of the shell.

Subroutine BDB (K,B,DB,D,DD)

This subroutine computes the nondimensional inplane and bending stiffnesses of the shell.

Subroutine PLØAD(K)

This subroutine computes the nondimensional Fourier coefficients of the loads applied to the shell.

Subroutine TLØAD(K)

This subroutine computes the nondimensional thermal loads.

Subroutine INITL

This subroutine computes the initial conditions on z and $\partial z/\partial t$.

Subroutine PMATRX

This subroutine calls subroutines HJ(K,MN), EFG(K,MN), ABC, and PANDD(K,MN) to set up the P, (P), $\overline{P}$, (DEE), and $\hat{P}$,(DST), matrices given by equations (30). Matrices DL, DG, and DF are set up for the calculation of x_1 given by equation (31a), where

$$x_1 = DL\ell_1 + DGg_1 + DFf_1$$

The special P matrix for a shell with an initial pole, given in Ref. [1], is also computed here. Matrices ZF1M, ZF2M, ZF3M, and ZF4M are set up for the calculation of Z_{K+1} given by equation (31b) where

$$Z_{K+1} = ZF1M\ell_K + ZF2MX_K + ZF3MX_{K-1} + ZF4Mf_K$$

If the shell has a final pole, the matrices CL0, CL1 and CL2 are prepared for the calculation of Z_K given by equation (D-3) in Ref. [1], where

$$Z_K = CL0X_{K-1} \text{ or } CL1X_{K-1} \text{ or } CL2X_K-1$$

depending upon whether n = 0 , 1 or 2.

Subroutine HJ(K,MN)

This subroutine computes the elements of the H and JAY matrices for both boundaries of the shell. The elements of H and JAY are defined in Ref. [1].

Subroutine EFG(K,MN)

This subroutine prepares the elements of the E, F, and G matrices for each meridian station K and for each Fourier mode MN. The matrices E, F, and G are given in Ref. [1].

Subroutine ABC

This subroutine computes the elements of the A, BEE, and C matrices defined by equation (25).

Subroutine PANDD(K,MN)

This subroutine computes the elements of the P, (P), $\overline{P}$, (DEE), and $\hat{P}$,(DST), matrices for each meridian station K and Fourier mode MN. These matrices are computed and saved because they do not change during either the iteration procedure or the load increment procedure, i.e., they are a function of the shell's initial geometry and stiffness.

Subroutine XANDZ

This subroutine computes the x vector using the P, $\bar{P}$, and $\hat{P}$ matrices and solves for the z vector for the applied and pseudo loads. The subourtines PHIBET(K) and TEAETA(K) are called and the previous solution for z, or the estimated value of z, is used to calculate the nonlinear Beta and Eta terms. The matrices FFS and FLS are the values of f at the initial and final edges of the shell. The subroutine FØRCE(K) is called to calculate the load vector g, (GEE), and the x vector at each meridian station. Once the x vector is obtained for all meridian stations the solution for z_{K+1} given by equation (31b) is obtained, and the solution for z_i at all the other meridian stations defined by equation (29b) is obtained. The solution z at the imaginary station off the initial edge of the shell is obtained last. The test for convergence of the solution is made as z is computed. The special conditions for computing z at either an initial or a final pole are also in this routine.

Subroutine INLPØL

This subroutine computes, for a shell with an initial pole, the nonlinear terms β_s, β_θ, $\beta_{s\theta}$, η_{ss}, and $\eta_{\theta s}$ at the pole. The appropriate equations are given in Ref. [1].

Subroutine FNLPØL

This subroutine computes, for a shell with a final pole, the nonlinear terms β_s, β_θ, $\beta_{s\theta}$, η_{ss}, and $\eta_{\theta s}$ at the pole. The appropriate equations are given in Ref. [1].

Subroutine MØDES

In MØDES, arrays that define those sets of indices that combine to equal each value of n in the problem are determined. MØDES is called prior to the first iteration and after every iteration until a specified number of Fourier terms is reached. Each Fourier index in the problem is subtracted from all other Fourier indices and the result is compared with all Fourier indices to see if the new value exists in the program. (The same comparison is never made twice.) If it does, the locations of the two indices that made the combination are stored in two special two-dimensional arrays, ID and JD. One argument of each array is the value of the new index and the other is the number of combinations of indices that also give this value of the index. If there is no index in the program that matches the new one, then a new Fourier term has been generated and will be considered in the next iteration for solution. The variable MAXD stores the total number of such combinations for each value of the Fourier index. In a similar manner, each index is added to every other index and the sum compared with all indices. This result is stored in the two two-dimensional arrays, IS and JS, in the same manner as was done for the subtraction case. The variable MAXS stores the total number of summation combinations for each value of the Fourier index. A special routine handles the cases where the index is added to and subtracted from itself. The two-dimensional array IJS stores the location of the index and the variable MAXSY stores the total number of such combinations. With this procedure the series of products that make up the β's and η's contain no zero terms, and the summation is carried out in PHIBET(K) and TEAETA(K) over specifically defined limits.

Subroutine ØUTPUT (IMØDE)

This subroutine prepares the printout material. Every IPRINT converged solution is printed. The Fourier coefficients of the inplane forces, meridional transverse force, circumferential bending moment, twisting moment and rotations can be computed and printed with the solution z for the Fourier coefficients of the three displacements and meridional bending moment. This output material is converted from dimensionless form to dimensional form here. Provision is made to print only at stations 1, IFREQ+1, 2IFREQ+1, etc. This subroutine also performs the summation process for computing the total values of the forces, moments, displacements, and rotations at the NTHMAX positions around circumference prescribed in the input data.

Subroutine PØLE(K)

This subroutine prints the solution at an initial and a final pole.

Subroutine PHIBET(K)

This subroutine calculates the Phis and carries out the multiplying and summation procedure for computing the Beta non-linear terms for a given meridional station K. The arrays IS, JS, ID, JD, IJS, MAXS, MAXD, AND MAXSY prepared in subroutine MØDES are used here.

Subroutine TEAETA(K)

This subroutine calculates the inplane forces and carries out the multiplying and summation procedure for computing the Eta nonlinear terms for a given meridional station K. The arrays IS, JS, ID, JD, IJS, MAXS, MAXD, AND MAXSY prepared in subroutine MØDES are used here.

Subroutine FØRCE(K)

This subroutine computes the $\bar{g}$, (GEE), vector, equation (28), and the x vector, equation (29a), for a given meridional station K. The vector GEES is the nonlinear value of $\bar{g}$ at station 1.

Subroutine UPDATE

This subroutine updates the storage locations of the Betas and Etas. It is called in subroutine XANDZ after a meridian station change.

Subroutine MATINV (A, N, B, M, DETERM, IPIVØT, INDEX, N MAX, ISCALE)

This subroutine solves the matrix equation AX = B where A is a square coefficient matrix and B is a matrix of constant vectors. A^{-1} is also obtained and the determinant of A is available. Jordan's method is used to reduce a matrix A to the identity matrix I through a succession of elementary transformations: $_n, _{n-1}, \ldots _1, A = I$. If these transformations are simultaneously applied to I and to a matrix B of constant vectors, the result is A^{-1} and X where AX = B. Each transformation is selected so that the largest element is used in the pivotal position. The subroutine has been compiled with a variable dimension statement A(N MAX, N MAX), B(N MAX, M). The following must be dimensioned in the calling program: IPIVOT(N MAX), INDEX(N MAX, 2), A(N MAX, N MAX), B(N MAX, M) where IPIVØT and INDEX are temporary storage blocks. An overflow may be caused by a singular matrix. The definition of the arguments of this subroutine are as follows:

A = first location of a 2-dimensional array of the A matrix.

N = location of order of A;

$$1 \leq N \leq N\ MAX$$

B = first location of a 2-dimensional array of the constant vectors B.

M = location of the number of column vectors in B.
M = 0 signals that the subroutine is to be used solely for inversion, however, in the call statement an entry corresponding to B must still be present.

DETERM - Gives the value of the determinant by the following formula:

$$DET(A) = (10^{18})^{ISCALE}(DETERM)$$

IPIVØT - temporary storage block.

INDEX - temporary storage block.

N MAX = location of maximum order of A as stated in dimension statement of calling program.

ISCALE - used in obtaining the value of the determinant by the following formula:

$$DET(A) = (10^{18})^{ISCALE}(DETERM)$$

At the return to the calling program A^{-1} is stored at A and X is stored at B.

APPENDIX A

CONVERSION OF U.S. CUSTOMARY UNITS TO SI UNITS

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

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

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

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

APPENDIX B

PROGRAM LISTING FOR DYNAMIC EXAMPLE

```
C     A CCMPUTER PROGRAM FOR THE GEOMETRICALLY NONLINEAR STATI
C        DYNAMIC ANALYSIS OF ARBITRARILLY LOADED SHELLS OF REV
CTHIS PROGRAM CONSISTS OF THE MAIN PROGRAM AND THE FOLLOWING
C     GECM--SPECIFIES THE SHELL GEOMETRY
C     BDB--SPECIFIES THE SHELL STIFFNESSES
C     PLOAD--SPECIFIES THE APPLIED PRESSURE LOADS
C     TLOAD-- SPECIFIES THE THERMAL LOADS
C     INITL--SPECIFIES THE INITIAL CONDITIONS
C     INLPOL
C     FNLPOL
C     MODES
C     XANDZ
C     ABC
C     PANDD
C     HJ
C     TEAETA
C     FORCE
C     UPDATE
C     MATINV
C     PMATRX
C     POLE
C     PHIBET
C     EFG
C     OUTPUT
        REAL NU,LAM,LAM2,JAY,MT,LSD18,LSD1N,MASS
        INTEGER SORD
        COMMON /IBL1/MNMAX/IBL2/N(10),MNINIT/IBL3/MO,M1,M2,M3/
       1/IBL5/IBCINL,IBCFNL/IBL6/KLL/IBL7/MNMAXO,MAXD(1C),MAXS
       20),IS(10,10),JS(10,10),ID(10,10),JD(10,10),IJS(10)/IBL
       3IBL9/MAXM/IBL1C/IFREQ,NTHMAX/IBL11/ICORFL,IPASS/IBL12/
       4NCONV/BL1/A(4,4),BEE(4,4),C(4,4)/BL3/PR(10),PX(10),PT(
       5/BL4/P(4,4,200),X(4,200),ZF1M(4,4,10),ZF2M(4,4,10),ZF3
       6ZF4M(4,4,10)
       7/BL5/TT(10), MT(10),DT(1C),DMT(10)
       9/BL7/D1,S1
       8/BL6/Z(4,220),SOE,OSE,ALOAD
       O/BL8/R(200),GAM(20C),OMT(200)
       1/BL9/FFS(4,1C),EL1S(4),GEES(4,10)
       2/BL10/PHIX(1C),PHIT(10),PHI(10)
       3/BL11/OMXI(2CO),PHEE,TO,T2
        COMMCN/BL12/TDLI,TD
       1EL/BL13/OMEG1(4,4),CAPL1(4,4),OMEGL(4,4),CAPLL(4,4),UN
       2/LAM2,LSD18,LSD1N/BL15/NU,U1(10),V1(10),W1(10),V2(10),
       3),U3(10),V3(10),W3(10)/BL16/EPS/BL17/DEL/BL18/EL1(4),E
       4TH(6)
       4/BL20/DEOMX(200)
       7/BL23/JAY(4,4),H(4,4)/BL24/DL
       5(4,4,10),DG(4,4,10),DF(4,4,10)/BL25/E(4,4),F(4,4),G(4,
       610),BT3(10),BXT3(10),BE3(10)/BL28/EXX3(10),ETT3(10),ET
       710),EX3(10),ET3(10)/BL29/BX1(10),BT1(10),BXT1(10),BE1(
       8BT2(10),BXT2(10),BE2(10)/BL3C/EXX1(10),ETT1(10),ETX1(1
       9ET1(10),EXX2(10),ETT2(10),ETX2(10),EXT2(10),EX2(10),ET
       ODELSQ,EXT1(10)
        COMMON
       1/BL34/DEE(4,4,200),DST(4,4,200)
       2/BL32/TKN,ELAST,CHAR,SIGO
        COMMON /IBL13/ ITRMAX,LSMAX
        COMMON /BL100/SORD,TEEO/BL101/ZO(4,220),Z2(4,220),Z3(4
       1/BL102/DELOAD/BL103/MASS(200)
       1 /BL104/ZDOT(4,220)
          DIMENSION SIGT(2),SIGC(2)
        DIMENSION TITLE(18)
  500   READ(5,862) TITLE
  502   READ(5,101) NO,SORD,IMODE,NDIMEN,NTHMAX,IFREQ,IPRINT,I
     1      ,KMAX,MNMAX,MAXM,LSMAX,LCHMAX,ITRMAX,IC
        READ(5,102) NU,SIGO,ELAST,TKN,CHAR,TEEO
        READ(5,102) DELOAD,EPS
        READ(5,101) ITAPE
        DELSD=DELOAD*DELOAD
        KL=KMAX-1
        KLL=KMAX-2
        KMAX1=KMAX+1
```

```
      KMAX2=KMAX+2
      AK=KL
       SIGT(1) = SIGO*TKN
       SIGT(2) = SIGO/ELAST
       SIGC(1) = SIGO*CHAR/ELAST
       SIGC(2) = SIGO*TKN**3/CHAR
      IF(NTHMAX.EQ.0) GO TO 10
      READ(5,105)(TH(NTH),NTH=1,NTHMAX)
   10 WRITE(6,201)NO
      WRITE(6,100) TITLE
      WRITE(6,203)
      IF(IBCINL.LT.0) GO TO 11
      READ(5,103)OMEG1,CAPL1,EL1
      WRITE(6,204)
      WRITE(6,205) (OMEG1(1,J),J=1,4),(CAPL1(1,J),J=1,4),EL1
      WRITE(6,206) (OMEG1(2,J),J=1,4),(CAPL1(2,J),J=1,4),EL1
      WRITE(6,207) (OMEG1(3,J),J=1,4),(CAPL1(3,J),J=1,4),EL1
      WRITE(6,208) (OMEG1(4,J),J=1,4),(CAPL1(4,J),J=1,4),EL1
       DO 98  I = 1,4
       DO 98  J = 1,4
       KKLM = J/4 + 1
       OMEG1(I,J) = OMEG1(I,J)*SIGT(KKLM)
   98  CAPL1(I,J) = CAPL1(I,J)*SIGC(KKLM)
   14 IF(IBCFNL.LT.0) GO TO 12
      READ(5,103)OMEGL,CAPLL,ELL
       WRITE (6,224)
      WRITE(6,205) (OMEGL(1,J),J=1,4),(CAPLL(1,J),J=1,4),ELL
      WRITE(6,206) (OMEGL(2,J),J=1,4),(CAPLL(2,J),J=1,4),ELL
      WRITE(6,207) (OMEGL(3,J),J=1,4),(CAPLL(3,J),J=1,4),ELL
      WRITE(6,208) (OMEGL(4,J),J=1,4),(CAPLL(4,J),J=1,4),ELL
       DO 99 I = 1,4
       DO 99 J = 1,4
       KKLM = J/4 + 1
       OMEGL(I,J) = OMEGL(I,J)*SIGT(KKLM)
   99  CAPLL(I,J) = CAPLL(I,J)*SIGC(KKLM)
      GO TO 13
   11 WRITE(6,210)
      GO TO 14
   12 WRITE(6,211)
   13 WRITE(6,307)
  307 FORMAT(////)
      IF(SORD.NE.C) GO TO 18
      WRITE(6,212)KMAX,MAXM,DELOAD,LSMAX,ITRMAX,LCHMAX,EPS
      WRITE(6,213)CHAR,TKN,ELAST,SIGO,NU
      GO TO 19
   18 WRITE(6,262)KMAX,MAXM,DELOAD,LSMAX,ITRMAX,EPS
      WRITE(6,263)CHAR,TKN,ELAST,SIGO,TEEO,NU
   19 IF(NTHMAX.EQ.0) GO TO 17
      WRITE(6,216)
      WRITE(6,217)(TH(NTH),NTH=1,NTHMAX)
   17 LAM=TKN/CHAR
      SOE=SIGO/ELAST
      OSE=.5*SOE
      O1=1.-NU
      S1=1.+NU
      LAM2=LAM**2
      IF(NDIMEN.LT.1) GO TO 228
      SIGO=1.
      ELAST=1.
      TKN=1.
      CHAR=1.
      WRITE(6,225)
  225 FORMAT(////////////43X,34HTHE DATA IS IN NONDIMENSIONA
     14H____________________________________)
      GO TO 229
  228 WRITE(6,226)
  226 FORMAT(////////////45X,31HTHE DATA IS IN DIMENSIONAL F
     1_________________________________)
  229 CONTINUE
      DO 230 M=1,MAXM
      N(M)=0.
      PX(M)=0.
```

```
      PT(M)=0.
      PR(M)=0.
      TT(M)=0.
      MT(M)=0.
      DT(M)=0.
      DMT(M)=0.
      MAXD(M)=0
      MAXS(M)=0
  230 MAXSY(M)=0
      CALL GEOM
      IF(SORD.NE.0) GO TO 804
      DO 86 K=1,KMAX
   86 MASS(K)=0.
      WRITE(6,802)
  802 FORMAT(1H1,17X,15H STATION        16H     RADIUS      16
     1       16H      OMEGA S     16H      OMEGA THETA16H      DEO
      DO 978 K=1,KMAX
      RKK=R(K)*CHAR
      OMXIK=OMXI(K)/CHAR
      GAMK=GAM(K)/CHAR
      OMTK=OMT(K)/CHAR
      DEOMXK=DEOMX(K)/(CHAR*CHAR)
  978 WRITE(6,803) K,RKK,GAMK,OMXIK,OMTK,DEOMXK
  803 FORMAT(20X,I3, 9X,5E16.4)
      GO TO 805
  804 WRITE(6,810)
  810 FORMAT(1H1, 5X,15H STATION        16H     RADIUS      16
     1       16H      OMEGA S     16H      OMEGA THETA16H      DEO
     2      MASS          //)
      TEED=TEEO
      IF(NDIMEN.EQ.1) TEED=1.
      DO 979 K=1,KMAX
      RKK=R(K)*CHAR
      OMXIK=OMXI(K)/CHAR
      GAMK=GAM(K)/CHAR
      OMTK=OMT(K)/CHAR
      DEOMXK=DEOMX(K)/(CHAR*CHAR)
      AMSS=MASS(K)*TEED**2*ELAST*TKN/CHAR**2
  979 WRITE(6,813) K,RKK,GAMK,OMXIK,OMTK,DEOMXK,AMSS
  813 FORMAT( 8X,I3, 9X,6E16.4)
  805 M0=0
      M1=0
      M2=0
      M3=0
      ABN=CHAR/SIGO/TKN
      ZN=SIGO*TKN
      WRITE(6,112)
  112 FORMAT(////17X,12H  STATION    20H        B STIFFNESS    2
     1IFFNESS    20H         B PRIME          20H        D PRIME
      DO 888 K=1,KMAX
      CALL BDB(K,B,DB,D,DD)
      BST=ELAST*TKN
      ZST=ELAST*TKN**3
      B=B*BST
      D = D*ZST
      DB=DB/CHAR*BST
      DD=DD/CHAR*ZST
  888 WRITE(6,71) K,B,D,DB,DD
   71 FORMAT(20X,I3,4X,4E20.6)
      CALL PLOAD(1)
      CALL TLOAD(1)
      IF(SORD.NE.0) GO TO 891
      DO 889 M=1,MNMAX
      WRITE(6,113) N(M)
  113 FORMAT(///25X,44HPRESSURE AND TEMPERATURE COEFFICIENTS
     1  FOLLOW//)
      WRITE(6,114)
  114 FORMAT(5X,7HSTATION,3X,15H         PR        15H          PX
     1   PT            15H        TT          15H        MT          15H
     25H        DMT          //)
      DO 890 K=1,KMAX
      CALL PLOAD(K)
```

```
      CALL TLOAD(K)
      PRM=PR(M)/ABN
      PTM=PT(M)/ABN
      PXM=PX(M)/ABN
      TTM=TT(M)*ZN
      EMTM=MT(M)/CHAR*ZN*TKN*TKN
      DTM=DT(M)/CHAR*ZN
      DMTM=DMT(M)*ZN*TKN*TKN/(CHAR*CHAR)
890   WRITE(6,115) K,PRM,PXM,PTM,TTM,EMTM,DTM,DMTM
115   FORMAT(6X,I3,7X,7E15.4)
889   CONTINUE
891   CONTINUE
      DELSQ=DEL**2
      TDLI=.5/DEL
      TDEL=2.*DEL
      MNINIT=1
      MNMAXO=MNMAX
      DO 20 I=1,4
      DO 20 J=1,4
      UNIT(I,J)=0.
      IF(I.EQ.J) UNIT(I,J)=1.
 20   CONTINUE
      NMAX=MAXM*KMAX2
      DO 22 K=1,NMAX
      DO 22 I=1,4
      ZDOT(I,K)=0.
      Z2(I,K)=0.
      Z3(I,K)=0.
      ZO(I,K)=0.
 22   Z(I,K)=0.
      ALOAD=DELOAD
      IF(IC.EQ.0) GO TO 834
      IF(ITAPE.GT.1) WRITE(6,280)
280   FORMAT(////5X,28HTHIS IS A RESTARTED SOLUTION//)
      IF(ITAPE.GT.1) GO TO 834
      CALL INITL
      ACO=CHAR*SIGO/ELAST
      ACM=SIGO*TKN**3/CHAR
      DO 830 M=1,MNMAX
      MM=(M-1)*KMAX2
      WRITE(6,126) N(M)
126   FORMAT(////5X,29HTHE INITIAL CONDITIONS FOR N=I3,8H   F
      WRITE(6,127)
127   FORMAT(19X,7HSTATION,3X,20H        U              20H
     1        20H       W                 20H        M S             //)
      DO 831 K=2,KMAX1
      MK=K+MM
      TU=ACO*ZO(1,MK)
      TV=ACO*ZO(2,MK)
      TW=ACO*ZO(3,MK)
      TM=ACM*ZO(4,MK)
      KK=K-1
      WRITE(6,71) KK,TU,TV,TW,TM
831   CONTINUE
      WRITE(6,129)
129   FORMAT(///19X,7HSTATION,3X,20H       U DOT          20H
     1        20H        W DOT            20H       M S DOT        /
      DO 832 K=2,KMAX1
      ACD=CHAR*SIGO/(ELAST*TEEO)
      AMD=SIGO*TKN**3/(CHAR*TEEO)
      MK=K+MM
      TU=ACD*ZDOT(1,MK)
      TV=ACD*ZDOT(2,MK)
      TW=ACD*ZDOT(3,MK)
      TM=AMD*ZDOT(4,MK)
      KK=K-1
      WRITE(6,71) KK,TU,TV,TW,TM
      DO 833 I=1,4
      Z(I,MK)=ZO(I,MK)+ZDOT(I,MK)*DELOAD
      Z2(I,MK)=ZO(I,MK)-ZDOT(I,MK)*DELOAD
833   Z3(I,MK)=ZO(I,MK)-2.*ZDOT(I,MK)*DELOAD
832   CONTINUE
```

```
  830 CONTINUE
  834 CONTINUE
      IF(SORD.NE.0) ALOAD=1.
C     READ RESTART DATA ON TAPE UNIT 8
      IF(ITAPE.NE.0) REWIND 8
      IF(ITAPE.EQ.2.OR.ITAPE.EQ.3) READ(8) ((Z(I,JP),ZO(I,JP
     1 ,Z3(I,JP),I=1,4),JP=1,NMAX)
      CALL PMATRX
      LSTEP=1
      LCHANG=0
      ITR=1
      ICORFL=0
      IF(MNMAX.EQ.MAXM) ICORFL=1
      IPASS=0
  400 CALL XANDZ
      IF(ITRMAX.EQ.1) GO TO 50
      MNMAXO=MNMAX
      IF(IPASS.LT.2) CALL MODES
      IF(NCONV.EQ.1) GO TO 50
      IF(ITR.LT.ITRMAX) GO TO 23
      IF(SORD.NE.0) GO TO 365
      IF(LCHANG.LT.LCHMAX) GO TO 30
      WRITE(6,220) NO
      GO TO 500
   50 FL=LSTEP
      FI=IPRINT
      LI=LSTEP/IPRINT
      FLI=LI
      FT=FLI-FL/FI
      IF(FT.EQ.0.) CALL OUTPUT(IMODE)
      IF(SORD.EQ.0) GO TO 60
      DO 65 MN=1,MNMAXO
      DO 65 K=1,KMAX2
      IK=K+(MN-1)*KMAX2
      DO 65 I=1,4
      ZN=3.*(Z(I,IK)-ZO(I,IK))+Z2(I,IK)
      Z3(I,IK)=Z2(I,IK)
      Z2(I,IK)=ZO(I,IK)
      ZO(I,IK)=Z(I,IK)
   65 Z(I,IK)=ZN
      IF(LSTEP.GE.LSMAX) GO TO 360
      GO TO 62
   60 DO 61 MN=1,MNMAXO
      DO 61 K=1,KMAX2
      IK=K+(MN-1)*KMAX2
      DO 61 I=1,4
      ZN=2.*Z(I,IK)-ZO(I,IK)
      ZO(I,IK)=Z(I,IK)
   61 Z(I,IK)=ZN
      IF(LSTEP.GE.LSMAX) GO TO 360
   62 ALOAD=ALOAD+DELOAD
      IF(SORD.NE.0) ALOAD=1.
      LSTEP=LSTEP+1
      ITR=1
      GO TO 400
  360 IF(SORD.NE.0) GO TO 361
      WRITE(6,221) NO
      GO TO 500
   23 ITR=ITR+1
      GO TO 400
   30 IF(LSTEP-1) 310,310,320
  310 WRITE(6,223)
      GO TO 500
  320 WRITE(6,222)
      LCHANG=LCHANG+1
      LSTEP=LSTEP-1
      ALOAD=ALOAD-DELOAD
      DELOAD=DELOAD/5.
  325 DO 31 MN=1,MNMAXO
      DO 31 K=1,KMAX2
      IK=K+(MN-1)*KMAX2
      DO 31 I=1,4
```

```
  31 Z(I,IK)=ZO(I,IK)
     GO TO 62
 361 WRITE(6,271)
C    WRITE RESTART DATA ON TAPE UNIT 8
     IF(ITAPE.NE.0) REWIND 8
      IF(ITAPE.EQ.1.OR.ITAPE.EQ.3) WRITE(8) ((Z(I,JP),ZO(I,
    1    ,Z3(I,JP),I=1,4),JP=1,NMAX)
     GO TO 500
 365 WRITE(6,266) ITRMAX,LSTEP,NO
     GO TO 500
 862 FORMAT(18A4)
 100 FORMAT(38X,18A4,///)
 101 FORMAT(16I5)
 102 FORMAT(6E12.3)
 103 FORMAT(4E16.8)
 105 FORMAT(6E12.3)
 201 FORMAT(1H1,48X,16H--PROBLEM NUMBER,I4,2H--,///)
 202 FORMAT(1X12A6///)
 203 FORMAT(49X,21H--INPUT DATA RECORD--,//34H      THE BOUN
    1ONS ARE: //)
 204 FORMAT(29H            AT THE INITIAL EDGE//2X,40H-------
    1A BAR----------------15X,40H---------------LAMBDA BAR-
    2--12X,10H---EL-----/)
 205 FORMAT(1X,1H(,4E10.3,15H)  N S        (,4E10.3,12H)  U
    1    E10.3)
 206 FORMAT(1X,1H(,4E10.3,15H)  N ST   +   (,4E10.3,12H)  V
    1    E10.3)
 207 FORMAT(1X,1H(,4E10.3,15H)  Q S        (,4E10.3,12H)  W
    1    E10.3)
 208 FORMAT(1X,1H(,4E10.3,15H)  PHI S      (,4E10.3,12H)  M
    1    E10.3)
 210 FORMAT(39H            THE SHELL HAS AN INITIAL POLE/)
 211 FORMAT(1H0,36H             THE SHELL HAS A FINAL POLE//)
 212 FORMAT(5X,40HNUMBER OF STATIONS----------------------I
    1ER OF MODES-----------------------I3/5X,40HINCREMENT
    2OR----------------F6.3/5X,40HMAXIMUM NUMBER OF LOAD S
    3----I3/5X,40HMAXIMUM NUMBER OF ITERATIONS------------I
    4MUM NUMBER OF LOAD FACTOR CHANGES---I3/5X,40HCONVERGEN
    5------------------F6.3//)
 213 FORMAT(5X,33HCHARACTERISTIC SHELL DIMENSION---E12.4/ 5
    1CE THICKNESS--------------E12.4/5X,33HREFERENCE ELASTI
    2-----E12.4/5X,33HREFERENCE STRESS-----------------E12.
    3SON'S RATIO-----------------E12.4//)
 216 FORMAT( 5X,62HCIRCUMFERENTIAL COORDINATES FOR PRINT RE
    1IANS, ARE:/)
 217 FORMAT(10X,E16.6,5(1H,,E16.6,1H ))
 220 FORMAT(1H1,80H             THE MAXIMUM NUMBER OF LOAD CHA
    1EN MADE.  END PROBLEM NUMBERI4)
 221 FORMAT(1H1,79H             THE MAXIMUM NUMBER OF LOAD STE
    1 TAKEN.  END PROBLEM NUMBERI4)
 222 FORMAT(1H1,119H            THE SOLUTION DID NOT CONVERGE
    1AXIMUM NUMBER OF ITERATIONS.  THE LOAD FACTOR HAS BEEN
    25.)
 223 FORMAT(1H1,69H             THE SOLUTION DID NOT CONVERGE
    1T LOAD INCREMENT./11X,71HLOOK FOR AN ERROR IN THE INPU
    2RY A SMALLER VALUE FOR DELOAD.)
 224 FORMAT(1H0,26H            AT THE FINAL EDGE//2X,40H------
    1GA BAR----------------15X,40H---------------LAMBDA BAR
    2---12X,10H---EL-----/)
 262 FORMAT(5X,40HNUMBER OF STATIONS----------------------I
    1ER OF MODES-----------------------I3/5X,40HINCREMENT
    2------------------E9.3/5X,40HMAXIMUM NUMBER OF TIME S
    3----I5/5X,40HMAXIMUM NUMBER OF ITERATIONS------------I
    4ERGENCE CRITERION------------------F6.3//)
 263 FORMAT(5X,33HCHARACTERISTIC SHELL DIMENSION---E12.4/ 5
    1CE THICKNESS--------------E12.4/5X,33HREFERENCE ELASTI
    2-----E12.4/5X,33HREFERENCE STRESS-----------------E12.
    3RENCE TIME------------------E12.4/5X,33HPOISSON'S RAT
    4--------E12.4//)
 266 FORMAT(1H1,35H    THE SOLUTION DID NOT CONVERGE INI3,24
    1 AT TIME STEPI5,21H.  END PROBLEM NUMBERI4,1H.)
 271 FORMAT(1H1,79H             THE MAXIMUM NUMBER OF TIME STE
```

```
     1 TAKEN.   END PROBLEM NUMBERI4)
      END

      SUBROUTINE GEOM
      REAL MASS
      COMMON
     1/IBL4/KMAX,KL
     0/BL8/R(200),GAM(200),OMT(200)
     3/BL11/OMXI(200),PHEE,TO,T2
     4/BL17/DEL
     4/BL20/DEOMX(200)
     6/BL32/TKN,ELAST,CHAR,SIGO
     1/BL102/DELOAD/BL103/MASS(200)
C    GEOMETRY DATA
      AKX=KMAX-1
      DEL=1./AKX
      THET=ARSIN(2.2801/CHAR)
      DO 11 K=1,KMAX
      AK=K
      R(K)=(7.9499+(AK-1.)*(2.2801)/AKX)/CHAR
      GAM(K)=(2.2801/CHAR)/R(K)
      OMXI(K)=0.
      DEOMX(K)=0.
      OMT(K)=COS(THET)/R(K)
      MASS(K)=1.
   11 CONTINUE
      RETURN
      END

      SUBROUTINE BDB(K,B,DB,D,DD)
      REAL NU
      COMMON
     1/BL32/TKN,ELAST,CHAR,SIGO/BL15
     2/NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(10),V
     3/BL17/DEL
C   STIFFNESS DATA
      B=1.089082
      D=.09075683
      DB=0.
      DD=0.
      RETURN
      END

      SUBROUTINE PLOAD(K)
      COMMON
     1/BL32/TKN,ELAST,CHAR,SIGO/IBL2/NN(10),MNINIT
     8/BL6/Z(4,220),SOE,OSE,ALOAD
     4/IBL1/MNMAX
     5/BL3/PR(10),PX(10),PT(10)
      COMMON/IBL4/KMAX,KL
      COMMON/BL8/R(200),GAM(200),OMT(200)
     1/IBL8/LSTEP,ITR
     1/BL102/DELOAD/BL103/MASS(200)
      RETURN
      END

      SUBROUTINE TLOAD(K)
      REAL NU
      COMMON
     1/IBL1/MNMAX/IBL2/NN(10),MNINIT
     2/BL5/TT(10),EMT(10),DT(10),DMT(10)
     3/BL32/TKN,ELAST,CHAR,SIGO
     4/BL6/Z(4,220),SOE,OSE,ALOAD/BL15/
     5NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(10),V3
     1/IBL8/LSTEP,ITR
      RETURN
      END
```

```
      SUBROUTINE INITL
      COMMON /BL101/Z0(4,220),Z2(4,220),Z3(4,220),DELSD
     1  /BL104/ZDOT(4,220)/BL6/Z(4,220),SOE,OSE,ALOAD
     1/IBL2/NN(10),MNINIT
     2 /IBL1/MNMAX/IBL9/MAXM/IBL12/KMAX1,KMAX2,NCONV/IBL4/KM
     3/BL32/TKN,ELAST,CHAR,SIGO/BL100/SORD,TEEO
C    INITIAL CONDITIONS DATA
      NN(1)=0
      NN(2)=1
      NN(3)=2
      NN(4)=4
      PI=3.14159
      DO 2 M=1,MAXM
      IF(M.EQ.1) VEL=-444.08/PI
      IF(M.EQ.2) VEL=-444.08/2.
      IF(M.EQ.3) VEL=-444.08*2./(3.*PI)
      IF(M.EQ.4) VEL=444.08*2./(15.*PI)
      DO 2 K=2,KL
      I=K+1+(M-1)*KMAX2
    2 ZDOT(3,I)=VEL*ELAST*TEEO/(CHAR*SIGO)*10.
      RETURN
      END


      SUBROUTINE INLPOL
      COMMON
     1/IBL1/MNMAX
     1/IBL3/MO,M1,M2,M3
     2/IBL12/KMAX1,KMAX2,NCONV
     8/BL6/Z(4,220),SOE,OSE,ALOAD
     4/BL7/D1,S1
     3/BL11/OMXI(200),PHEE,T0,T2
     6/BL17/DEL
     7/BL29/BX1(10),BT1(10),BXT1(10),BE1(10),BX2(10),BT2(10)
     8BE2(10)
     9/BL30/EXX1(10),ETT1(10),ETX1(10),EX1(10),ET1(10),EXX2(
     0,ETX2(10),EXT2(10),EX2(10),ET2(10)/BL31/DELSQ,EXT1(10)
     1/BL5/TT(10),EMT(10),DT(10),DMT(10)
     2/IBL13/ITRMAX,LSMAX
      DO 1 MN=1,MNMAX
      BX1 (MN)=0.
      BT1 (MN)=0.
      BXT1(MN)=0.
      BE1 (MN)=0.
      EX1 (MN)=0.
      ET1 (MN)=0.
      ETX1(MN)=0.
    1 EXX1(MN)=0.
      IF(M1.EQ.0) RETURN
      I2=2+(M1-1)*KMAX2
      I3=I2+1
      I4=I3+1
      PHEE=(1.5*Z(3,I2)-2.*Z(3,I3)+.5*Z(3,I4))/DEL+OMXI(1)*Z
      BET=.5*PHEE**2
      IF(ITRMAX.EQ.1) BET=0.
      T2=0.
      IF(M2.EQ.0) GO TO 2
      CALL BDB(1,B,DB,D,DD)
      I2=2+(M2-1)*KMAX2
      I3=I2+1
      I4=I3+1
      T2=B*D1*((-1.5*Z(1,I2)+2.*Z(1,I3)-.5*Z(1,I4))/DEL+.5*S
      Q1=.5*PHEE*T2
      BX1(M2)=BET
      BT1(M2)=-BET
      BXT1(M2)=-BET
      ETX1(M1)=Q1
      IF(M3.EQ.0) GO TO 2
```

```
      EXX1(M3)=Q1
      ETX1(M3)=-Q1
    2 TO=0.
      IF(MO.EQ.0) GO TO 3
      BX1(MO)=BET
      BT1(MO)=BET
      CALL BDB(1,B,DB,D,DD)
      CALL TLOAD(1)
      I2=2+(MO-1)*KMAX2
      I3=I2+1
      I4=I3+1
      TO=B*S1*((-1.5*Z(1,I2)+2.*Z(1,I3)-.5*Z(1,I4))/DEL+OMXI
     1+.5*SOE*BET)-TT(MO)*ALOAD
    3 EXX1(M1)=PHEE*(TO+.5*T2)
      RETURN
      END

      SUBROUTINE FNLPOL
      COMMON
     1/IBL1/MNMAX
     2/IBL3/MO,M1,M2,M3
     3/IBL4/KMAX,KL
     4/IBL12/KMAX1,KMAX2,NCONV
     8/BL6/Z(4,220),SOE,OSE,ALOAD
     6/BL7/D1,S1
     3/BL11/OMXI(200),PHEE,TO,T2
     8/BL17/DEL
     9/BL27/BX3(10),BT3(10),BXT3(10),BE3(10)
     0/BL28/EXX3(10),ETT3(10),ETX3(10),EXT3(10),EX3(10),ET3(
     1/BL5/TT(10),EMT(10),DT(10),DMT(10)
     2/IBL13/ITRMAX,LSMAX
      DO 1 MN=1,MNMAX
      BX3 (MN)=0.
      BT3 (MN)=0.
      BXT3(MN)=0.
      BE3 (MN)=0.
      EX3 (MN)=0.
      ET3 (MN)=0.
      ETX3(MN)=0.
    1 EXX3(MN)=0.
      CALL BDB(KMAX,B,DB,D,DB)
      IF(M1.EQ.0) RETURN
      KM=KMAX1+(M1-1)*KMAX2
      KM1=KM-1
      KM2=KM-2
      PHEE=-(1.5*Z(3,KM)-2.*Z(3,KM1)+.5*Z(3,KM2))/DEL+OMXI(K
      BET=.5*PHEE**2
      IF(ITRMAX.EQ.1) BET=0.
      T2=0.
      IF(M2.EQ.0) GO TO 2
      KM=KMAX1+(M2-1)*KMAX2
      KM1=KM-1
      KM2=KM-2
      T2=B*D1*((1.5*Z(1,KM)-2.*Z(1,KM1)+.5*Z(1,KM2))/DEL+.5*
      Q1=.5*PHEE*T2
      BX3(M2)=BET
      BT3(M2)=-BET
      BXT3(M2)= BET
      ETX3(M1)=Q1
      IF(M3.EQ.0) GO TO 2
      EXX3(M3)=Q1
      ETX3(M3)=-Q1
    2 TO=0.
      IF(MO.EQ.0) GO TO 3
      CALL TLOAD(KMAX)
      KM=KMAX1+(MO-1)*KMAX2
      KM1=KM-1
      KM2=KM-2
      BX3(MO)=BET
      BT3(MO)=BET
      TO=B*S1*((1.5*Z(1,KM)-2.*Z(1,KM1)+.5*Z(1,KM2))/DEL+OMX
```

```
   1Z(3,KM)+.5*SOE*BET)-TT(MO)*ALOAD
 3  EXX3(M1)=PHEE*(T0+.5*T2)
    RETURN
    END

    SUBROUTINE MODES
    COMMON
   1/IBL1/MNMAX
   2/IBL2/N(10),MNINIT
   3/IBL7/MNMAXO,MAXD(10),MAXS(10),MAXSY(10),IS(10,10),JS(
   4,10),JD(10,10),IJS(10)
   5/IBL9/MAXM
   6/IBL11/ICORFL,IPASS
    IF(MAXM.EQ.1) RETURN
    IF(MNINIT.GT.MAXM) RETURN
    DO 1 MN=1,MNMAXO
    NMN=N(MN)
    NNS=MN
    IF(MNINIT.GT.MN) NNS=MNINIT
    DO 1 MM=NNS,MNMAXO
    NMM=N(MM)
    NTEST=IABS(NMN-NMM)
    DO 2 MMFT=1,MNMAX
    IF(NTEST.EQ.N(MMFT)) GO TO 10
 2  CONTINUE
    IF(ICORFL.EQ.1) GO TO 1
    MNMAX=MNMAX+1
    N(MNMAX)=NTEST
    MMFT=MNMAX
    IF(MNMAX.EQ.MAXM) ICORFL=1
10  IF(NMN-NMM) 11,1,12
11  LOCD=MAXD(MMFT)+1
    MAXD(MMFT)=LOCD
    ID(LOCD,MMFT)=MM
    JD(LOCD,MMFT)=MN
    GO TO 1
12  LOCD=MAXD(MMFT)+1
    MAXD(MMFT)=LOCD
    ID(LOCD,MMFT)=MN
    JD(LOCD,MMFT)=MM
 1  CONTINUE
    DO 301 MN=1,MNMAXO
    NMN=N(MN)
    NNS=MN
    IF(MNINIT.GT.MN) NNS=MNINIT
    DO 301 MM=NNS,MNMAXO
    NMM=N(MM)
    NTEST=NMN+NMM
    DO 302 MMFT=1,MNMAX
    IF(NTEST.EQ.N(MMFT)) GO TO 310
302 CONTINUE
    IF(ICORFL.EQ.1) GO TO 301
    IF(MNMAX.GE.MAXM) GO TO 301
    MNMAX=MNMAX+1
    N(MNMAX)=NTEST
    MMFT=MNMAX
    IF(MNMAX.GE.MAXM) ICORFL=1
310 IF(NMN.EQ.NMM) GO TO 360
    LOCS=MAXS(MMFT)+1
    MAXS(MMFT)=LOCS
    IS(LOCS,MMFT)=MN
    JS(LOCS,MMFT)=MM
    GO TO 301
360 IF(NMN.EQ.0) GO TO 301
    MAXSY(MMFT)=1
    IJS(MMFT)=MN
301 CONTINUE
    MNINIT=MNMAXO+1
    IF(ICORFL.GT.0) IPASS=IPASS+1
    IF(IPASS.LT.2.AND.MNINIT.LE.MNMAX) CALL PMATRX
    RETURN
```

```
      END

      SUBROUTINE XANDZ
      REAL NU,LAM2,MT,MASS
      COMMON/BL5/TT(10),MT(10),DT(10),DMT(10)
      COMMON
     1/IBL1/MNMAX
     2/IBL3/MO,M1,M2,M3
     3/IBL4/KMAX,KL
     4/IBL5/IBCINL,IBCFNL
     5/IBL6/KLL
     3/IBL7/MNMAXO,MAXD(10),MAXS(10),MAXSY(10),IS(10,10),JS(
     4,10),JD(10,10),IJS(10)
      COMMON/IBL8/LSTEP,ITR
     8/IBL12/KMAX1,KMAX2,NCONV
     1/IBL13/ITRMAX,LSMAX
     9/BL1/A(4,4),BEE(4,4),C(4,4)
     2/BL3/PR(10),PX(10),PT(10)
     5/BL4/P(4,4,200),X(4,200),ZF1M(4,4,10),ZF2M(4,4,10),ZF3
     1ZF4M(4,4,10)
     8/BL6/Z(4,220),SOE,OSE,ALOAD
     3/BL7/D1,S1
     0/BL8/R(200),GAM(200),OMT(200)
     5/BL9/FFS(4,10),EL1S(4),GEES(4,10)
     6/BL14/LAM2,LSD18,LSD1N
     7/BL15/NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(
     8W3(10)
      COMMON
     1/BL16/EPS
     2/BL27/BX3(10),BT3(10),BXT3(10),BE3(10)
     3/BL28/EXX3(10),ETT3(10),ETX3(10),EXT3(10),EX3(10),ET3(
     4/BL29/BX1(10),BT1(10),BXT1(10),BE1(10),BX2(10),BT2(10)
     5BE2(10)
     6/BL30/EXX1(10),ETT1(10),ETX1(10),EX1(10),ET1(10),EXX2(
     7,ETX2(10),EXT2(10),EX2(10),ET2(10)
     8/BL31/DELSQ,EXT1(10)
     9/BL18/EL1(4),ELL(4)
      COMMON /BL100/SORD,TEEO/BL101/ZO(4,220),Z2(4,220),Z3(4
     1 /BL104/ZDOT(4,220)/BL102/DELOAD
     1/BL103/MASS(200)
      DIMENSION ELLS(4),FLS(4),ZT(4),IPIVOT(4),INDEX(4,2)
     1,CLO(4,4),CL1(4,4),CL2(4,4)
     2,TZMAX(4,10),ZDD(4)
      EQUIVALENCE (CLC(1),ZF1M(1)),(CL1(1),ZF2M(1)),(CL2(1),
      DO 201 I=1,4
      DO 201 M=1,MNMAX
      MJ=1+(M-1)*KMAX2
      TZMAX(I,M)=ABS(Z(I,MJ))
      DO 201 K=2,KMAX2
      KM=K+(M-1)*KMAX2
      AZTST=ABS(Z(I,KM))
      IF(AZTST.GT.TZMAX(I,M))  TZMAX(I,M)=AZTST
  201 CONTINUE
      NCONV=1
      IF(ITRMAX.EQ.1) GO TO 66
      DO 1 M=1,MNMAXO
      I=1+(KMAX+2)*(M-1)
      U1(M)=Z(1,I)
      V1(M)=Z(2,I)
      W1(M)=Z(3,I)
      I1=I+1
      U2(M)=Z(1,I1)
      V2(M)=Z(2,I1)
    1 W2(M)=Z(3,I1)
      IF(IBCINL.LT.0) GO TO 100
      CALL PHIBET(1)
      DO 2 M=1,MNMAX
      BX1(M)=BX3(M)
      BT1(M)=BT3(M)
      BXT1(M)=BXT3(M)
    2 BE1(M)=BE3(M)
```

```
      CALL TEAETA(1)
      DO 3 M=1,MNMAX
      EXX1(M)=EXX3(M)
      ETT1(M)=ETT3(M)
      ETX1(M)=ETX3(M)
      EXT1(M)=EXT3(M)
      EX1(M)=EX3(M)
    3 ET1(M)=ET3(M)
  102 CALL PHIBET(2)
      DO 4 M=1,MNMAX
      BX2(M)=BX3(M)
      BT2(M)=BT3(M)
      BXT2(M)=BXT3(M)
    4 BE2(M)=BE3(M)
      CALL TEAETA(2)
      DO 5 M=1,MNMAX
      EXX2(M)=EXX3(M)
      ETT2(M)=ETT3(M)
      ETX2(M)=ETX3(M)
      EXT2(M)=EXT3(M)
      EX2(M)=EX3(M)
    5 ET2(M)=ET3(M)
      CALL PHIBET(3)
      CALL TEAETA(3)
   66 CONTINUE
      IF(IBCINL.LT.0) GO TO 20
      CALL BDB(1,B1,DB,D,DD)
      GAM1=GAM(1)
      CALL TLOAD(1)
      DO 8 M=1,MNMAX
      IF(ITRMAX.EQ.1) GO TO 67
      FFS(1,M)=-TT(M)*ALOAD+OSE*(BX1(M)+BE1(M)+NU*(BT1(M)+BE
      FFS(2,M)=OSE*(B1   *D1      *BXT1(M)+EX1(M)+ET1(M))
      FFS(3,M)=LAM2*GAM1*D1*MT(M)*ALOAD-(EXX1(M)+ETX1(M))*SO
      GO TO 8
   67 FFS(1,M)=-TT(M)*ALOAD
      FFS(2,M)=0.
      FFS(3,M)=LAM2*GAM1*D1*MT(M)*ALOAD
    8 FFS(4,M)=0.
      DO 9 I=1,4
    9 EL1S(I)=ALOAD*EL1(I)
      CALL FORCE(1)
   20 CALL FORCE(2)
      DO 10 K=3,KLL
      KP=K+1
      IF(ITRMAX.EQ.1) GO TO 10
      CALL UPDATE
      CALL PHIBET(KP)
      CALL TEAETA(KP)
   10 CALL FORCE(K)
      IF(ITRMAX.NE.1) CALL UPDATE
      IF(IBCFNL.LT.0) GO TO 120
      IF(ITRMAX.EQ.1) GO TO 11
      CALL PHIBET(KMAX)
      CALL TEAETA(KMAX)
   11 CALL FORCE(KL)
      CALL FORCE(KMAX)
      DO 12 I=1,4
   12 ELLS(I)=ALOAD*ELL(I)
      CALL BDB(KMAX,BL,DB,D,DD)
      GAML=GAM(KMAX)
      FLS(4)=0.
      CALL TLOAD(KMAX)
      DO 14 M=1,MNMAX
      IF(ITRMAX.EQ.1) GO TO 68
      FLS(1)=-TT(M)*ALOAD+OSE*(BX3(M)+BE3(M)+NU*(BT3(M)+BE3(
      FLS(2)   =OSE*(BL*D1*BXT3(M)+EX3(M)+ET3(M))
      FLS(3)=LAM2*GAML*D1*MT(M)*ALOAD-(EXX3(M)+ETX3(M))*SOE
      GO TO 69
   68 FLS(1)=-TT(M)*ALOAD
      FLS(2)=0.
      FLS(3)=LAM2*GAML*D1*MT(M)*ALOAD
```

```
   69 CONTINUE
      IK=KL+KMAX*(M-1)
      IJ=KMAX*M
      L=M*KMAX2
      DO 14 I=1,4
      SUMZ=0.
      DO 15 J=1,4
   15 SUMZ=SUMZ+ZF1M(I,J,M)*ELLS(J)+ZF2M(I,J,M)*X(J,IJ)+ZF3M
     1X(J,IK)+ZF4M(I,J,M)*FLS(J)
   14 Z(I,L)=SUMZ
      LS=1
  150 DO 16 M=1,MNMAX
      DO 16 L=LS,KMAX
      K=KMAX2-L
      KPX=K-1
      KZ=K+1
      IJ=KPX+(M-1)*KMAX
      JK=KZ+(M-1)*KMAX2
      KK=JK-1
      DO 17 I=1,4
      SUMZ=0.
      DO 18 J=1,4
   18 SUMZ=SUMZ-P(I,J,IJ)*Z(J,JK)
      SUMZ=SUMZ+X(I,IJ)
      ASUMZ=ABS(SUMZ)
      IF(ASUMZ.GT.1.E+15) ITR=ITRMAX
      IF(NCONV.NE.1 .OR. ASUMZ .LT. 1.E-05) GO TO 17
      DELZ=ABS(Z(I,KK)-SUMZ)
      ZTEST=EPS*TZMAX(I,M)
      IF(DELZ.GT.ZTEST) NCONV=0
   17 Z(I,KK)=SUMZ
   16 CONTINUE
      IF(IBCINL.LT.0) GO TO 30
      DO 25 M=1,MNMAX
      CALL EFG(1,M)
      CALL ABC
      IJ=2+(M-1)*KMAX2
      IJ1=IJ+1
      IJ2=IJ-1
      DO 21 I=1,4
      SUMZ=0.
      DO 22 J=1,4
   22 SUMZ=SUMZ-A(I,J)*Z(J,IJ1)-BEE(I,J)*Z(J,IJ)
   21 ZT(I)=SUMZ+GEES(I,M)
      CALL MATINV(C,4,ZT,1,DETERM,IPIVOT,INDEX,4,ISCALE)
      DO 23 I=1,4
   23 Z(I,IJ2)=ZT(I)
   25 CONTINUE
   30 RETURN
  100 CALL INLPOL
      DO 101 M=1,MNMAXO
      U1(M)=U2(M)
      V1(M)=V2(M)
      W1(M)=W2(M)
      IJ=3+KMAX2*(M-1)
      U2(M)=Z(1,IJ)
      V2(M)=Z(2,IJ)
  101 W2(M)=Z(3,IJ)
      GO TO 102
  120 IF(ITRMAX.NE.1) CALL FNLPOL
      CALL FORCE(KL)
      IF(M2.EQ.0) GO TO 122
      L=KL+(M2-1)*KMAX
      L1=KMAX1+(M2-1)*KMAX2
      DO 130 I=1,4
      SUM=0.
      DO 131 J=1,4
  131 SUM=SUM+CL2(I,J)*X(J,L)
      ASUMZ=ABS(SUM)
      IF(NCONV.NE.1 .OR. ASUMZ.LT.1.E-05) GO TO 130
      DELZ=ABS(Z(I,L1)-SUM)
      ZTEST=EPS*TZMAX(I,M2)
```

```
      IF(DELZ.GT.ZTEST) NCONV=0
  130 Z(I,L1)=SUM
  122 IF(M1.EQ.0) GO TO 123
      L=KL+(M1-1)*KMAX
      L1=KMAX1+(M1-1)*KMAX2
      DO 132 I=1,4
      SUM=0.
      DO 133 J=1,4
  133 SUM=SUM+CL1(I,J)*X(J,L)
      ASUMZ=ABS(SUM)
      IF(NCONV.NE.1 .OR. ASUMZ .LT. 1.E-05) GO TO 132
      DELZ=ABS(Z(I,L1)-SUM)
      ZTEST=EPS*TZMAX(I,M1)
      IF(DELZ.GT.ZTEST) NCONV=0
  132 Z(I,L1)=SUM
  123 IF(MO.EQ.0) GO TO 124
      L=KL+(MO-1)*KMAX
      L1=KMAX1+(MO-1)*KMAX2
      DO 134 I=1,4
      SUM=0.
      DO 135 J=1,4
  135 SUM=SUM+CLO(I,J)*X(J,L)
      ASUMZ=ABS(SUM)
      IF(NCONV.NE.1 .OR. ASUMZ.LT.1.E-06) GO TO 134
      DELZ=ABS(Z(I,L1)-SUM)
      ZTEST=EPS*TZMAX(I,MO)
      IF(DELZ.GT.ZTEST) NCONV=0
  134 Z(I,L1)=SUM
  124 LS=2
      GO TO 150
      END

      SUBROUTINE ABC
      COMMON
     1/BL17/DEL
     2/BL25/E(4,4),F(4,4),G(4,4)
     3/BL1/A(4,4),BEE(4,4),C(4,4)/BL12/TDLI,TDEL
      D2=2./DEL
      DO 1 I=1,4
      DO 1 J=1,4
      DEIJ=D2*E(I,J)
      FIJ=F(I,J)
      BEE(I,J)=-2.*DEIJ+TDEL*G(I,J)
      C(I,J)=DEIJ-FIJ
    1 A(I,J)=DEIJ+FIJ
      RETURN
      END

      SUBROUTINE PANDD(K,MN)
      COMMON
     1/IBL4/KMAX,KL
     2/BL1/A(4,4),BEE(4,4),C(4,4)
     5/BL4/P(4,4,200),X(4,200),ZF1M(4,4,10),ZF2M(4,4,10),ZF3
     4ZF4M(4,4,10)
     1/BL34/DEE(4,4,200),DST(4,4,200)
      DIMENSION TM(4,4),IPIVOT(4),INDEX(4,2),X2(4)
      IKL=K+KMAX*(MN-1)
      KLI=IKL-1
      DO 1 I=1,4
      DO 1 J=1,4
      SUM=0.
      DO 2 L=1,4
    2 SUM=SUM+C(I,L)*P(L,J,KLI)
    1 TM(I,J)=BEE(I,J)-SUM
      CALL MATINV(TM,4,X2,0,DETERM,IPIVOT,INDEX,4,ISCALE)
      DO 5 I=1,4
      DO 5 J=1,4
      SUMA=0.
      SUMC=0.
      DO 6 L=1,4
```

```
      SUMA=SUMA+TM(I,L)*A(L,J)
    6 SUMC=SUMC+TM(I,L)*C(L,J)
      P(I,J,IKL)=SUMA
      DEE(I,J,IKL)=TM(I,J)
    5 DST(I,J,IKL)=SUMC
      RETURN
      END


      SUBROUTINE HJ(K,MN)
      COMMON
     0/BL8/R(200),GAM(200),OMT(200)
     2/BL20/DEOMX(200)
     3/BL11/OMXI(200),PHEE,TO,T2
     3/BL14/LAM2,LSD18,LSD1N
     4/BL15/NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(
     5W3(10)
     6/BL17/DEL
     7/BL23/JAY(4,4),H(4,4)
     8/IBL2/N(10),MNINIT/IBL4/KMAX,KL
      EQUIVALENCE(L2,LAM2)
      REAL L2,LAM2,LSD1N,        LSD18,JAY,NU
      CALL BDB(K,B,DB,D,DD)
      YAH=1.
      IF(K.EQ.1.OR.K.EQ.KMAX)YAH=2.
      D1=(1.-NU)
      GA=GAM(K)
      OX=OMXI(K)
      RA=R(K)
      EN=N(MN)
      ENR=EN/RA
      REG=0.
      IF(YAH.EQ.2.) REG=1.
      OT=OMT(K)
      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)=0.
      H(3,1)=0.
      H(3,2)=DL*OTX*YAH/4.
      ENR2=ENR**2
      H(3,3)=L2*D*D1*(YAH*ENR2+(1.+NU)*GA**2)
      GA2=GA**2
      H(3,4)=L2
      H(4,1)=0.
      H(4,2)=0.
      H(4,3)=-1.
      H(4,4)=0.
      JAY(1,1)=NU*GA*B
      JAY(1,2)=NU*B*ENR
      JAY(1,3)=B*(OX+NU*OT)
      JAY(1,4)=0.
      JAY(2,1)=-B*D1*ENR/2.-DL/8.*OXT*OTX*REG
      JAY(2,2)=-GA*H(2,2)
      JAY(2,3)=-GA*H(2,3)
      JAY(2,4)=0.
      JAY(3,1)=-L2*D*D1*((1.+NU)*GA2*OX+ENR2/4.*OXT*YAH)
      JAY(3,2)=-GA*DL/2.*(     2.*OT*(1.+NU)+OTX/2.*YAH)
      JAY(3,3)=-L2*D*D1*(1.+NU+YAH)*GA*ENR2
      JAY(3,4)=L2*D1*GA
      JAY(4,1)=OX
      JAY(4,2)=0.
      JAY(4,3)=0.
      JAY(4,4)=0.
      DO 1 I=1,4
```

```
      DO 1 J=1,4
    1 H(I,J)=H(I,J)/2./DEL
      RETURN
      END

      SUBROUTINE TEAETA(K)
      REAL NU,MT
      COMMON/BL5/TT(10),MT(10),DT(10),DMT(10)
     1/IBL1/MNMAX
     2/IBL2/N(10),MNINIT
     3/IBL7/MNMAXO,MAXD(10),MAXS(10),MAXSY(10),IS(10,10),JS(
     4,10),JD(10,10),IJS(10)
     1/IBL8/LSTEP,ITR
     2/IBL13/ITRMAX,LSMAX
     8/BL6/Z(4,220),SOE,OSE,ALOAD
     6/BL7/D1,S1
     0/BL8/R(200),GAM(200),OMT(200)
     8/BL12/TDLI,TDEL
     9/BL15/NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(
     OW3(10)
     1/BL27/BX3(10),BT3(10),BXT3(10),BE3(10)
     2/BL10/PHIX(10),PHIT(10),PHI(10)
     3/BL28/EXX3(10),ETT3(10),ETX3(10),EXT3(10),EX3(10),ET3(
     3/BL11/OMXI(200),PHEE,TO,T2
      DIMENSION TX(10),TTH(10),TXT(10)
      RRA=1./R(K)
      GA=GAM(K)
      OX=OMXI(K)
      OT=OMT(K)
      CALL BDB(K,BS,DB,DS,DD)
      DO 1 M=1,MNMAXO
      EN=N(M)
      CALL TLOAD(K)
      TTS=TT(M)*ALOAD
      EX=(U3(M)-U1(M))*TDLI+OX*W2(M)+OSE*(BX3(M)+BE3(M))
      ET= EN *V2(M)*RRA+GA*U2(M)+OT*W2(M)+OSE*(BT3(M)+BE3(M)
      EXT=.5*(TDLI*(V3(M)-V1(M))- EN *U2(M)*RRA-GA*V2(M))+OS
      TX(M)=BS*(EX+NU*ET)-TTS
      TTH(M)=BS*(ET+NU*EX)-TTS
    1 TXT(M)=BS*D1*EXT
      DO 9 M=1,MNMAX
      SMF=0.
      SMS=0.
      SMV=0.
      SME=0.
      SMN=0.
      SMT=0.
      IF(N(M).EQ.0) GO TO 20
      MAXL=MAXS(M)
      IF(MAXL.EQ.0) GO TO 2
      DO 3 L=1,MAXL
      I=IS(L,M)
      J=JS(L,M)
      SMF=SMF+TX(I)*PHIX(J)+TX(J)*PHIX(I)
      SMS=SMS+TTH(I)*PHIT(J)+TTH(J)*PHIT(I)
      SMV=SMV-PHIT(I)*TXT(J)-PHIT(J)*TXT(I)
      SME=SME+PHIX(I)*TXT(J)+PHIX(J)*TXT(I)
      SMN=SMN+TX(I)*PHI(J)+TX(J)*PHI(I)
    3 SMT=SMT+TTH(I)*PHI(J)+TTH(J)*PHI(I)
    2 MAXL=MAXD(M)
      IF(MAXL.EQ.0) GO TO 4
      DO 5 L=1,MAXL
      I=ID(L,M)
      J=JD(L,M)
      SMF=SMF+TX(I)*PHIX(J)+TX(J)*PHIX(I)
      SMS=SMS-TTH(I)*PHIT(J)+TTH(J)*PHIT(I)
      SMV=SMV+PHIT(I)*TXT(J)+PHIT(J)*TXT(I)
      SME=SME-PHIX(I)*TXT(J)+PHIX(J)*TXT(I)
      SMN=SMN-TX(I)*PHI(J)+TX(J)*PHI(I)
    5 SMT=SMT-TTH(I)*PHI(J)+TTH(J)*PHI(I)
    4 IF(MAXSY(M).EQ.0) GO TO 10
```

```
      I=IJS(M)
      SMF=SMF+TX(I)*PHIX(I)
      SMS=SMS+TTH(I)*PHIT(I)
      SMV=SMV-PHIT(I)*TXT(I)
      SME=SME+PHIX(I)*TXT(I)
      SMN=SMN+TX(I)*PHI(I)
      SMT=SMT+TTH(I)*PHI(I)
      GO TO 10
   20 DO 21 L=1,MNMAXO
      SMF=SMF+TX(L)*PHIX(L)
   21 SMV=SMV+PHIT(L)*TXT(L)
      IF(M.GT.MNMAXO) GO TO 10
      SMF=SMF+TX(M)*PHIX(M)
   10 EXX3(M)=SMF*.5
      ETT3(M)=SMS*.5
      ETX3(M)=SMV*.5
      EXT3(M)=SME*.5
      EX3(M)=SMN*.5
      ET3(M)=SMT*.5
    9 CONTINUE
      DO 30 M=1,MNMAXO
      U1(M)=U2(M)
      V1(M)=V2(M)
      W1(M)=W2(M)
      U2(M)=U3(M)
      V2(M)=V3(M)
   30 W2(M)=W3(M)
      RETURN
      END


      SUBROUTINE FORCE(K)
      REAL NU,MT,LAM2,MASS,MAS
      COMMON
     1/IBL1/MNMAX/BL5/TT(10),MT(10),DT(10),DMT(10)
     2/IBL2/N(10),MNINIT
     3/IBL4/KMAX,KL
     1/IBL8/LSTEP,ITR/IBL12/KMAX1,KMAX2,NCONV
     2/IBL13/ITRMAX,LSMAX
     5/BL4/P(4,4,200),X(4,200),ZF1M(4,4,10),ZF2M(4,4,10),ZF3
     5ZF4M(4,4,10)
     8/BL6/Z(4,220),SOE,OSE,ALOAD
     7/BL7/D1,S1
     0/BL8/R(200),GAM(200),OMT(200)
     9/BL9/FFS(4,10),ELIS(4),GEES(4,10)
     3/BL11/OMXI(200),PHEE,TO,T2
     1/BL12/TDLI,TDEL
     2/BL14/LAM2,LSD18,LSD1N
     3/BL15/NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(
     4W3(10)
     5/BL17/DEL
     6/BL24/DL(4,4,10),DG(4,4,10),DF(4,4,10)
      COMMON
     1/BL27/BX3(10),BT3(10),BXT3(10),BE3(10)
     2/BL28/EXX3(10),ETT3(10),ETX3(10),EXT3(10),EX3(10),ET3(
     3/BL29/BX1(10),BT1(10),BXT1(10),BE1(10),BX2(10),BT2(10)
     4BE2(10)
     5/BL30/EXX1(10),ETT1(10),ETX1(10),EX1(10),ET1(10),EXX2(
     6,ETX2(10),EXT2(10),EX2(10),ET2(10)
     7/BL31/DELSQ,EXT1(10)/BL3/PR(10),PX(10),PT(10)
     1/BL34/DEE(4,4,200),DST(4,4,200)
      COMMON /BL100/SORD,TEEO/BL101/ZO(4,220),Z2(4,220),Z3(4
     1/BL102/DELOAD/BL103/MASS(200)
      DIMENSION GEE(4)
      FDIFF(A,B,C)=(-1.5*A+2.*B-.5*C)/DEL
      RS=R(K)
      RR=1./RS
      GA=GAM(K)
      OX=OMXI(K)
      OT=OMT(K)
      DL2=D1*LAM2
      CALL BDB(K,BS,DBS,D,DD)
```

```
      CALL PLOAD(K)
      CALL TLOAD(K)
      MAS=MASS(K)
      DO 4 M=1,MNMAX
      IZ=K+1+(M-1)*KMAX2
      IK=K+(M-1)*KMAX
      IK1=IK-1
      EN=N(M)
      ENR=EN*RR
      EMT=MT(M)
      GEE(1) =   (-PX(M)+DT(M)-DL2*GA*OX*EMT)*TDEL*ALOAD
     1   + MAS*(-5.*ZO(1,IZ )+4.*Z2(1,IZ)-Z3(1,IZ))*TDEL/DELS
      GEE(2) =   (-PT(M)-ENR*TT(M)-DL2*ENR*OT*EMT)*TDEL*ALOAD
     1   + MAS*(-5.*ZO(2,IZ )+4.*Z2(2,IZ)-Z3(2,IZ))*TDEL/DELS
      GEE(3) =   (-PR(M)-(OX+OT)*TT(M)-DL2*(GA*DMT(M)-(OX*OT-
     1*MT(M)))*TDEL*ALOAD
     1   + MAS*(-5.*ZO(3,IZ )+4.*Z2(3,IZ)-Z3(3,IZ))*TDEL/DELS
     1 GEE(4) =   MT(M)*TDEL*ALOAD
      IF(ITRMAX.EQ.1) GO TO 50
      IF(K.GT.1) GO TO 6
      BX2T=BX1(M)
      BT2T=BT1(M)
      BXT2T=BXT1(M)
      BE2T=BE1(M)
      EXX2T=EXX1(M)
      ETT2T=ETT1(M)
      ETX2T=ETX1(M)
      EXT2T=EXT1(M)
      EX2T=EX1(M)
      ET2T=ET1(M)
      DBX= FDIFF(BX2T,BX2(M),BX3(M))
      DBT= FDIFF(BT2T,BT2(M),BT3(M))
      DBXT= FDIFF(BXT2T,BXT2(M),BXT3(M))
      DBE= FDIFF(BE2T,BE2(M),BE3(M))
      DET= FDIFF(ET2T,ET2(M),ET3(M))
      DEX= FDIFF(EX2T,EX2(M),EX3(M))
      DEXX= FDIFF(EXX2T,EXX2(M),EXX3(M))
      DETX= FDIFF(ETX2T,ETX2(M),ETX3(M))
      GO TO 7
    6 IF(K.LT.KMAX) GO TO 8
      BX2T=BX3(M)
      BT2T=BT3(M)
      BXT2T=BXT3(M)
      BE2T=BE3(M)
      EXX2T=EXX3(M)
      ETT2T=ETT3(M)
      ETX2T=ETX3(M)
      EXT2T=EXT3(M)
      EX2T=EX3(M)
      ET2T=ET3(M)
      DBX=-FDIFF(BX2T,BX2(M),BX1(M))
      DBT=-FDIFF(BT2T,BT2(M),BT1(M))
      DBXT=-FDIFF(BXT2T,BXT2(M),BXT1(M))
      DBE=-FDIFF(BE2T,BE2(M),BE1(M))
      DET=-FDIFF(ET2T,ET2(M),ET1(M))
      DEX=-FDIFF(EX2T,EX2(M),EX1(M))
      DEXX=-FDIFF(EXX2T,EXX2(M),EXX1(M))
      DETX=-FDIFF(ETX2T,ETX2(M),ETX1(M))
      GO TO 7
    8 DBX=TDLI*(BX3(M)-BX1(M))
      DBT=TDLI*(BT3(M)-BT1(M))
      DBE=TDLI*(BE3(M)-BE1(M))
      DBXT=TDLI*(BXT3(M)-BXT1(M))
      DET=TDLI*(ET3(M)-ET1(M))
      DEX=TDLI*(EX3(M)-EX1(M))
      DEXX=TDLI*(EXX3(M)-EXX1(M))
      DETX=TDLI*(ETX3(M)-ETX1(M))
      BX2T=BX2(M)
      BT2T=BT2(M)
      BXT2T=BXT2(M)
      BE2T=BE2(M)
      EXX2T=EXX2(M)
```

```
      ETT2T=ETT2(M)
      ETX2T=ETX2(M)
      EXT2T=EXT2(M)
      EX2T=EX2(M)
      ET2T=ET2(M)
    7 GEE(1)=GEE(1)-OSE*(BS*(DBX+DBE+GA*D1*(BX2T-BT2T)+NU*(D
     1         +ENR*D1*BXT2T)+DBS*(BX2T+BE2T+NU*(BT2T+BE2T))-
     2                (EXX2T+ETX2T)-ENR*(EX2T+ET2T))*TDEL
      GEE(2)=GEE(2)+OSE*(BS*(ENR*(BT2T+BE2T+NU*(BX2T+BE2T))-
     1      (DBXT+2.*GA*BXT2T))-D1*DBS*BXT2T+2.*OT*(ETT2T+EXT
     2       -(DEX+DET))*TDEL
      GEE(3)=GEE(3)+OSE*(BS*((OX+NU*OT)*(BX2T+BE2T)+(OT+NU*O
     1       (BT2T+BE2T))+2.*(GA*(EXX2T+ETX2T)+DEXX+DETX+ENR
     2       (EXT2T+ETT2T)))*TDEL
   50 IF(K.GT.1) GO TO 10
      DO 20 I=1,4
      GEES(I,M)=GEE(I)
      SUMX=0.
      DO 21 J=1,4
   21 SUMX=SUMX+DL(I,J,M)*EL1S(J)+DG(I,J,M)*GEE(J)+DF(I,J,M)
   20 X(I,IK)=SUMX
      GO TO 4
   10 DO 11 I=1,4
      SUMX=0.
      DO 12 J=1,4
   12 SUMX=SUMX+DEE(I,J,IK)*GEE(J)-DST(I,J,IK)*X(J,IK1)
   11 X(I,IK)=SUMX
    4 CONTINUE
      RETURN
      END

      SUBROUTINE UPDATE
      COMMON
     1/IBL1/MNMAX
     2/BL27/BX3(10),BT3(10),BXT3(10),BE3(10)
     3/BL28/EXX3(10),ETT3(10),ETX3(10),EXT3(10),EX3(10),ET3(
     4/BL29/BX1(10),BT1(10),BXT1(10),BE1(10),BX2(10),BT2(10)
     5BE2(10)
     6/BL30/EXX1(10),ETT1(10),ETX1(10),EX1(10),ET1(10),EXX2(
     7,ETX2(10),EXT2(10),EX2(10),ET2(10)
     8/BL31/DELSQ,EXT1(10)
      DO 1 M=1,MNMAX
      BX1(M)=BX2(M)
      BT1(M)=BT2(M)
      BXT1(M)=BXT2(M)
      BE1(M)=BE2(M)
      BX2(M)=BX3(M)
      BT2(M)=BT3(M)
      BXT2(M)=BXT3(M)
      BE2(M)=BE3(M)
      EXX1(M)=EXX2(M)
      ETT1(M)=ETT2(M)
      ETX1(M)=ETX2(M)
      EXT1(M)=EXT2(M)
      EX1(M)=EX2(M)
      ET1(M)=ET2(M)
      EXX2(M)=EXX3(M)
      ETT2(M)=ETT3(M)
      ETX2(M)=ETX3(M)
      EXT2(M)=EXT3(M)
      EX2(M)=EX3(M)
    1 ET2(M)=ET3(M)
      RETURN
      END
```

```
      SUBROUTINE MATINV(A,N,B,M,DETERM,IPIVOT,INDEX,NMAX,ISC
C     MATRIX INVERSION WITH ACCOMPANYING SOLUTION OF LINEAR
C
C
      DIMENSION IPIVOT(N),A(NMAX,N),B(NMAX,M),INDEX(NMAX,2)
      EQUIVALENCE (IROW,JROW), (ICOLUM,JCOLUM), (AMAX, T, SW
C
C     INITIALIZATION
C
    5 ISCALE=0
    6 R1=10.0**18
    7 R2=1.0/R1
   10 DETERM=1.0
   15 DO 20 J=1,N
   20 IPIVOT(J)=0
   30 DO 550 I=1,N
C
C     SEARCH FOR PIVOT ELEMENT
C
   40 AMAX=0.0
   45 DO 105 J=1,N
   50 IF (IPIVOT(J)-1) 60, 105, 60
   60 DO 100 K=1,N
   70 IF (IPIVOT(K)-1) 80, 100, 740
   80 IF (ABS(AMAX)-ABS(A(J,K)))85,100,100
   85 IROW=J
   90 ICOLUM=K
   95 AMAX=A(J,K)
  100 CONTINUE
  105 CONTINUE
  110 IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1
C
C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
C
  130 IF (IROW-ICOLUM) 140, 260, 140
  140 DETERM=-DETERM
  150 DO 200 L=1,N
  160 SWAP=A(IROW,L)
  170 A(IROW,L)=A(ICOLUM,L)
  200 A(ICOLUM,L)=SWAP
  205 IF(M) 260, 260, 210
  210 DO 250 L=1, M
  220 SWAP=B(IROW,L)
  230 B(IROW,L)=B(ICOLUM,L)
  250 B(ICOLUM,L)=SWAP
  260 INDEX(I,1)=IROW
  270 INDEX(I,2)=ICOLUM
  310 PIVOT=A(ICOLUM,ICOLUM)
C
C     SCALE THE DETERMINANT
C
 1000 PIVOTI=PIVOT
 1005 IF(ABS(DETERM)-R1)1030,1010,1010
 1010 DETERM=DETERM/R1
      ISCALE=ISCALE+1
      IF(ABS(DETERM)-R1)1060,1020,1020
 1020 DETERM=DETERM/R1
      ISCALE=ISCALE+1
      GO TO 1060
 1030 IF(ABS(DETERM)-R2)1040,1040,1060
 1040 DETERM=DETERM*R1
      ISCALE=ISCALE-1
      IF(ABS(DETERM)-R2)1050,1050,1060
 1050 DETERM=DETERM*R1
      ISCALE=ISCALE-1
 1060 IF(ABS(PIVOTI)-R1)1090,1070,1070
 1070 PIVOTI=PIVOTI/R1
      ISCALE=ISCALE+1
```

```
      IF(ABS(PIVOTI)-R1)320,1080,1080
 1080 PIVOTI=PIVOTI/R1
      ISCALE=ISCALE+1
      GO TO 320
 1090 IF(ABS(PIVOTI)-R2)2000,2000,320
 2000 PIVOTI=PIVOTI*R1
      ISCALE=ISCALE-1
      IF(ABS(PIVOTI)-R2)2010,2010,320
 2010 PIVOTI=PIVOTI*R1
      ISCALE=ISCALE-1
  320 DETERM=DETERM*PIVOTI
C
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT
C
  330 A(ICOLUM,ICOLUM)=1.0
  340 DO 350 L=1,N
  350 A(ICOLUM,L)=A(ICOLUM,L)/PIVOT
  355 IF(M) 380, 380, 360
  360 DO 370 L=1,M
  370 B(ICOLUM,L)=B(ICOLUM,L)/PIVOT
C
C     REDUCE NON-PIVOT ROWS
C
  380 DO 550 L1=1,N
  390 IF(L1-ICOLUM) 400, 550, 400
  400 T=A(L1,ICOLUM)
  420 A(L1,ICOLUM)=0.0
  430 DO 450 L=1,N
  450 A(L1,L)=A(L1,L)-A(ICOLUM,L)*T
  455 IF(M) 550, 550, 460
  460 DO 500 L=1,M
  500 B(L1,L)=B(L1,L)-B(ICOLUM,L)*T
  550 CONTINUE
C
C     INTERCHANGE COLUMNS
C
  600 DO 710 I=1,N
  610 L=N+1-I
  620 IF (INDEX(L,1)-INDEX(L,2)) 630, 710, 630
  630 JROW=INDEX(L,1)
  640 JCOLUM=INDEX(L,2)
  650 DO 705 K=1,N
  660 SWAP=A(K,JROW)
  670 A(K,JROW)=A(K,JCOLUM)
  700 A(K,JCOLUM)=SWAP
  705 CONTINUE
  710 CONTINUE
  740 RETURN
      END

      SUBROUTINE PMATRX
      REAL JAY
      COMMON
     1/IBL1/MNMAX
     2/IBL2/N(10),MNINIT
     3/IBL3/MO,M1,M2,M3
     4/IBL4/KMAX,KL
     5/IBL5/IBCINL,IBCFNL
     6/BL1/A(4,4),BEE(4,4),C(4,4)
     5/BL4/P(4,4,200),X(4,200),ZF1M(4,4,10),ZF2M(4,4,10),ZF3
     8ZF4M(4,4,10)
     9/BL13/OMEG1(4,4),CAPL1(4,4),OMEGL(4,4),CAPLL(4,4),UNIT
     A/BL23/JAY(4,4),H(4,4)
     B/BL24/DL(4,4,10),DG(4,4,10),DF(4,4,10)
     C/BL25/E(4,4),F(4,4),G(4,4)
      DIMENSION PATA(4,4),PBTA(4,4),POTA(4,4),PJTA(4,4),DLL(
     14),DGG(4,4),ZF1(4,4),ZF2(4,4),ZFPO(4,4),ZFP1(4,4),ZFP2
     2T(4),INDEX(4,2),CLO(4,4),CL1(4,4),CL2(4,4),G1(4)
      EQUIVALENCE (CLO(1),ZF1M(1)),(CL1(1),ZF2M(1)),(CL2(1),
     1(ZFPO(1),PATA(1)),(ZFP1(1),PBTA(1)),(ZFP2(1),POTA(1))
     2,(ZF1(1),DLL(1)),(ZF2(1),PTR(1))
```

```
      IF(IBCINL.LT.0) GO TO 10
      DO 1 MN=MNINIT,MNMAX
      CALL HJ(1,MN)
      CALL EFG(1,MN)
      CALL ABC
      CALL MATINV(C,4,G1,0,DETERM,IPIVOT,INDEX,4,ISCALE)
      DO 3 I=1,4
      DO 3 J=1,4
      SUMA=0.
      SUMB=0.
      SUMO=0.
      SUMJ=0.
      DO 4 L=1,4
      SUMJ=SUMJ+OMEG1(I,L)*JAY(L,J)
      SUMA=SUMA+C(I,L)*A(L,J)
      SUMB=SUMB+C(I,L)*BEE(L,J)
    4 SUMO=SUMO+OMEG1(I,L)*H(L,J)
      PATA(I,J)=SUMA+UNIT(I,J)
      PBTA(I,J)=SUMB
      POTA(I,J)=SUMO
    3 PJTA(I,J)=SUMJ
      DO 5 I=1,4
      DO 5 J=1,4
      SUMOB=0.
      SUMOA=0.
      SUMOC=0.
      DO 6 L=1,4
      SUMOB=SUMOB+POTA(I,L)*PBTA(L,J)
      SUMOA=SUMOA+POTA(I,L)*PATA(L,J)
    6 SUMOC=SUMOC+POTA(I,L)*C(L,J)
      DLL(I,J)=SUMOB+PJTA(I,J)+CAPL1(I,J)
      PTR(I,J)=SUMOA
    5 DGG(I,J)=SUMOC
      CALL MATINV(DLL,4,PTR,4,DETERM,IPIVOT,INDEX,4,ISCALE)
      DO 1 I=1,4
      DO 1 J=1,4
      SUMD=0.
      SUME=0.
      DO 7 L=1,4
      SUMD=SUMD+DLL(I,L)*DGG(L,J)
    7 SUME=SUME-DLL(I,L)*OMEG1(L,J)
      DL(I,J,MN)=DLL(I,J)
      DG(I,J,MN)=SUMD
      DF(I,J,MN)=SUME
      IJ=1+KMAX*(MN-1)
    1 P(I,J,IJ)=PTR(I,J)
      GO TO 20
   10 DO 11 MN=MNINIT,MNMAX
      NN=N(MN)
      IJ=1+KMAX*(MN-1)
      DO 14 I=1,4
      X(I,IJ)=0.
      DO 14 J=1,4
   14 P(I,J,IJ)=0.
      IF(NN.GT.3) GO TO 11
      IF(NN.GT.2) GO TO 90
      IF(NN.GT.1) GO TO 12
      IF(NN.GT.0) GO TO 13
      P(3,3,IJ)=-1.
      P(4,4,IJ)=-1.
      MO=MN
      GO TO 11
   12 P(4,4,IJ)=-1.
      M2=MN
      GO TO 11
   90 M3=MN
      GO TO 11
   13 P(1,1,IJ)=-1.
      P(2,1,IJ)=1.
      M1=MN
   11 CONTINUE
   20 KLAST=KMAX
```

```
      IF(IBCFNL.LT.0) KLAST=KL
      DO 23 K=2,KLAST
      DO 23 MN=MNINIT,MNMAX
      CALL EFG(K,MN)
      CALL ABC
   23 CALL PANDD(K,MN)
      IF(IBCFNL.LT.0) GO TO 30
      DO 40 MN=MNINIT,MNMAX
      IKL=MN*KMAX-1
      JKL=KMAX*MN
      CALL HJ(KMAX,MN)
      DO 41 I=1,4
      DO 41 J=1,4
      SUMO=0.
      SUMP=0.
      SUMJ=0.
      DO 42 L=1,4
      SUMO=SUMO+OMEGL(I,L)*H(L,J)
      SUMP=SUMP+P(I,L,IKL)*P(L,J,JKL)
   42 SUMJ=SUMJ+OMEGL(I,L)*JAY(L,J)
      PATA(I,J)=SUMO
      PBTA(I,J)=UNIT(I,J)-SUMP
   41 PJTA(I,J)=SUMJ+CAPLL(I,J)
      DO 43 I=1,4
      DO 43 J=1,4
      SUMOP=0.
      SUMJP=0.
      SUMOM=0.
      DO 44 L=1,4
      SUMOP=SUMOP+PATA(I,L)*PBTA(L,J)
      SUMJP=SUMJP+PJTA(I,L)*P(L,J,JKL)
   44 SUMOM=SUMOM-PATA(I,L)*P(L,J,IKL)
      ZF1(I,J)=SUMOP-SUMJP
   43 ZF2(I,J)=SUMOM-PJTA(I,J)
      CALL MATINV(ZF1,4,ZF2,4,DETERM,IPIVOT,INDEX,4,ISCALE)
      DO 45 I=1,4
      DO 45 J=1,4
      SZF3=0.
      SZF4=0.
      DO 46 L=1,4
      SZF3=SZF3+ZF1(I,L)*PATA(L,J)
   46 SZF4=SZF4-ZF1(I,L)*OMEGL(L,J)
      ZF3M(I,J,MN)=SZF3
      ZF4M(I,J,MN)=SZF4
      ZF1M(I,J,MN)=ZF1(I,J)
   45 ZF2M(I,J,MN)=ZF2(I,J)
   40 CONTINUE
      RETURN
   30 DO 31 MN=MNINIT,MNMAX
      IKL=MN*KMAX-1
      NN=N(MN)
      IF(NN   .GT.3) GO TO 31
      IF(NN.GT.2) GO TO 300
      IF(NN   .GT.1) GO TO 33
      IF(NN   .GT.0) GO TO 34
      MO=MN
      DO 35 J=1,4
      DO 35 I=1,4
      CLO(I,J)=0.
   35 ZFPO(I,J)=0.
      ZFPO(1,1)=1.
      ZFPO(2,2)=1.
      ZFPO(3,1)=P(3,1,IKL)
      ZFPO(3,2)=P(3,2,IKL)
      ZFPO(3,3)=P(3,3,IKL)+1.
      ZFPO(3,4)=P(3,4,IKL)
      ZFPO(4,1)=P(4,1,IKL)
      ZFPO(4,2)=P(4,2,IKL)
      ZFPO(4,3)=P(4,3,IKL)
      ZFPO(4,4)=P(4,4,IKL)+1.
      CLO(3,3)=1.
      CLO(4,4)=1.
```

```
      CALL MATINV(ZFPO,4,CLO,4,DETERM,IPIVOT,INDEX,4,ISCALE)
      GO TO 31
  300 M3=MN
      GO TO 31
   34 M1=MN
      DO 60 J=1,4
      DO 60 I=1,4
      CL1(I,J)=0.
   60 ZFP1(I,J)=0.
      ZFP1(1,1)=P(1,1,IKL)+1.
      ZFP1(1,2)=P(1,2,IKL)
      ZFP1(1,3)=P(1,3,IKL)
      ZFP1(1,4)=P(1,4,IKL)
      ZFP1(2,1)=1.
      ZFP1(2,2)=-1.
      ZFP1(3,3)=1.
      ZFP1(4,4)=1.
      CL1(1,1)=1.
      CALL MATINV(ZFP1,4,CL1,4,DETERM,IPIVOT,INDEX,4,ISCALE)
      GO TO 31
   33 M2=MN
      DO 70 J=1,4
      DO 70 I=1,4
      CL2(I,J)=0.
   70 ZFP2(I,J)=0.
      ZFP2(1,1)=1.
      ZFP2(2,2)=1.
      ZFP2(3,3)=1.
      ZFP2(4,1)=P(4,1,IKL)
      ZFP2(4,2)=P(4,2,IKL)
      ZFP2(4,3)=P(4,3,IKL)
      ZFP2(4,4)=P(4,4,IKL)+1.
      CL2(4,4)=1.
      CALL MATINV(ZFP2,4,CL2,4,DETERM,IPIVOT,INDEX,4,ISCALE)
   31 CONTINUE
      RETURN
      END

      SUBROUTINE POLE(K)
      REAL NU,MT,MX,MTH,MXT,MTS,KX,KT,KXT,LAM,LAM2,MASS
      COMMON /IBL2/N(10),MNINIT
     2/IBL3/MO,M1,M2,M3
     1/IBL4/KMAX,KL
     2/IBL5/IBCINL,IBCFNL
     3/IBL7/MNMAXO,MAXD(10),MAXS(10),MAXSY(10),IS(10,10),JS(
     4,10),JD(10,10),IJS(10)
     5/IBL8/LSTEP,ITR
     6/IBL10/IFREQ,NTHMAX
     7/IBL12/KMAX1,KMAX2,NCONV
     8/BL6/Z(4,220),SOE,OSE,ALOAD
     9/BL7/D1,S1
     0/BL8/R(200),GAM(200),OMT(200)
     3/BL11/OMXI(200),PHEE,TO,T2
     4/BL20/DEOMX(200)
     1/BL10/PHIX(10),PHIT(10),PHI(10)
     3/BL12/TDLI,TDEL
     4/BL14/LAM2,LSD18,LSD1N
     5/BL15/NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(
     6W3(10)
     8/BL27/BX3(10),BT3(10),BXT3(10),BE3(10)
      COMMON/BL32/TKN,ELAST,CHAR,SIGO
     1/BL31/DELSQ,EXT1(10)
     2/BL17/DEL
     3/BL19/TH(6)
      COMMON/BL5/TT(10),MT(10),DT(10),DMT(10)
      COMMON/BL4/P(4,4,200),X(4,200),ZF1M(4,4,10),ZF2M(4,4,1
     1,ZF3M(4,4,10),ZF4M(4,4,10)
      COMMON /BL100/SORD,TEEO/BL101/ZO(4,220),Z2(4,220),Z3(4
     1/BL102/DELOAD/BL103/MASS(200)
     2/BL110/TX(10),TTH(10),TXT(10),MX(10),MTH(10),MXT(10),Q
     3/BL111/ABZ,ABZO,ABZN,ABZ3,DD2
```

```
      CALL BDB(K,BS,DB,DS,DD)
      IF(K.EQ.KMAX) GO TO 301
      DO 202 MN=1,MNMAXO
      U1(MN)=U2(MN)
      V1(MN)=V2(MN)
      W1(MN)=W2(MN)
      I3=3+(MN-1)*KMAX2
      I2=I3-1
      U2(MN)=Z(1,I3)
      V2(MN)=Z(2,I3)
      W2(MN)=Z(3,I3)
      PHIX(MN)=0.
      PHI T(MN)=0.
      PHI(MN)=0.
      MX(MN)=Z(4,I2)*ABZ3
      MTH(MN)=0.
      MXT(MN)=0.
      QS(MN)=0.
      TX(MN)=0.
      TTH(MN)=0.
  202 TXT(MN)=0.
      IF(M1.EQ.0) GO TO 203
      CALL INLPOL
      PHIX(M1)=PHEE*ABZO
      PHIT(M1)=-PHEE*ABZO
      II=3+(M1-1)*KMAX2
      IP1=II+1
      IM1=II-1
      GA2=GAM(2)
      CALL BDB(2,BS,DB,DS,DD)
      CALL TLOAD(2)
      PHIXX=-Z(3,IP1)*TDLI+OMXI(2)*Z(1,II)
      PHITT=Z(3,II)/R(2)+OMT(2)*Z(2,II)
      PHII=((Z(2,IP1)-Z(2,IM1))*TDLI+GA2*Z(2,II)+Z(1,II)/R(2
      QS(M1)=SIGO*TKN*LAM2*((2.-NU)*Z(4,II)-DS*DD2*(PHITT/R(
     1   *GA2)+D1*MT(M1)*ALOAD+DS*D1*(-PHIXX/R(2)-GA2*PHITT+
     2       (2))*PHII+(Z(3,IP1)*TDLI-GA2*Z(3,II))/R(2)+GA2*(
     3      (2))*Z(2,II)+OMT(2)*(Z(2,IP1)-Z(2,IM1))*TDLI)*.5)
      IF(MO.EQ.0) GO TO 204
       TX(MO) =TO*ABZ
      TTH(MO) =TO*ABZ
      MTH(MO) =MX(MO)
  204 IF(M2.EQ.0) GO TO 205
       TX(M2) = T2*ABZ
      TTH(M2) =-T2*ABZ
      TXT(M2)=-T2*ABZ
      MTH(M2) =-MX(M2)
      MXT(M2) =MTH(M2)
      GO TO 205
  203 IF(MO.EQ.0) GO TO 206
      I3=3+(MO-1)*KMAX2
      I4=I3+1
      CALL TLOAD(1)
      TX(MO)=BS*S1*((2.*Z(1,I3)-.5*Z(1,I4))/DEL+OMXI(1)*Z(3,
     1      -TT(MO)*ABZ*ALOAD
      TTH(MO)=TX(MO)
      MTH(MO)=MX(MO)
  206 IF(M2.EQ.0) GO TO 205
      I3=3+(M2-1)*KMAX2
      I4=I3+1
      TX(M2)=BS*D1*(2.*Z(1,I3)-.5*Z(1,I4))/DEL
       TX(M2) = TX(M2)*ABZ
      TTH(M2) =-TX(M2)
      TXT(M2) =-TX(M2)
      MTH(M2) =-MX(M2)
      MXT(M2) =-MX(M2)
  205 RETURN
  301 CONTINUE
      DO 302 MN=1,MNMAXO
      U1(MN)=U2(MN)
      V1(MN)=V2(MN)
      W1(MN)=W2(MN)
```

```
      PHIX(MN)=0.
      PHIT(MN)=0.
      PHI(MN)=0.
      IK=KMAX1+(MN-1)*KMAX2
      MX(MN)=Z(4,IK)*ABZ3
      MTH(MN)=0.
      MXT(MN)=0.
      QS(MN)=0.
      TX(MN)=0.
      TTH(MN)=0.
  302 TXT(MN)=0.
      IF(M1.EQ.0) GO TO 303
      CALL FNLPOL
      PHIX(M1)=PHEE*ABZO
      PHIT(M1)= PHEE*ABZO
      II=KMAX+(M1-1)*KMAX2
      IP1=II+1
      IM1=II-1
      GAK=GAM(KL)
      CALL BDB(KL,BS,DB,DS,DD)
      CALL TLOAD(KL)
      PHIXX=Z(3,IM1)*TDLI+OMXI(KL)*Z(1,II)
      PHITT=Z(3,II)/R(KL)+OMT(KL)*Z(2,II)
      PHII=((Z(2,IP1)-Z(2,IM1))*TDLI+GAK*Z(2,II)+Z(1,II)/R(K
      QS(M1)=-SIGO*TKN*LAM2*((2.-NU)*Z(4,II)-DS*DD2*(PHITT/R
     1*GAK)+D1*MT(M1)*ALOAD-DS*D1*(-PHIXX/R(KL)-GAK*PHITT+(O
     2       (KL))*PHII+(-Z(3,IM1)*TDLI-GAK*Z(3,II))/R(KL)+GA
     3            -OMT(KL))*Z(2,II)+OMT(KL)*(Z(2,IP1)-Z(2,IM1))
     4       /DEL
      IF(MO.EQ.0) GO TO 304
       TX(MO) =TO*ABZ
      TTH(MO) =TO*ABZ
      MTH(MO) =MX(MO)
  304 IF(M2.EQ.0) GO TO 305
       TX(M2) = T2*ABZ
      TTH(M2) =-T2*ABZ
      TXT(M2) = T2*ABZ
      MTH(M2) =-MX(M2)
      MXT(M2) =MX(M2)
      GO TO 305
  303 IF(MO.EQ.0) GO TO 306
      IKM=KMAX+(MO-1)*KMAX2
      IM1=IKM-1
      CALL TLOAD(KMAX)
      TX(MO)=BS*S1*((-2.*Z(1,IKM)+.5*Z(1,IM1))/DEL+OMXI(KMAX
     1           )*ABZ-TT(MO)*ABZ*ALOAD
      TTH(MO) =TX(MO)
      MTH(MO) =MX(MO)
  306 IF(M2.EQ.0) GO TO 305
      IKM=KMAX+(M2-1)*KMAX2
      IM1=IKM-1
      TX(M2)=BS*D1*(-2.*Z(1,IKM)+.5*Z(1,IM1))/DEL
      TX(M2)=TX(M2)*ABZ
      TTH(M2) =-TX(M2)
      TXT(M2) = TX(M2)
      MTH(M2) =-MX(M2)
      MXT(M2) = MX(M2)
  305 RETURN
      END

      SUBROUTINE PHIBET(K)
      REAL NU
      COMMON
     1/IBL1/MNMAX
     2/IBL2/N(10),MNINIT
     3/IBL4/KMAX,KL
     3/IBL7/MNMAXO,MAXD(10),MAXS(10),MAXSY(10),IS(10,10),JS(
     4,10),JD(10,10),IJS(10)
     6/IBL12/KMAX1,KMAX2,NCONV
     2/IBL13/ITRMAX,LSMAX
     8/BL6/Z(4,220),SOE,OSE,ALOAD
```

```
   0/BL8/R(200),GAM(200),OMT(200)
   9/BL10/PHIX(10),PHIT(10),PHI(10)
   3/BL11/OMXI(200),PHEE,TO,T2
   1/BL12/TDLI,TDEL
   2/BL15/NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(
   3W3(10)
   4/BL27/BX3(10),BT3(10),BXT3(10),BE3(10)
      OX=OMXI(K)
      OT=OMT(K)
      RRA=1./R(K)
      GA=GAM(K)
      KP2=K+2
      DO 1 M=1,MNMAXO
      EN=N(M)
      IK=KP2+(M-1)*KMAX2
      U3(M)=Z(1,IK)
      V3(M)=Z(2,IK)
      W3(M)=Z(3,IK)
      PHIX(M)=-TDLI*(W3(M)-W1(M))+OX*U2(M)
      PHIT(M)=EN*W2(M)*RRA+V2(M)*OT
    1 PHI(M)=(TDLI*(V3(M)-V1(M))+GA*V2(M)+EN*U2(M)*RRA)*.5
      IF(ITRMAX.EQ.1) RETURN
      DO 9 M=1,MNMAX
      SMO=0.
      SMT=0.
      SMR=0.
      SMF=0.
      IF(N(M).EQ.0) GO TO 20
      MAXL=MAXS(M)
      IF(MAXL.EQ.0) GO TO 2
      DO 3 L=1,MAXL
      I=IS(L,M)
      J=JS(L,M)
      SMO=SMO+PHIX(I)*PHIX(J)
      SMT=SMT-PHIT(I)*PHIT(J)
      SMR=SMR+PHIX(I)*PHIT(J)+PHIX(J)*PHIT(I)
    3 SMF=SMF-PHI(I)*PHI(J)
    2 MAXL=MAXD(M)
      IF(MAXL.EQ.0) GO TO 4
      DO 5 L=1,MAXL
      I=ID(L,M)
      J=JD(L,M)
      SMO=SMO+PHIX(I)*PHIX(J)
      SMT=SMT+PHIT(I)*PHIT(J)
      SMR=SMR-PHIX(I)*PHIT(J)+PHIX(J)*PHIT(I)
    5 SMF=SMF+PHI(I)*PHI(J)
    4 IF(MAXSY(M).EQ.0) GO TO 10
      I=IJS(M)
      SMO=SMO+PHIX(I)**2/2.
      SMT=SMT-PHIT(I)**2/2.
      SMR=(SMR+PHIX(I)*PHIT(I))
      SMF=SMF-PHI(I)**2/2.
      GO TO 10
   20 DO 21 L=1,MNMAXO
      SMO=SMO+PHIX(L)**2
      SMT=SMT+PHIT(L)**2
   21 SMF=SMF+PHI(L)**2
      IF(M.GT.MNMAXO) GO TO 11
      SMO=SMO+PHIX(M)**2
   11 BX3(M)=SMO*.5
      BT3(M)=SMT*.5
      BE3(M)=SMF*.5
      BXT3(M)=0.
      GO TO 9
   10 BX3(M)=SMO
      BT3(M)=SMT
      BXT3(M)=SMR*.5
      BE3(M)=SMF
    9 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE EFG(K,MN)
      COMMON
     1/IBL2/NN(10),MNINIT
     0/BL8/R(200),GAM(200),OMT(200)
     3/BL11/OMXI(200),PHEE,TO,T2
     4/BL20/DEOMX(200)
     4/BL14/LAM2,LSD18,LSD1N
     5/BL15/NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(
     6W3(10)
     7/BL25/E(4,4),F(4,4),G(4,4)
      COMMON /BL100/SORD,TEEO/BL101/ZO(4,220),Z2(4,220),Z3(4
     1/BL102/DELOAD/BL103/MASS(200)
      REAL NU,N,     LAM2,LSD18,LSD1N,MASS,MAS
      N=NN(MN)
      MAS=MASS(K)
      CALL BDB(K,B,DB,D,DD)
      E(1,1)=B
      E(1,2)=0.
      E(1,3)=0.
      E(1,4)=0.
      E(2,1)=0.
      D1=(1.-NU)
      RA=R(K)
      GA=GAM(K)
      OX=OMXI(K)
      OT=OMT(K)
      DEX=DEOMX(K)
      REX=(3.*OT-OX)
      GA2=GA**2
      RXE=(3.*OX-OT)
      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)=0.
      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)=0.
      E(4,2)=0.
      E(4,3)=-D
      E(4,4)=0.
      F(1,1)=GA*B+DB
      F(1,2)=(1.+NU)*B*N/(2.*RA)+DNLR*REX*RXE/4.
      F(1,3)=B*(OX+NU*OT)+LAM2*D*D1*((1.+NU)*GA2*OX+RAN*RXE/
      F(1,4)=LAM2*OX
      F(2,1)=-F(1,2)
      F(2,2)=(D1/2.)*(GA*B+DB)-(LAM2*D*D1*REX/8.)*(2.*DEX-GA
     1-3.*OT))+LAM2*DD*D1*REX**2/8.
      F(2,3)=DNLR*(2.*(1.+NU)*GA*OT-DEX+3.*GA*(OX-OT))+DDNLR
      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*R
     1+LAM2*DD*D1*((1.+NU)*GA2+2.*RAN)
      F(3,4)=LAM2*GA*(2.-NU)
      F(4,1)=D*OX
      F(4,2)=0.
      F(4,3)=-D*NU*GA
      F(4,4)=0.
      G(1,1)=NU*DB*GA-NU*B*OTX-B*GA2-D1*B*RAN/2.-LAM2*D*D1*(
     1OX**2+RXE**2*RAN/8.)
     2  - 2.*MAS/DELSD
      G(1,2)=NU*N*DB/RA-(3.-NU)/(2.*RA)*GA*B*N-DNLR*2.*GA*(R
     1+(1.+NU)*OTX)
```

```
      G(1,3)=B*(DEX+GA*(OX-OT))+DB*(OX+NU*OT)-LAM2*D*D1*GA*R
     11.+NU)*OX)
      G(1,4)=LAM2*D1*GA*OX
      G(2,1)=-B*GA*N*(3.-NU)/(2.*RA)-D1*N*DB/(2.*RA)+DNLR*2.
     1NU)*GA*OTX+GA/8.*(6.*OTX-7.*OX**2-3.*OT**2)-DEX/4.*(5.
     2-DDNLR/4.*REX*RXE
      G(2,2)=-GA*F(2,2)+D1/2.*B*OTX-B*RAN-LAM2*D*D1*((1.+NU)
     1-OTX/8.*REX**2)
     2  - 2.*MAS/DELSD
      G(2,3)=-B*N*(OT+NU*OX)/RA+DNLR*(GA*DEX-2.*GA2*OX-2.*(1
     1*RAN+REX*(GA2+OTX))-DDNLR*REX*GA
      G(2,4)=-NU*LAM2*OT*N/RA
      G(3,1)=-B*GA*(OT+NU*OX)+LAM2*D*D1*(GA*(1.+NU)*(-GA*DEX
     1-OX*RAN+2.*OTX*OX)+RAN/2.*(GA*OX-GA*OT-3.*DEX))
     2-LAM2*DD*D1*((1.+NU)*GA2*OX+RAN/2.*RXE)
      G(3,2)=-B*N*(OT+NU*OX)/RA+DNLR*(2.*(1.+NU)*(OTX*OT-GA2
     1*OT-OT*RAN)+GA*DEX+3.*GA2*(OT-OX)+OTX*REX)-DDNLR*(2.*(
     2*OT+GA*REX)
      G(3,3)=-B*(OX**2+2.*NU*OTX+OT**2)+LAM2*D*D1*RAN*((1.+N
     1+2.*GA2)+2.*(GA2+OTX))-LAM2*DD*D1*RAN*(3.+NU)*GA
     2  - 2.*MAS/DELSD
      G(3,4)=-LAM2*(D1*OTX+NU*RAN)
      G(4,1)=D*(DEX+NU*GA*OX)
      G(4,2)=D*NU*N*OT/RA
      G(4,3)=D*NU*RAN
      G(4,4)=-1.
      RETURN
      END


      SUBROUTINE OUTPUT(IMODE)
      REAL NU,MT,MX,MTH,MXT,MTS,KX,KT,KXT,LAM,LAM2,MASS
      COMMON /IBL2/N(10),MNINIT
     2/IBL3/MO,M1,M2,M3
     1/IBL4/KMAX,KL
     2/IBL5/IBCINL,IBCFNL
     3/IBL7/MNMAXO,MAXD(10),MAXS(10),MAXSY(10),IS(10,10),JS(
     4,10),JD(10,10),IJS(10)
     5/IBL8/LSTEP,ITR
     6/IBL10/IFREQ,NTHMAX
     7/IBL12/KMAX1,KMAX2,NCONV
     2/IBL13/ITRMAX,LSMAX
      COMMON/BL4/P(4,4,200),X(4,200),ZF1M(4,4,10),ZF2M(4,4,1
     1,ZF3M(4,4,10),ZF4M(4,4,10)
      COMMON/BL5/TT(10),MT(10),DT(10),DMT(10)
     8/BL6/Z(4,220),SOE,OSE,ALOAD
     9/BL7/D1,S1
     0/BL8/R(200),GAM(200),OMT(200)
     3/BL11/OMXI(200),PHEE,TO,T2
     4/BL20/DEOMX(200)
     1/BL10/PHIX(10),PHIT(10),PHI(10)
     3/BL12/TDLI,TDEL
     4/BL14/LAM2,LSD18,LSDIN
     5/BL15/NU,U1(10),V1(10),W1(10),V2(10),U2(10),W2(10),U3(
     6W3(10)
     8/BL27/BX3(10),BT3(10),BXT3(10),BE3(10)
      COMMON/BL32/TKN,ELAST,CHAR,SIGO
     1/BL31/DELSQ,EXT1(10)
     2/BL17/DEL
     3/BL19/TH(6)
      COMMON /BL100/SORD,TEEO/BL101/ZO(4,220),Z2(4,220),Z3(4
     1/BL102/DELOAD/BL103/MASS(200)
     2/BL110/TX(10),TTH(10),TXT(10),MX(10),MTH(10),MXT(10),Q
     3/BL111/ABZ,ABZO,ABZN,ABZ3,DD2
      DIMENSION PTF(200),PF(200)
      ABZO=SIGO/ELAST
      IF(SORD.NE.0) GO TO 181
      WRITE(6,101) LSTEP,ALOAD,ITR
      GO TO 182
  181 TI=LSTEP*DELOAD
      DTI=TI*TEEO
      WRITE(6,151) LSTEP,TI,DTI,ITR
```

```
182 LAM=TKN/CHAR
    ENL=1.
     ABZ=SIGO*TKN
     ABZ3 =ABZ*TKN*TKN/CHAR
     ABZN=CHAR*SIGO/ELAST
    IF(ITRMAX.EQ.1) ENL=0.
    DD2=1.-NU**2
    D2I=1./DD2
    DPI=1./S1
    DNI=1./D1
    TDLSQI=.5/DELSQ
    IF(NTHMAX.EQ.0) GO TO 991
    DO 21NTH=1,NTHMAX
    DO 1 MN=1,MNMAXO
    I1=1+(MN-1)*KMAX2
    I2=I1+1
    U1(MN)=Z(1,I1)
    U2(MN)=Z(1,I2)
    V1(MN)=Z(2,I1)
    V2(MN)=Z(2,I2)
    W1(MN)=Z(3,I1)
  1 W2(MN)=Z(3,I2)
    THET=TH(NTH)
    WRITE(6,116) THET
    DO 121 K=1,KMAX
    K1=K+1
    CALL BDB(K,BS,DB,DS,DD)
    IF(K.EQ.1.AND.IBCINL.LT.0) CALL POLE(K)
    IF(K.EQ.1.AND.IBCINL.LT.0) GO TO 999
    IF(K.EQ.KMAX.AND.IBCFNL.LT.0) CALL POLE(K)
    IF(K.EQ.KMAX.AND.IBCFNL.LT.0) GO TO 999
    CALL PHIBET(K)
    DEX=DEOMX(K)
    RRA=1./R(K)
    OX=OMXI(K)
    OT=OMT(K)
    GA=GAM(K)
    DOXT=OX-OT
    GDO=GA*DOXT
    DD2D=DD2*DS
    DO 3 MN=1,MNMAXO
    EN=N(MN)
    ENR=EN*RRA
    CALL TLOAD(K)
    TTS=TT(MN)*ALOAD
    EX=(U3(MN)-U1(MN))*TDLI +OX*W2(MN) + ENL*OSE*(BX3(MN)+
    ET=ENR*V2(MN) + GA*U2(MN) + OT*W2(MN) + ENL*OSE*(BT3(M
    EXT=.5*((V3(MN)-V1(MN))*TDLI - ENR*U2(MN) - GA*V2(MN)
   1    + ENL*SOE*BXT3(MN))
    KT=ENR*PHIT(MN) + GA*PHIX(MN)
    KXT=.5*(ENR*(-PHIX(MN) -GA*W2(MN) + (W3(MN)-W1(MN))*TD
   1    + GDO*V2(MN) + OT*(V3(MN)-V1(MN))*TDLI
   2    -GA*PHIT(MN)-DOXT*PHI(MN))
    TX(MN)=BS*(EX+NU*ET)-TTS
    TTH(MN)=BS*(ET+NU*EX)-TTS
    TXT(MN)=BS*D1*EXT
    MK1=K1+(MN-1)*KMAX2
    MX(MN)=Z(4,MK1)
    MTH(MN)=NU*MX(MN)+DD2D*KT-D1*MT(MN)*ALOAD
    MXT(MN)=DS*D1*KXT
    MK11=MK1+1
    MKK1=MK1-1
    QS(MN)=SIGO*TKN*LAM2*(GA*MX(MN)+(Z(4,MK11)-Z(4,MKK1))*
   1      +ENR*MXT(MN)-GA*MTH(MN))
      MX(MN)=MX(MN)*ABZ3
     MTH(MN) = MTH(MN)*ABZ3
     MXT(MN)=MXT(MN)*ABZ3
     TX(MN)=TX(MN)*ABZ
     TTH(MN)=TTH(MN)*ABZ
     TXT(MN)=TXT(MN)*ABZ
    PHIX(MN)=PHIX(MN)*ABZO
    PHIT(MN)=PHIT(MN)*ABZO
```

```
      PHI(MN)=PHI(MN)*ABZO
       U1(MN)=U2(MN)
       U2(MN)=U3(MN)
       V1(MN)=V2(MN)
       V2(MN)=V3(MN)
       W1(MN)=W2(MN)
    3  W2(MN)=W3(MN)
      FK=K-1
      FIFREQ=IFREQ
      KTST=(K-1)/IFREQ
      FKTST=KTST
      FKTEST=FK/FIFREQ-FKTST
      IF(K.EQ.1.OR.K.EQ.KMAX) GO TO 999
      IF(FKTEST.NE.0.) GO TO 2
  999 X(1,K)=0.
      X(2,K)=0.
      X(3,K)=0.
      X(4,K)=0.
      PTF(K)=0.
      PF(K)=0.
      AMX=0.
      AMTH=0.
      AMXTH=0.
      ANX=0.
      ANTH=0.
      ANXTH=0.
       AQS = 0.0
      DO 72 MN=1,MNMAXO
      EN=N(MN)
      FC=EN*THET
      SN=SIN(FC)
      CS=COS(FC)
      X(1,K)=X(1,K)+U1(MN)*CS*ABZN
      X(2,K)=X(2,K)+V1(MN)*SN*ABZN
      X(3,K)=X(3,K)+W1(MN)*CS*ABZN
      X(4,K)=X(4,K)+PHIX(MN)*CS
      PTF(K)=PTF(K)+PHIT(MN)*SN
      AMX=AMX+MX(MN)*CS
      AMTH=AMTH+MTH(MN)*CS
      AMXTH=AMXTH+MXT(MN)*SN
      ANX=ANX+TX(MN)*CS
      ANTH=ANTH+TTH(MN)*CS
          AQS = AQS + QS(MN)*CS
      ANXTH=ANXTH+TXT(MN)*SN
   72 PF(K)=PF(K)+PHI(MN)*SN
  429 CONTINUE
      IF(K.EQ.1) WRITE(6,117)
  117 FORMAT(/,8H STATION15H        N S        16H         N THETA
     1N STHETA 16H          Q S        16H        M S        16H
     216H         M STHETA //)
      WRITE(6,118) K,ANX,ANTH,ANXTH,AQS,AMX,AMTH,AMXTH
  118 FORMAT(1X,I3,3X,7E16.4)
  101 FORMAT(1H1,34H                THE LOAD STEP NUMBER IS I2,29H
     1E LOAD FACTOR IS E11.4,36H              THE SOLUTION CONVE
     21H ITERATIONS////)
  151 FORMAT(1H1,26H  THE TIME STEP NUMBER IS I4,18H          TH
     1.2,4H OR E9.3,40H SECONDS        THE SOLUTION CONVERGED
     3ERATIONS////)
  102 FORMAT(1H0,35H                MODAL OUTPUT FOR STATION I3//)
  103 FORMAT(//7H    N    16H          U          16H         V
     1W         16H         PHI S      16H         PHI THETA10H
  104 FORMAT(2X,I2,3X,6E16.4)
  105 FORMAT(7H    N    16H          N S        16H         N THETA
     1STHETA 16H          Q S        16H        M S        16H
     2H          M STHETA /)
  106 FORMAT(1H0,8HSTATION I3,14H- THETA OUTPUT/ )
  107 FORMAT(                        14H      THETA     14H      U
     1 V          14H       W            14H      PHI S     14H       PH
     2    PHI         )
  108 FORMAT(8E14.4)
  109 FORMAT(1H0,                         14H      THETA      14H      M
     1      M THETA    14H      M STHETA  14H      N S           14H
```

```
     2H     N STHETA  14H      Q S         )
  114 FORMAT(2X,I2,3X,7E16.4)
  115 FORMAT(7H    N    16H        N S       16H       N THETA
     1STHETA 16H       M S      16H        M THETA  15H
C 116 FORMAT(1H1,84H              THE SUMMED FORCES, MOMENTS, DI
  116 FORMAT(1H0,84H              THE SUMMED FORCES, MOMENTS, DI
     1AND ROTATIONS FOLLOW FOR THETA =E15.6///)
    2 CONTINUE
  121 CONTINUE
      DO 660 K=1,KMAX
      FK=K-1
      FIFREQ=IFREQ
      KTST=(K-1)/IFREQ
      FKTST=KTST
      FKTEST=FK/FIFREQ-FKTST
      IF(K.EQ.1.OR.K.EQ.KMAX) GO TO 661
      IF(FKTEST.NE.0.) GO TO 658
  661 IF(K.EQ.1) WRITE(6,217)
      WRITE(6,218) K,X(1,K),X(2,K),X(3,K),X(4,K),PTF(K),PF(K
  217 FORMAT(//8H STATION,15H         U         16H       V
     1 W         16H       PHI S     16H        PHI THETA16H
     2 /)
  218 FORMAT(1X,I3,3X,6E16.4)
  658 DO 659 I=1,4
  659 X(I,K)=0.
  660 CONTINUE
   21  CONTINUE
  991 IF(IMODE.LE.0) RETURN
      DO 534 MN=1,MNMAXO
      WRITE(6,749) N(MN)
  749 FORMAT(1H1,40X,27H MODAL OUTPUT FOR MODE N = I3,8H FOL
      DO 521 MM=1,MNMAXO
      I1=1+(MM-1)*KMAX2
      I2=I1+1
      U1(MM)=Z(1,I1)
      U2(MM)=Z(1,I2)
      V1(MM)=Z(2,I1)
      V2(MM)=Z(2,I2)
      W1(MM)=Z(3,I1)
      W2(MM)=Z(3,I2)
  521 CONTINUE
      DO 445 K=1, KMAX
      K1=K+1
      CALL BDB(K,BS,DB,DS,DD)
      IF(K.EQ.1.AND.IBCINL.LT.0) CALL POLE(K)
      IF(K.EQ.KMAX.AND.IBCFNL.LT.0) CALL POLE(K)
      TXZ=TX(MN)
      TTHZ=TTH(MN)
      TXTZ=TXT(MN)
      AMXZ=MX(MN)
      AMTHZ=MTH(MN)
      AMXTZ=MXT(MN)
      QSZ=QS(MN)
      X(1,K)=PHIX(MN)
      X(2,K)=PHIT(MN)
      X(3,K)=PHI(MN)
      IF(K.EQ.1.AND.IBCINL.LT.0) GO TO 583
      IF(K.EQ.KMAX.AND.IBCFNL.LT.0) GO TO 583
      CALL PHIBET(K)
      DEX=DEOMX(K)
      RRA=1./R(K)
      OX=OMXI(K)
      OT=OMT(K)
      GA=GAM(K)
      DOXT=OX-OT
      GDO=GA*DOXT
      DD2D=DD2*DS
      EN=N(MN)
      ENR=EN*RRA
      CALL TLOAD(K)
      TTS=TT(MN)*ALOAD
      EX=(U3(MN)-U1(MN))*TDLI +OX*W2(MN) + ENL*OSE*(BX3(MN)+
```

```
      ET=ENR*V2(MN) + GA*U2(MN) + OT*W2(MN) + ENL*OSE*(BT3(M
      EXT=.5*((V3(MN)-V1(MN))*TDLI - ENR*U2(MN) - GA*V2(MN)
     1     + ENL*SOE*BXT3(MN))
      KT=ENR*PHIT(MN) + GA*PHIX(MN)
      KXT=.5*(ENR*(-PHIX(MN) -GA*W2(MN) + (W3(MN)-W1(MN))*TD
     1     + GDO*V2(MN) + OT*(V3(MN)-V1(MN))*TDLI
     2     -GA*PHIT(MN)-DOXT*PHI(MN))
      TXZ = (BS*(EX+NU*ET)-TTS)*ABZ
      TTHZ = (BS*(ET+NU*EX)-TTS)*ABZ
      TXTZ =  BS*D1*EXT*ABZ
      MK1=K1+(MN-1)*KMAX2
      AMXZ = Z(4,MK1)
      AMTHZ = NU*AMXZ+DD2D*KT-D1*MT(MN)*ALOAD
      AMXTZ = DS*D1*KXT
      MK11=MK1+1
      MKK1=MK1-1
      QSZ =   SIGO*TKN*LAM2*(GA*AMXZ  +(Z(4,MK11)-Z(4,MKK1))*
     1      +ENR*AMXTZ  -GA*AMTHZ)
      AMXZ=AMXZ*ABZ3
      AMTHZ=AMTHZ*ABZ3
      AMXTZ=AMXTZ*ABZ3
      X(1,K) = PHIX(MN)*ABZO
      X(2,K) = PHIT(MN)*ABZO
      X(3,K) =PHI(MN)*ABZO
      DO 533 MM=1,MNMAXO
      U1(MM)=U2(MM)
      U2(MM)=U3(MM)
      V1(MM)=V2(MM)
      V2(MM)=V3(MM)
      W1(MM)=W2(MM)
  533 W2(MM)=W3(MM)
      FK=K-1
      FIFREQ=IFREQ
      KTST=(K-1)/IFREQ
      FKTST=KTST
      FKTEST=FK/FIFREQ-FKTST
      IF(K.EQ.1.OR.K.EQ.KMAX) GO TO 583
      IF(FKTEST.NE.0.) GO TO 445
  583 CONTINUE
      IF(K.EQ.1) WRITE(6,117)
      WRITE(6,118) K,TXZ,TTHZ,TXTZ,QSZ,AMXZ,AMTHZ,AMXTZ
  445 CONTINUE
      WRITE(6,217)
      DO 446 K=1,KMAX
      FK=K-1
      FIFREQ=IFREQ
      KTST=(K-1)/IFREQ
      FKTST=KTST
      FKTEST=FK/FIFREQ-FKTST
      IF(K.EQ.1.OR.K.EQ.KMAX) GO TO 593
      IF(FKTEST.NE.0.) GO TO 446
  593 KZ=K+1+(MN-1)*KMAX2
      UP=Z(1,KZ)*ABZN
      VP=Z(2,KZ)*ABZN
      WP=Z(3,KZ)*ABZN
      WRITE(6,218) K,UP,VP,WP,X(1,K),X(2,K),X(3,K)
  446 CONTINUE
  534 CONTINUE
      RETURN
      END
```

TABLE I. IMPORTANT FORTRAN VARIABLES

FORTRAN Variable	Definition
A(4,4)	A matrix
BEE(4,4)	$\overline{B}$ matrix
BE1(10) BE2(10) BE3(10)	β at i - 1, i, and i + 1
BT1(10) BT2(10) BT3(10)	β_θ at i - 1, i, and i + 1
BX1(10) BX2(10) BX3(10)	β_s at i - 1, i, and i + 1
BXT1(10) BXT2(10) BXT3(10)	$\beta_{s\theta}$ at i - 1, i, and i + 1
C(4,4)	$\underline{C}$ matrix
CAPLL(4,4)	$\overline{\Lambda}_K$, Λ_K
CAPL1(4,4)	$\overline{\Lambda}_1$, Λ_1
DEE(4,4,200)	$\left[\overline{B}_i - C_i P_{i-1}\right]^{-1}$
DELSD	DELØAD*DELØAD
DST(4,4,200)	$\left[\overline{B}_i - C_i P_{i-1}\right]^{-1} C_i$
D1	$1 - \nu$
E(4,4)	E matrix
ELL(4)	ℓ_K
EL1(4)	ℓ_1
ET1(10) ET2(10) ET3(10)	η_θ at i - 1, i, and i + 1
ETT1(10) ETT2(10) ETT3(10)	$\eta_{\theta\theta}$ at i - 1 i, and i + 1
ETX1(10) ETX2(10) ETX3(10)	$\eta_{\theta s}$ at i - 1 i, and i + 1

FORTRAN Variable	Definition
EX1(10)	η_s at i - 1,
EX2(10) EX3(10)	i, and i + 1
EXT1(10)	$\eta_{s\theta}$ at i - 1
EXT2(10) EXT3(10)	i, and i + 1
EXX1(10)	η_{ss} at i - 1,
EXX2(10) EXX3(10)	i, and i + 1
F(4,4)	F matrix
FFS(4, 10)	f_1 matrix
FLS(4)	f_K matrix
G(4,4)	G matrix
GEE(4)	$\bar{g}$ matrix
GEES(4,10)	$\bar{g}_1$ matrix
H(4,4)	H matrix
ID(10,10)	See description of subroutine MODES
IJS(10)	See description of subroutine MODES
IS(10,10)	See description of subroutine MODES
ITR	Iteration number
JAY(4,4)	J matrix
JD(10,10)	See description of subroutine MODES

FORTRAN Variable	Definition
KL	KMAX-1
KLL	KL-1
KMAX1	KMAX+1
KMAX2	KMAX+2
JS(10,10)	See description of subrountine MODES
MAXD(10)	See description of subroutine MODES
MAXS(10)	See description of subroutine MODES
MAXSY(10)	See description of subroutine MODES
MO, M1, M2, and M3	N(MO)=0, N(M1)=1, N(M2)=2, N(M3)=3
N(10) or NN(10)	n
ØMEGL(4,4)	$\bar{\Omega}_K$, Ω_K
ØMEG1(4,4)	$\bar{\Omega}_1$, Ω_1
ØSE	$.5\sigma_o/E_o$
P(4,4,200)	P matrix
PHI(10)	Φ, φ
PHIT(10)	Φ_θ, φ_θ
PHIX(10)	Φ_s, φ_s
SØE	σ_o/E_o
S1	$1 + \nu$

FORTRAN Variable	Definition
TDEL	2Δ
TDLI	1/(2Δ)
TH(6)	θ
U1(10) U2(10) U3(10)	U, u at i - 1, i, and i + 1
V1(10) V2(10 V3(10)	V, v at i - 1, i, and i + 1
W1(10) W2(10) W3(10)	W, w at i - 1, i, and i + 1
X(4,200)	x matrix
Z(4,220)	z matrix
ZDØT(4,220)	∂z/∂t
ZØ(4,220)	z at t - δt
Z2(4,220)	z at t - 2δt
Z3(4,220)	z at t - 3δt

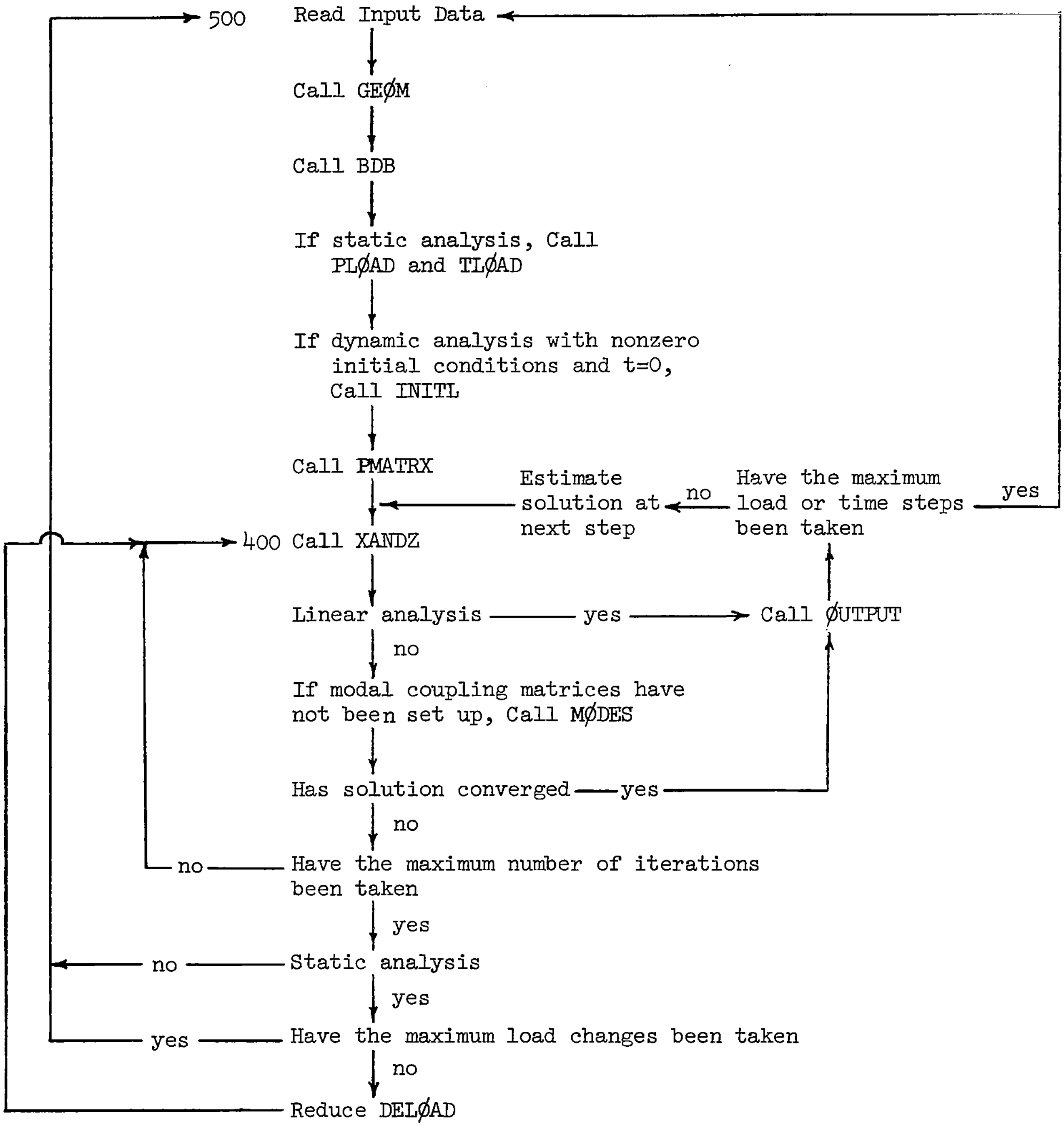

Figure 10. Flow of Program Logic in MAIN

REFERENCES

1. Ball, R. E., "A Geometrically Nonlinear Analysis of Arbitrarily Loaded Shells of Revolution" NASA CR-909 (January 1968)

2. Sanders, J. Lyell, Jr., "Nonlinear Theories for Thin Shells" Quart. Appl. Math. 21 21-36 (1963)

3. Houbolt, John C., "A Recurrence Matrix Solution for the Dynamic Response of Aircraft in Gusts" NACA Rpt. 1010 (1951)

4. Potters, M. L., "A Matrix Method for the Solution of a Second Order Difference Equation in Two Variables" Mathematics Centrum, Amsterdam Holland, Report MR 19 (1955)

5. Stilwell, W. C., "A Digital Computer Study of the Buckling of Shallow Spherical Caps and Truncated Hemispheres," A.E. Thesis, Naval Postgraduate School, (June 1970)

6. Ball, R. E., "A Program for the Nonlinear Static and Dynamic Analysis of Arbitrarily Loaded Shells of Revolution," presented at the AFFDL-LMSC Computer-oriented Analysis of Shell Structures Conf., Palo Alto, (August 1970)(to be published in the Journal of Computers and Structures)

7. Budiansky, B., and Radkowski, P., "Numerical Analysis of Unsymmetrical Bending of Shells of Revolution" AIAA Journal 1 1833-1842 (August 1963) (Discussion by G. A. Greenbaum, 2 590-592 (March 1964))

8. Famili, J., and Archer, R. R., "Finite Asymmetric Deformation of Shallow Spherical Shells," AIAA Journal 3 506-510 (March 1965)