NASA
Technical
Memorandum

NASA TM - 100365

# PRESSURE-VOLUME PROPERTIES OF METALLIC BELLOWS

By Larry Kiefling

Structures and Dynamics Laboratory
Science and Engineering Directorate

May 1989

(NASA-TM-100365) PRESSURE-VOLUME PROPERTIES OF METALLIC BELLOWS (NASA. Marshall Space Flight Center) 59 p CSCL 22B

N89-24422

Unclas
G3/18 0214650

National Aeronautics and
Space Administration

George C. Marshall Space Flight Center

MSFC - Form 3190 (Rev. May 1983)

TECHNICAL REPORT STANDARD TITLE PAGE

| 1. REPORT NO. | 2. GOVERNMENT ACCESSION NO. | 3. RECIPIENT'S CATALOG NO. |
|---|---|---|
| NASA TM - 100365 | | |

| 4. TITLE AND SUBTITLE | 5. REPORT DATE |
|---|---|
| Pressure-Volume Properties of Metallic Bellows | May 1989 |
| | 6. PERFORMING ORGANIZATION CODE |
| 7. AUTHOR(S) Larry Kiefling | 8. PERFORMING ORGANIZATION REPORT # |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>George C. Marshall Space Flight Center<br>Marshall Space Flight Center, Alabama 35812 | 10. WORK UNIT NO. |
| | 11. CONTRACT OR GRANT NO. |
| 12. SPONSORING AGENCY NAME AND ADDRESS<br>National Aeronautics and Space Administration<br>Washington, D.C. 20546 | 13. TYPE OF REPORT & PERIOD COVERED<br>Technical Memorandum |
| | 14. SPONSORING AGENCY CODE |

15. SUPPLEMENTARY NOTES

Prepared by Structures and Dynamics Laboratory, Science and Engineering Directorate.

16. ABSTRACT


Metallic bellows are commonly used as segments of propellant feedlines for rocket-propelled vehicles to accommodate temperature-induced length variations, manufacturing tolerances, and gimbaling of the engines. These bellows sections deform radially and change volume when internal pressure varies, and the magnitude of such deformation is much higher than that for the straight, cylindrical segments of the line. The greater flexibility, or lesser stiffness, of the bellows, decreases the frequency of acoustic oscillations in the line. These acoustic oscillations are a major factor in the so-called POGO phenomena which have plagued most of the larger liquid rocket-propelled vehicles for many years.

A method is developed to calculate the change in volume of a bellows due to a change in internal pressure. Results of an experiment are also presented along with a test-analysis comparison. The computer code is included.


| 17. KEY WORDS | 18. DISTRIBUTION STATEMENT |
|---|---|
| Bellows<br>POGO | Unclassified — Unlimited |

| 19. SECURITY CLASSIF. (of this report) | 20. SECURITY CLASSIF. (of this page) | 21. NO. OF PAGES | 22. PRICE |
|---|---|---|---|
| Unclassified | Unclassified | 61 | NTIS |

MSFC - Form 3292 (Rev. December 1972)

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

TABLE OF CONTENTS

## LIST OF ILLUSTRATIONS

## LIST OF SYMBOLS

| Symbol | Definition |
|---|---|
| $a$ | fluid sonic velocity in elastic pipe |
| $a_{j,k}$ | coefficients in polynomial displacement function for normal displacement w (j = 0,1,...,5) |
| $A$ | cross-sectional area of fluid conduit |
| $A_k$ | matrix which transforms displacements and rotations at the ends of an element to coefficients of polynomial displacement functions [see equation (16) and Appendix A] |
| $b_{o,k}, b_{1,k}$ $b_{2,k}, b_{3,k}$ | coefficients in polynomial displacement function for meridional displacement u |
| $B_k$ | matrix whose elements are coefficients in an expression for work done on the shell element in terms of actual displacements [see equation (26)] |
| $C_k$ | matrix whose elements are coefficients in an expression for the strain energy of a shell element in terms of polynomial displacement functions [see equation (22)] |
| $C_{11}, C_{12}, C_{22}$ | membrane stiffness constants |
| $D_k$ | matrix whose elements are coefficients in an expression for work done on an element in terms of coefficients of polynomial displacement functions [see equation (30)] |
| $D_{11}, D_{12}, D_{22}$ | flexural stiffness constants |
| $e_1, e_2, e_{12}$ | middle-surface strains [see equations (2a) and (2b)] |
| $E$ | Young's modulus |
| $G_k$ | force matrix for element [see equation (32)] |
| $G$ | shell force matrix |
| $G_1, G_2$ | submatrices of G [see equation (34)] |
| $h$ | wall thickness |
| $k_b$ | wall elastic stiffness constant |
| $K$ | number of elements used to represent a shell |
| $K_{11}, K_{12}, K_{22}$ | stiffness constants representing interaction between in-plane and out-of-plane strains |

| Symbol | Definition |
|---|---|
| $n$ | circumferential wave number |
| $p$ | internal pressure |
| $r$ | radius of a shell measured in-plane normal to shell axis |
| $R_1, R_2$ | principal radii of curvature of shell |
| $R$ | matrix whose elements are coefficients in an expression for strain energy of the shell element in terms of actual variables in strain energy [see equation (19) and Appendix B] |
| $s$ | meridional coordinate |
| $S_k$ | element stiffness matrix |
| $S$ | shell stiffness matrix |
| $S_{11}, S_{12}, S_{21}, S_{22}$ | submatrices of S [see equation (34)] |
| $s_o$ | meridional distance from origin of s to reference edge of a shell |
| $s_k$ | meridional distance from reference edge of shell to center of kth element |
| $t$ | time, transpose of matrix |
| $T_k$ | inverse of matrix $A_k$ |
| $u$ | meridional component of middle-surface displacement |
| $V_k$ | strain energy of kth element |
| $V$ | strain energy of shell, volume inside shell segment |
| $w$ | normal component of middle-surface displacement |
| $W_k$ | work integral [see equation (25)] |
| $x$ | meridional coordinate measured within a single element (see Fig. 2) |
| $X$ | matrix which describes assumed form of variables appearing in strain energy [equation (11)] |
| $y_k$ | column matrix of element displacement and rotations [see equation (9)] |
| $y$ | column matrix containing unknown displacements and rotations |
| $y_1, y_2$ | submatrices of y [see equation (34)] |
| $Y$ | matrix which describes the assumed form of displacements u and w |

| Symbol | Definition |
|---|---|
| $\beta$ | rotation of shell generator relative to unstrained direction [see equation (12)] |
| $\gamma_k$ | column matrix whose elements are coefficients of assumed-displacement polynomials [see equation (10)] |
| $\Delta V$ | volume change under applied pressure |
| $\epsilon_k$ | meridional length of kth element |
| $\theta$ | cylindrical coordinate |
| $\kappa$ | fluid bulk modulus |
| $\kappa_1,\kappa_2,\kappa_{12}$ | changes in curvatures [see equations (2c) and (2d)] |
| $\mu$ | Poisson's ratio |
| $\xi_k$ | column matrix whose elements are displacements and rotations at ends of an element [see equation (15)] |

Primes denote differentiation with respect to s or x; superscript t denotes transpose of a matrix.

TECHNICAL MEMORANDUM

# PRESSURE-VOLUME PROPERTIES OF METALLIC BELLOWS

## I. INTRODUCTION

The purpose of this report is to develop a method of calculating the elastic stiffness constant, $k_b$, of a propellant line wall with complex geometry, such as a bellows section, within the linear range. It may be noted that $k_b$ has significance in both the static and dynamic sense similar to that of the spring constant, which appears in both the force-deflection and the frequency equations for a single-degree-of-freedom spring-mass system. Thus, while the bellows equations of this report are developed from a static point of view and a static experiment is used for verification, the end result is used to calculate the sonic velocity in a bellows section.

Metallic bellows are commonly used as segments of propellant feedlines for rocket-propelled vehicles to accommodate temperature-induced length variations, manufacturing tolerances, and gimbaling of the engines. These bellows sections deform radially and change volume when internal pressure varies, and the magnitude of such deformation is much higher than that for the straight, cylindrical segments of the line. The greater flexibility, or lesser stiffness, of the bellows decreases the frequency of acoustic oscillations in the line. These acoustic oscillations are a major factor in the so-called POGO phenomena which have plagued most of the larger liquid rocket-propelled vehicles for many years.

Dynamic phenomena of fluids flowing in lines involving both inertial and elastic effects are commonly called water hammer. The equations given by Paynter [1] for the axial fluid sonic velocity in a line can be combined into the form

$$a^2 = \frac{1/\rho}{\frac{1}{\kappa} + \frac{1}{A}\frac{\partial A}{\partial p}} \qquad (1)$$

or alternatively

$$a^2 = \frac{1/\rho}{\frac{1}{\kappa} + \frac{1}{k_b}} \qquad (1a)$$

where a is the sonic velocity, $\rho$ is the fluid density, $\kappa$ is the fluid bulk modulus, and $k_b$ is the wall elastic stiffness constant. Then $1/k_b$, the wall elastic flexibility, is

$$\frac{1}{A}\frac{\partial A}{\partial p} \quad .$$

Values of $1/k_b$ have been tabulated in Reference 1 for straight walls of various thicknesses. Equation (1a) is the equation for two springs in series.

For an incremental length,

$$\frac{1}{A}\frac{\partial A}{\partial p} = \frac{1}{V}\frac{\partial V}{\partial p}$$

where V is the volume, equation (1) can also be written

$$a^2 = \frac{1/\rho}{\frac{1}{\kappa} + \frac{1}{V}\frac{\partial V}{\partial p}} \qquad (1b)$$

By definition, $1/\kappa$ is the change in fluid volume per unit volume per unit change in pressure, and the second term in the denominator is the corresponding change in container volume.

A literature search of material dating back to 1950 (which included NASA and DOD computer searches and the Engineering Index) revealed few references to bellows elasticity. Earlier work probably does not exist since the problem is complex enough to require a digital computer for practical solution. Some studies of axial and bending stiffnesses of bellows segments have been made, but not a single reference to volumetric stiffness calculation has been found. Reference 2, a recent and extensive report on bellows analysis, gives simple formulae for axial and lateral spring constants and a comparison with experimental data. Methods for stress calculation are also given, but internal volume changes are not mentioned. References 3 and 4 constitute an extensive bibliography on fluid component technology with 54 references to bellows structures. Several concern axial or bending stiffness, but again, there is no reference to pressure-volume calculations or measurements. Much of the current work is being done in Japan and, unfortunately, has not been translated. Miyazono [5] has, for example, calculated the strains and axial force-deflection relationship for an unpressurized bellows. Daniels [6] describes a semi-empirical method of determining the modes of a bellows filled with liquid. The existence of the fluid column mode was not expected by this investigator until it was found in the experiment. Most current POGO analysts do not mention in their reports what approximations are used in the development of their line wall elasticity constants.

This study makes extensive use of a method developed by Adelman, Catherines, and Walton [7], who have developed a normal mode vibration analysis using a finite shell element of revolution with arbitrary meriodional curvature. The stiffness matrix derivation given is that explained in the reference, except that the provision for circumferential motion was removed ($n = 0$).

The major steps which are needed for the development of the static analysis were: the calculation of the nodal forces from the internal pressure, including provision for a more complex shell geometry; addition of matrix inversion for calculation of deflection; the inclusion of additional end conditions; and the calculation of volume change. An experimental verification was also made.

## II. ANALYTICAL METHOD

### A. Stiffness Matrix

The stiffness matrix derivation given follows closely that given by Adelman [7].

The structure to be analyzed may be taken as a thin shell of revolution with given meridional curvature (coordinates are shown in Fig. 1). The displacements in the meridional and normal directions are given by u and w, respectively, and $R_1$ and $R_2$ are the radii of curvature in the meridional and normal planes, respectively. The radius normal to the axis is denoted by r. All three radii are functions of the meridional coordinate, s. Derivatives with respect to s are denoted by primes.

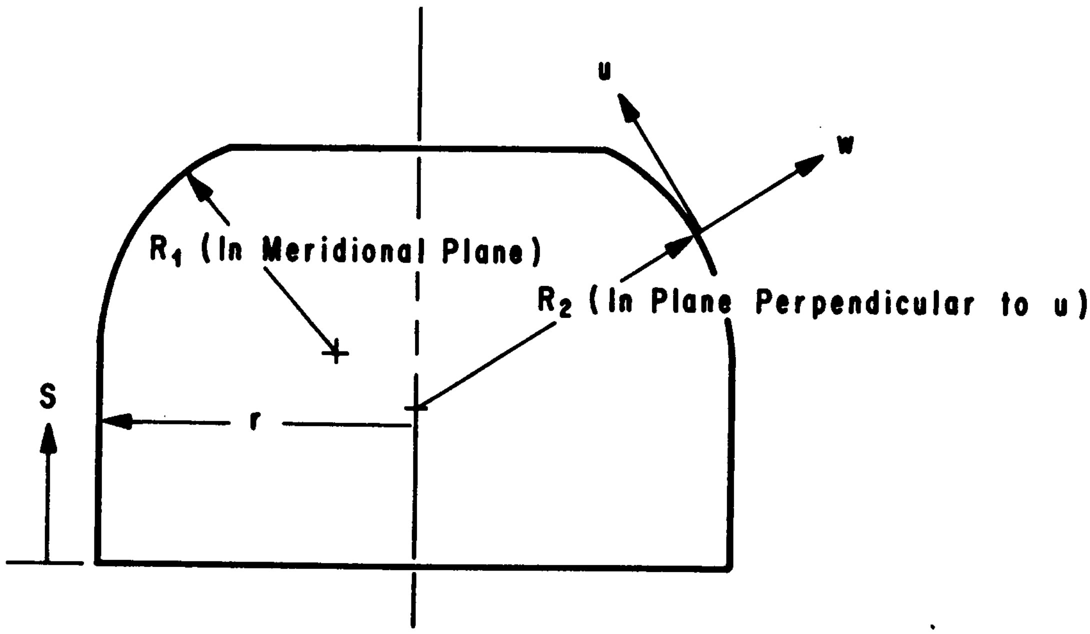


Figure 1. Shell geometry and coordinates.

The six strain displacement relations describing the local state of strain for a thin shell of revolution, as given by Novozhilov [8] and modified by the removal of all circumferential terms are:

Membrane strain in meridional direction:

$$e_1 = u' + \frac{w}{R_1} \quad . \tag{2a}$$

Membrane strain in circumferential direction:

$$e_2 = \frac{1}{r} r'u + \frac{w}{R_2} \quad . \tag{2b}$$

Change of curvature in meridional direction:

$$\kappa_1 = -w'' + \frac{1}{R_1} u' - \frac{1}{R_1^{\,2}} R_1'u \quad . \tag{2c}$$

Change of curvature in circumferential direction:

$$\kappa_2 = \frac{r'w'}{r} + \frac{1}{rR_1} r'u \quad . \tag{2d}$$

The plane shear strain $e_{12}$ and twist of the middle surface $\kappa_{12}$ are zero.

The strain energy for the shell is:

$$V = \pi \int (C_{11}e_1^{\,2} + 2C_{12}e_1e_2 + C_{22}e_2^{\,2})r\ ds + \pi \int (D_{11}\kappa_1^{\,2} + 2D_{12}\kappa_1\kappa_2 + D_{22}\kappa_2^{\,2})r\ ds$$

$$+ 2\pi \int [K_{11}e_1\kappa_1 + K_{12}(e_1\kappa_2 + e_2\kappa_1) + K_{22}e_2\kappa_2]r\ ds \quad , \tag{3}$$

where in equation (3) the integrations are taken over the shell surface, and the following definitions hold:

1) $C_{11}$, $C_{12}$, $C_{22}$ are membrane stiffnesses
2) $D_{11}$, $D_{12}$, $D_{22}$ are flexural stiffnesses
3) $K_{11}$, $K_{12}$, $K_{22}$ are stiffnesses due to the interaction between in-plane strains and changes in curvature.

All of these stiffnesses are, in general, functions of the meridional coordinate, s.

Substitution of the strains from equation (2) into the strain-energy expression of equation (3) yields the strain energy in terms of displacements. The amplitude of the strain energy is as follows:

$$V = \pi \int \left[ C_{11}\left(u' + \frac{w}{R_1}\right)^2 + 2C_{12}\left(u' + \frac{w}{R_1}\right)\left(\frac{r'}{r} u + \frac{w}{R_2}\right) + C_{22}\left(\frac{r'}{r} u + \frac{w}{R_2}\right)^2 \right] r\ ds$$

$$+ 2\pi \int \left[ K_{11}\left(u' + \frac{w}{R_1}\right)\left(-w'' + \frac{u'}{R_1} - \frac{R_1'}{R_1^{\,2}}\right) + K_{12}\left(u' + \frac{w}{R_1}\right) \cdot \left(-\frac{r'}{r} w' + \frac{r'}{rR_1} u\right) \right.$$

$$+ K_{12}\left(\frac{r'}{r} u + \frac{w}{R_2}\right) \cdot \left(-w'' + \frac{u'}{R_1} - \frac{R_1'}{R_1^{\,2}} w\right) + K_{22}\left(\frac{r'}{r} u + \frac{w}{R_2}\right) \tag{4}$$

(Continued)

$$\left(-\frac{r'}{r} w' + \frac{r'}{rR_1} u\right)\Big] r\, ds + \pi \int \left[D_{11} \left(-w'' + \frac{u'}{R_1} - \frac{R_1'}{R_1^2} u\right)^2\right.$$

$$\left. + 2D_{12} \left(-w'' + \frac{u'}{R_1} - \frac{R_1'}{R_1^2} u\right) \left(-\frac{r'}{r} w' + \frac{r'}{rR_1} u\right) + D_{22} \left(-\frac{r'}{r} w' + \frac{r'}{rR_1} u\right)^2\right] r\, ds \quad .$$

(4)
(Concluded)

The main steps of conventional finite-element analysis are followed by the present method. It is noted that each element coincides exactly with a slice of the actual shell.

A typical idealization of a shell of revolution is shown in Figure 2. Counting elements from the reference edge, the following definitions are made:

K = total number of elements

$\varepsilon_k$ = length of kth element, measured along meridian curve of shell

x = coordinate inside kth element, measured along meridian from center of kth interval so that

$$-\frac{\varepsilon_k}{2} \leq x \leq \frac{\varepsilon_k}{2} \quad . \tag{5}$$

$s_k$ = distance along meridian from reference edge of shell to center of the kth element.

From the foregoing definitions for x and $s_k$, it follows that

$$s = s_k + x \quad . \tag{6}$$

A numbering system has been adopted in which quantities such as displacement, derivatives of displacements, and rotations at $s = s_k - (\varepsilon_k/2)$ and $s = s_k + (\varepsilon_k/2)$ are indicated by subscripts k and k+1, respectively. Thus, for example, $w_k$ is the normal displacement at $s = s_k - (\varepsilon_k/2)$, and $u_{k+1}$ is the meridional displacement at $s = s_k + (\varepsilon_k/2)$. Also, it is necessary to have a notation for the radius of curvature $R_1$ at the locations $s = s_k + (\varepsilon_k/2)$. The symbols, $R_{1,k}$ and $R_{1,k+1}$ represent the respective values.

As an approximation, the displacements u and w are assumed to have the following polynomial forms [9] over the kth element:

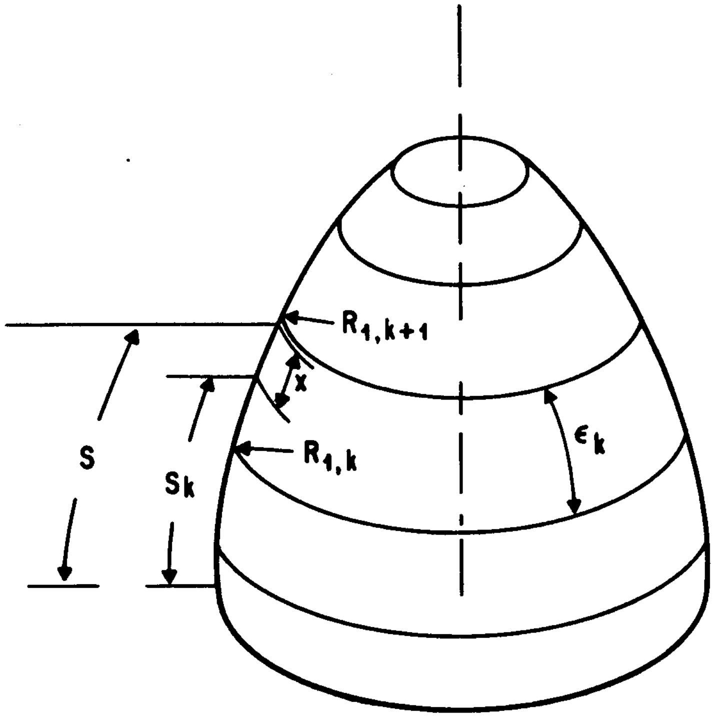


Figure 2. Typical idealization of shell of revolution.

$$w(x) = a_{o,k} + a_{1,k}x + a_{2,k}x^2 + a_{3,k}x^3 + a_{4,k}x^4 + a_{5,k}x^5$$

$$u(x) = b_{o,k} + b_{1,k}x + b_{2,k}x^2 + b_{3,k}x^3 \tag{7}$$

where the a's and b's are undetermined coefficients. From equation (7) it follows that

$$\{y_k\} = [X]\ \{\gamma_k\} \quad , \tag{8}$$

where

$$\{y_k\} \equiv (w\ w'\ w''\ u\ u')^t \quad , \tag{9}$$

$$\{\gamma_k\} = (a_{o,k}\ a_{1,k}\ a_{2,k}\ a_{3,k}\ a_{4,k}\ a_{5,k}\ b_{o,k}\ b_{1,k}\ b_{2,k}\ b_{3,k})^t \tag{10}$$

and

$$[X] = \begin{bmatrix} 1 & x & x^2 & x^3 & x^4 & x^5 & 0 & 0 & 0 & 0 \\ 0 & 1 & 2x & 3x^2 & 4x^3 & 5x^4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 6 & 12x^2 & 20x^3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & x & x^2 & x^3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 2x & 3x^2 \end{bmatrix} \quad . \tag{11}$$

The rotation of the meridian curve relative to the unstrained direction is defined as $\beta$ and is given by

$$\beta = w' - \frac{u}{R_1} \quad . \tag{12}$$

It follows that

$$\beta_k = w_k' - \frac{u_k}{R_{1,k}} \tag{13}$$

and

$$\beta_{k+1} = w_{k+1}' - \frac{u_{k+1}}{R_{1,k+1}} \quad . \tag{14}$$

The quantity $\beta'$ may now be defined as the meridional derivative of the meridional rotation; i.e., $\beta' = \partial\beta/\partial s$. Now a vector containing the end deflections of an element may be defined so that

$$\{\xi_k\} = w_k \; u_k \; \beta_k \; u_k' \; \beta_k' \; w_{k+1} \; u_{k+1} \; \beta_{k+1} \; u_{k+1}' \; \beta_{k+1}')^t \tag{15}$$

where the subscripts k and k+1 refer to the displacements at $x = -\varepsilon_k/2$ and $x = \varepsilon_k/2$, respectively.

Inserting $x = -\varepsilon_k/2$ and $x = \varepsilon_k/2$ into the appropriate locations in equation (8) results in the following relationship:

$$\{\xi_k\} = [A_k] \; \{\gamma_k\} \quad , \tag{16}$$

where the matrix $[A_k]$ is given by equation (A-1) of Appendix A. When equation (16) is inverted, the following relationship results:

$$\{\gamma_k\} = [T_k]\ \{\xi_k\} \quad , \tag{17}$$

where

$$[T_k] = [A_k]^{-1} \quad . \tag{18}$$

The inverse matrix $[T_k]$ is given by equation (A-2) of Appendix A.

From equation (4) the strain energy of an element may be written as follows:

$$V_k = \frac{\pi}{2} \int_{-\varepsilon_k/2}^{\varepsilon_k/2} \{y_k\}^t\ [R]\ \{y_k\}\ dx \tag{19}$$

where [R] is a 5 x 5 symmetric matrix, the elements of which are known functions of the meridional coordinate x. The elements of [R] are listed in Appendix B. Using equation (8) in equation (19) permits the strain energy to be written in terms of the undetermined polynomial coefficients as follows:

$$V_k = \frac{\pi}{2} \int_{-\varepsilon_k/2}^{\varepsilon_k/2} \{\gamma_k\}^t\ [X]^t\ [R]\ [X]\ \{\gamma_k\}\ dx \tag{20}$$

or

$$V_k = \frac{1}{2}\ \{\gamma_k\}^t\ [C_k]\ \{\gamma_k\} \quad , \tag{21}$$

where

$$[C_k] = \pi \int_{-\varepsilon_k/2}^{\varepsilon_k/2} [X]^t\ [R]\ [X]\ dx \quad . \tag{22}$$

Finally, use of the transformation expressed by equation (17) gives the strain energy as

$$V_k = \frac{1}{2} \{\xi_k\}^t [T_k]^t [C_k] [T_k] \{\xi_k\} \quad . \tag{23}$$

Inspection of equation (23) identifies the shell element stiffness matrix $[S_k]$ as

$$[S_k] = [T_k]^t [C_k] [T_k] \quad . \tag{24}$$

The type of bellows being considered is made from a single piece of metal. All radii and their first derivatives, the parameters which describe the shell geometry, are continuous within each segment.

## B. Force Matrix

The work done by the internal pressure, p, on an element may be defined as

$$W_k = \pi \int_{-\varepsilon_k/2}^{\varepsilon_k/2} [B_k] \left\{ \begin{matrix} w \\ u \end{matrix} \right\} dx \quad , \tag{25}$$

where

$$[B_k] = [p.r(x) \quad 0 \;] \quad . \tag{26}$$

Here the u displacement has been included to permit later studies for axial loads.

Based on the assumed displacements of equation (7), the following relation may be written:

$$\left\{ \begin{matrix} w \\ u \end{matrix} \right\} = [Y] \{\gamma_k\} \quad , \tag{27}$$

where

$$[Y] = \begin{bmatrix} 1 & x & x^2 & x^3 & x^4 & x^5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & x & x^2 & x^3 \end{bmatrix} \quad . \tag{28}$$

Substituting equation (27) into equation (25) yields

$$W_k = [D_k] \{\gamma_k\} \quad , \tag{29}$$

where

$$D_k = \pi \int_{-\epsilon_k/2}^{\epsilon_k/2} [B_k][Y] \, dx \quad . \tag{30}$$

Further substitution of equations (17) and (29) gives

$$W_k = [D_k][T_k]\{\xi_k\} \quad . \tag{31}$$

The force matrix, G, then is

$$[G_k] = [D_k][T_k] \quad . \tag{32}$$

## C. Assembly and Solution of Equations

The stiffness matrix $[S_k]$ and the force matrix $[G_k]$ for an element have now been computed. Using the direct stiffness method, the stiffness, forces, and displacements of all the elements are combined into a total stiffness matrix [S], a force matrix [G], and a displacement matrix {y}. The resulting equation is

$$[S]\{y\} = \{G\}. \tag{33}$$

This is the equation for the unrestrained shell. Rigid edge constraints are incorporated by deleting from the stiffness matrix of equation (33) those rows and columns which correspond to displacements and rotations that must vanish to satisfy the constraints, and deleting the same rows only from the force matrix. This may be demonstrated by partitioning the matrices of equation (33) in the following manner:

$$\left[\begin{array}{c|c} S_{11} & S_{12} \\ \hline S_{21} & S_{22} \end{array}\right] \left\{\begin{array}{c} y_1 \\ \hline y_2 \end{array}\right\} = \left\{\begin{array}{c} G_1 \\ \hline G_2 \end{array}\right\} \tag{34}$$

so that $y_1$ contains all the unrestrained coordinates of the structure and $y_2$ is null. Then equation (34) can be separated into two equations:

$$[S_{11}]\{y_1\} + [S_{12}]\{y_2\} = \{G_1\} \tag{35a}$$

$$[S_{21}]\{y_1\} + [S_{22}]\{y_2\} = \{G_2\} \quad . \tag{35b}$$

Equation (35a) is of interest because all quantities except $y_1$ are known. Eliminating the zero terms gives

$$[S_{11}]\{y_1\} = \{G_1\} \quad . \tag{36}$$

Since the form of equations (33) and (36) is identical and both the free and fixed conditions may be of interest, the notation of equation (33) will be used hereafter, but the fixity conditions will be applied as required.

The stiffness matrix is a banded matrix. The solution of equation (33) was obtained using a standard band matrix solution routine.

## D. Volume Integral

The solution vector {y} gives the displacements and slopes at the nodes, the points where the elements meet. To obtain the volume change due to the applied pressure, these nodal displacements must be transformed to find w as a function of x, and then integrated. This can be done considering one segment at a time. The portion of the {y} vector applying to one segment is $\{\xi_k\}$. Substituting equation (17) into (8) gives

$$\{y_k\} = [X][T_k]\{\xi_k\} \quad . \tag{37}$$

The change of volume then is

$$\Delta V = 2\pi \int_{-\varepsilon_k/2}^{\varepsilon_k/2} w(x)r(x)\, dx \quad . \tag{38}$$

The numerical integration is performed using 100 stations and the trapezoidal rule.

## E. Geometry of Typical Bellows Elements

Five parameters describing the radius as a function of the meridional coordinate are required for the calculations:

- $r(x)$ shell radius in-plane perpendicular to axis
- $r'(x)$ derivative of $r(x)$ with respect to x
- $R_1(x)$ shell radius in meridional plane
- $R_1'(x)$ derivative of $R_1(x)$ with respect to x

$R_2(x)$ shell radius in-plane perpendicular to both meridional and tangential planes.

The four types of shell segment which occur for the bellows are cylinder, cone, and the internal and external constant radii. These are shown in Figure 3 along with the coordinate system and nomenclature.

For the cylindrical segment:

$$r(x) = r \text{ (a constant)} \tag{39a}$$

$$r'(x) = 0 \tag{39b}$$

$$R_1(s) = \infty \text{ } (1/R_1 \text{ is used as computer program variable}) \tag{39c}$$

$$R_1'(x) = 0 \tag{39d}$$

$$R_2(x) = r \quad . \tag{39e}$$

For the conical segment:

$$r(x) = r(-\varepsilon_k/2) + x \sin\theta \tag{40a}$$

$$r'(x) = \sin\theta \tag{40b}$$

$$R_1(x) = \infty \tag{40c}$$

$$R_1'(x) = 0 \tag{40d}$$

$$R_2(x) = r(x)/\cos\theta \quad . \tag{40e}$$

For the internal radius segment:

$$r(x) = h - R\cos x/R \tag{41a}$$

$$r'(x) = \sin x/R \tag{41b}$$

$$R_1(x) = -R \tag{41c}$$

$$R_1'(x) = 0 \tag{41d}$$

$$R_2(x) = r(x)/\cos x/R \quad . \tag{41e}$$

For the external radius element:

$$r(x) = h + R\cos x/R \tag{42a}$$

$$r'(x) = -\sin x/R \tag{42b}$$

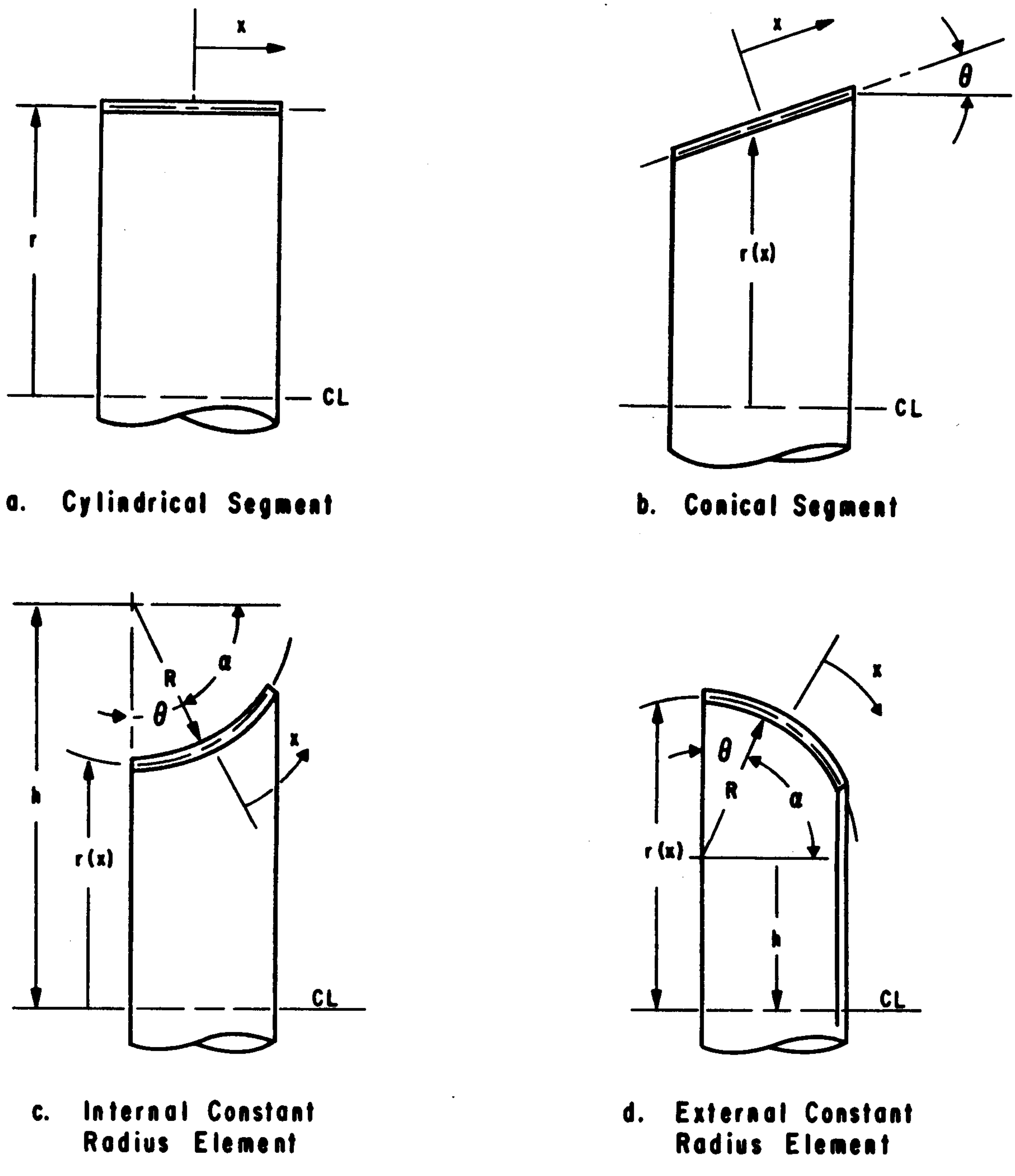


Figure 3. Shell elements of bellows.

$$R_1(x) = R \qquad (42c)$$

$$R_1'(x) = 0 \qquad (42d)$$

$$R_2(x) = r(x)/\cos x/R \quad . \qquad (42e)$$

## F. Example Problem

The bellows was obtained from the Marshall Space Flight Center Test Division to be used for experimental verification of the analytical calculations. This bellows, after removal of the cover and liner, is shown in Figure 4. The bellows was manufactured by Flexicraft Industries, Chicago, Illinois, who furnished the blueprint upon request. It is a nominal 4-in. (ID) bellows intended for long term, low stress service in a cryogenic test facility. The material is 0.037 in. Type 304 stainless steel. Since the radii of the corrugations were not dimensioned in the blueprint, these were measured with a radius gauge and found to be:

Outer corrugation - 11/32 OD

Inner corrugation - 9/32 OD

End radius - 0.780 ID.

The distance across four whole corrugations was measured to be 5-9/32 in. A clearance of about 0.002 in. was measured between the bellows stock and the flange, so the 0.037 in. thickness was used from the corrugation to weld in the calculations.

From the given and measured dimensions, the geometry of the shell middle surface was constructed. The geometry of the center and end corrugations is given in Figure 5, and the results of the initial modeling attempt are shown in Figure 6.

The bellows is formed by expanding the tube stock to form the corrugations. Kervick [10] predicts thinning of the wall proportional to radius for this method of forming, so this was assumed.

## G. Computer Program

The computer program furnished by Adelman [9] was modified to accept a static case by inserting the following changes:

1) Provision for symmetrical half-end conditions. The shell is constrained to zero motion in u and $\beta$ at the symmetry plane and w, u, and $\beta$ at the clamped end.

2) Provision for "floating radial" end conditions with u and $\beta$ fixed at each end.

3) Force matrix generated.

4) Band matrix solution routine added.

5) Deflection introduced into mode shape routine and print changes made to identify it.

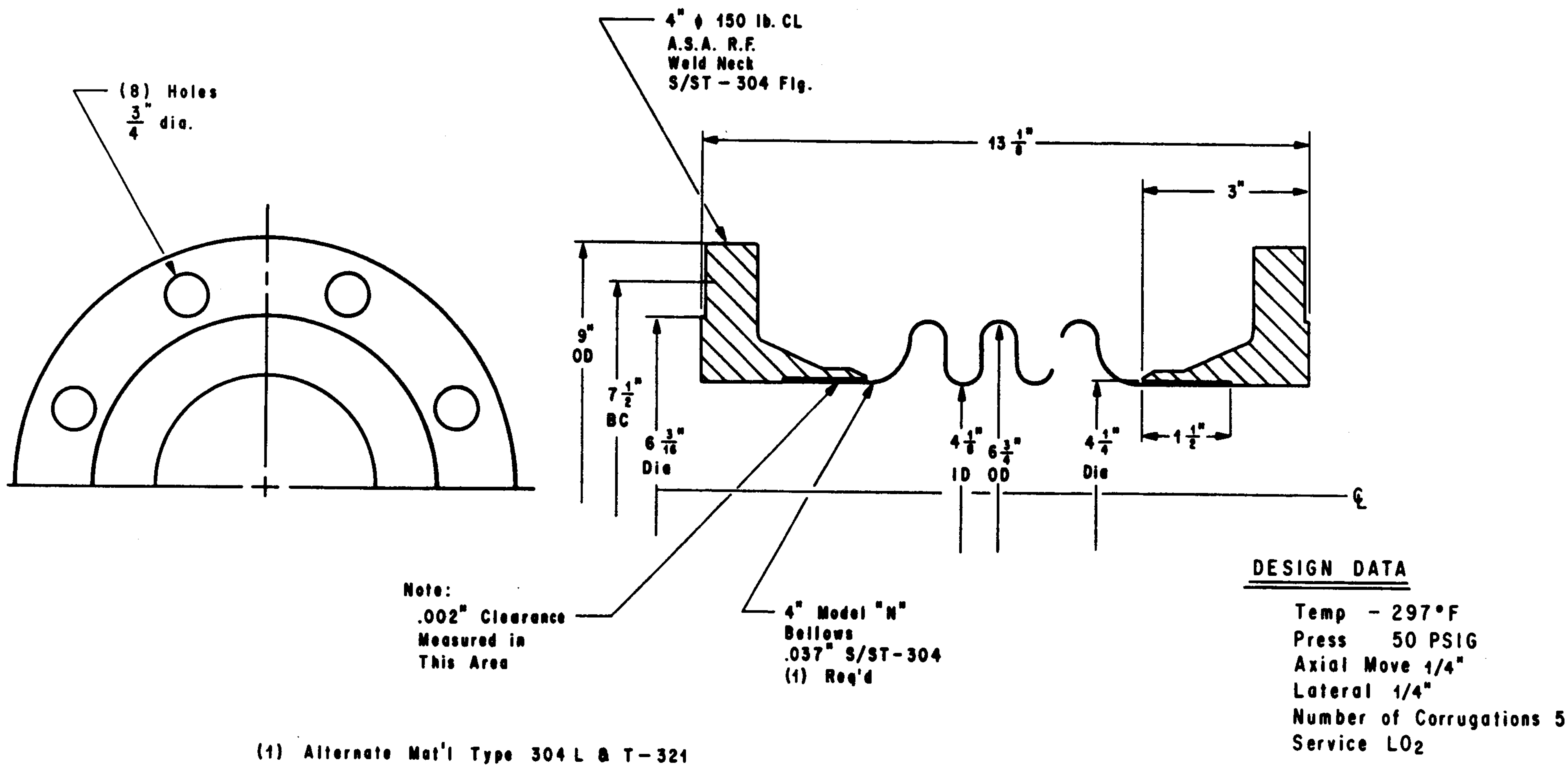


Figure 4. Facility bellows.

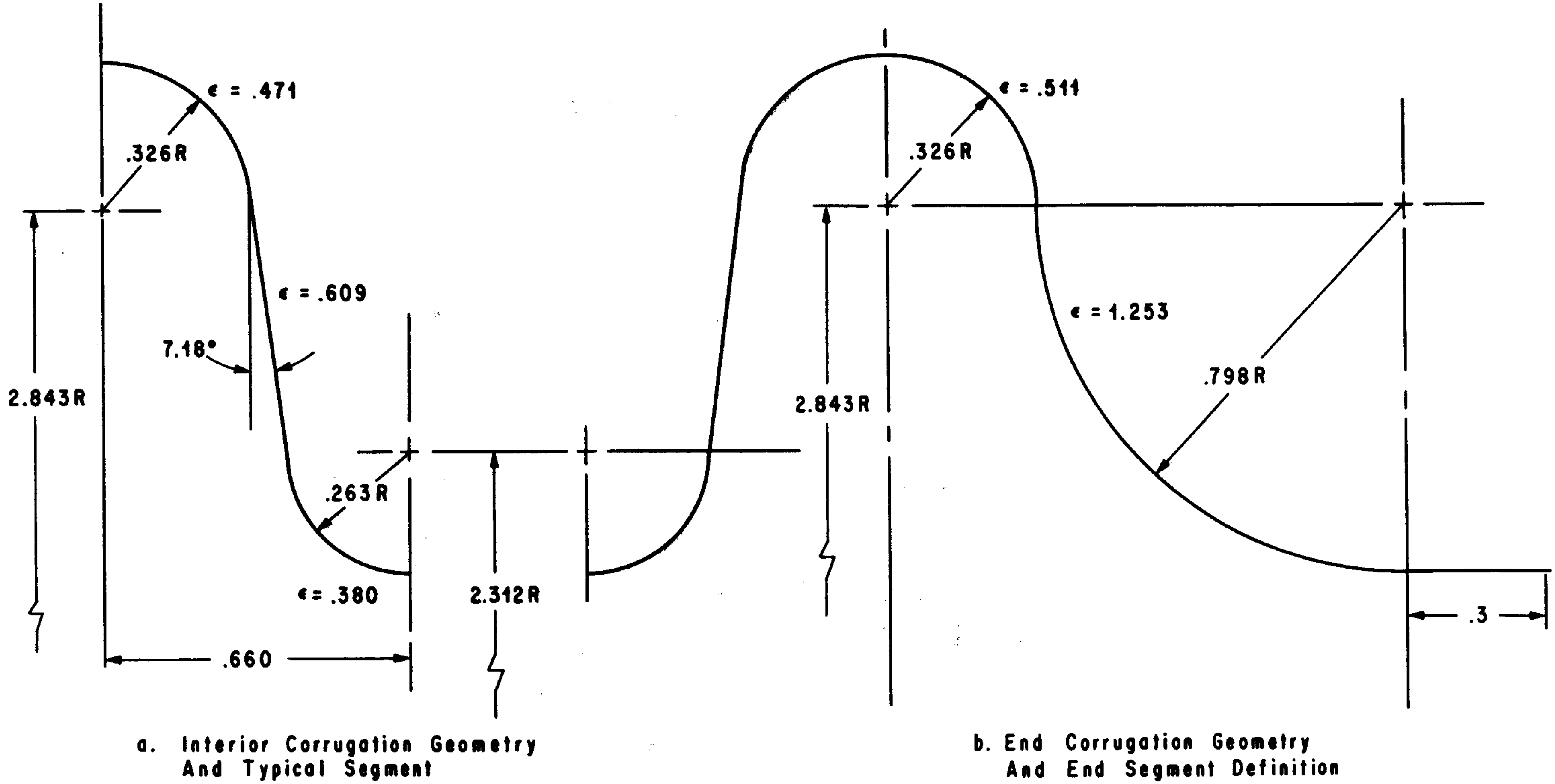


Figure 5. Corrugation geometry.

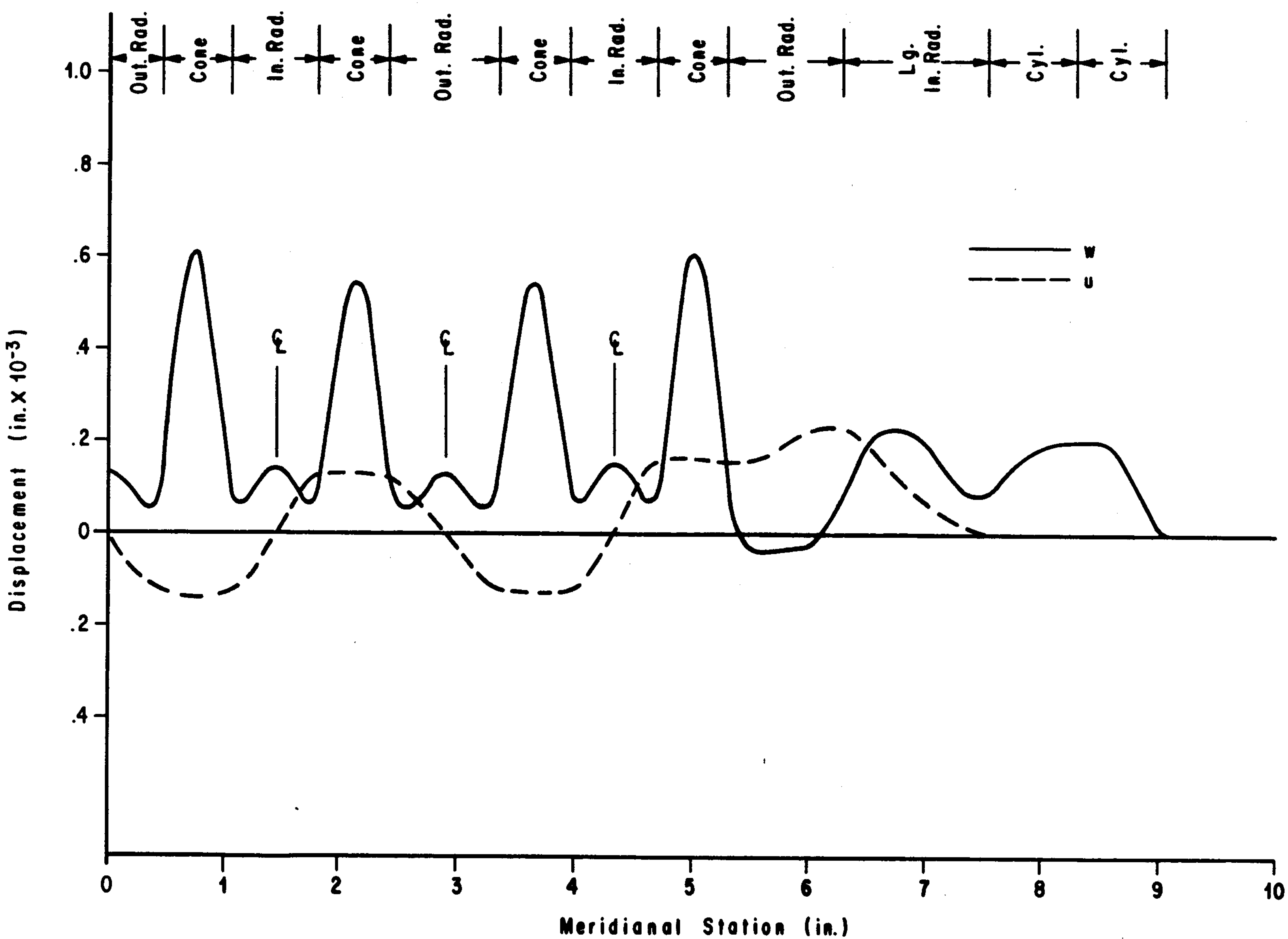


Figure 6. Displacements of symmetric - half bellows.

6) Volume change calculated and printed.

7) Geometry defined for each segment rather than total shell.

8) Subroutine for ring effects and plotting were removed.

9) Circumferential variation removed.

A list of the subroutines and a description of their primary functions are given in Appendix C, while a complete listing with a sample output is given in Appendix D.

## III. EXPERIMENT

A hydrostatic test was run to verify the results of the analysis. The apparatus, shown schematically in Figure 7, was set up in the University of Alabama in Huntsville shock tube laboratory where high pressure air and vacuum sources were available. First, the ends of the bellows were fixed relative to each other and heavy closure flanges attached to each end by eight 3/4-in. threaded rods. A chemical pipette, graduated in milliliters, was used as a sight gauge. It was bonded at its bottom end into a hole in the top closure flange and at its top end into a block supported by two of the extended threaded rods. Three valves permitted the introduction of either air pressure, vacuum, or water into the interior of the bellows by way of the pipette. Furthermore, the water was restricted to flow only into the bellows by gravity.

The vacuum was used to remove any entrapped air bubbles from the system and also to draw small amounts of water into the system so that the level at zero pressure (gauge) was slightly below the top of the sight gauge. Only one valve would normally be open at any time. A pressure regulator was used to reduce the source pressure to the exact values needed.

Data from the experiment are tabulated in Table 1 and plotted in Figure 8. Points were taken during both the initial pressure build-up and release and a slight hysteresis loop was formed. Subsequent cycles lay on the upper curve. The data is exhibiting some nonlinearity above 20 $lb/in.^2$, so a tangent was drawn to provide the low-level, linear characteristics compatible with the theory. The volume change from the graph then is 2.96 ml (0.181 $in.^3$) per 50 $lb/in.^2$.

No accurate measurement of the deflections appeared to be practical. A check with a dial indicator produced no deflections of more than 0.001 in. at any point in any direction.

Since the test apparatus is not truly rigid, three corrections must be made to the raw data, one experimental and the other two analytical.

The effect of the end flanges and gaskets was determined experimentally by removing the bellows and bolting the two closure flanges directly together. Application of 60-$lb/in.^2$ pressure produced 0.3 ml volume change. This is equal to 0.25 ml (0.015 $in.^3$) per 50 psi rated load.

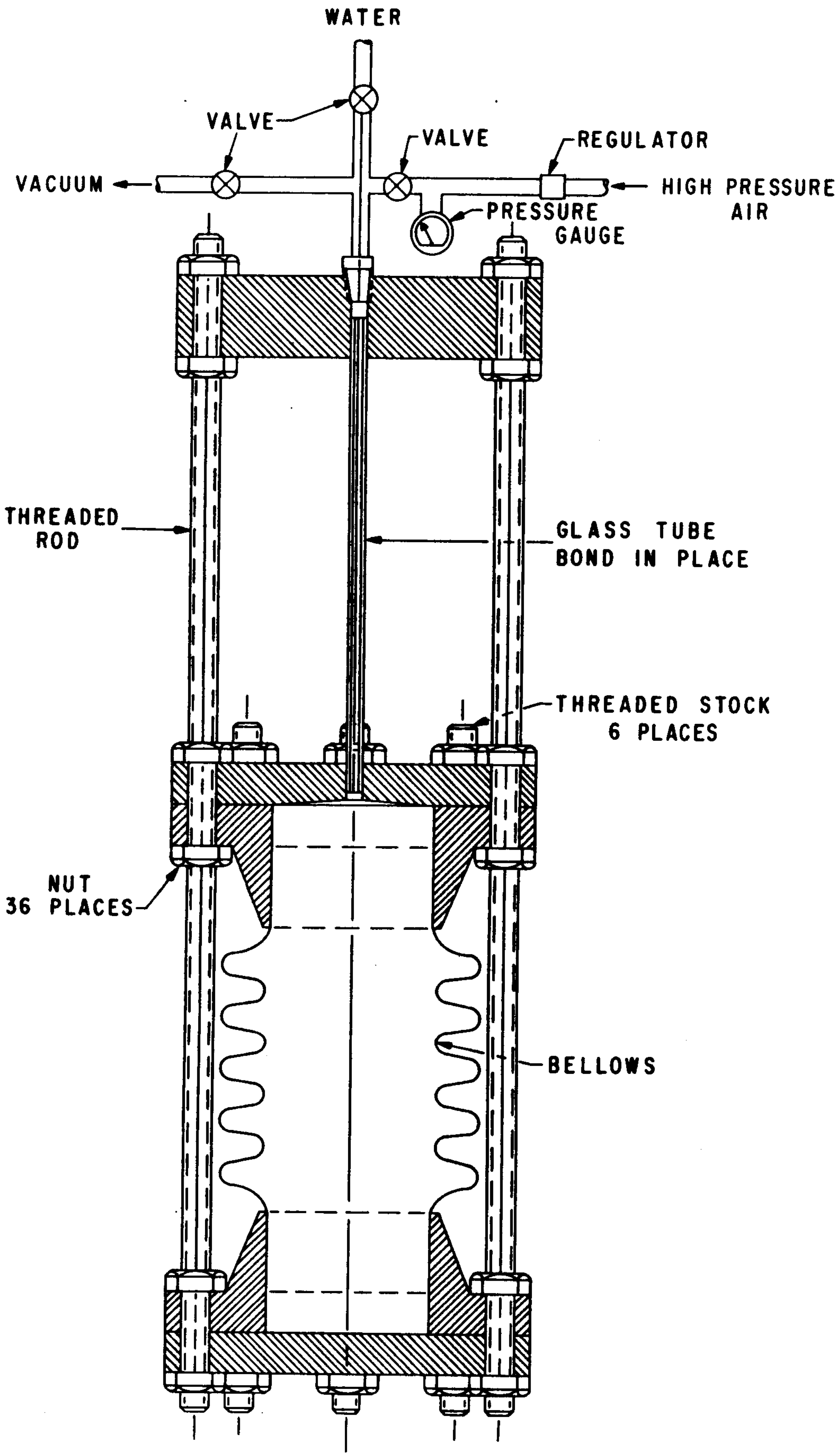


Figure 7. Bellows and equipment schematic.

TABLE 1. DATA FROM EXPERIMENT

| Pressure (psig) | Level (ml) |
|---|---|
| 0 | 0.2 |
| 9.5 | 0.85 |
| 20.0 | 1.5 |
| 32.0 | 2.05 |
| 40.0 | 2.45 |
| 52.2 | 3.0 |
| 40.0 | 2.50 |
| 30.0 | 2.00 |
| 20.0 | 1.55 |
| 10.0 | 0.95 |
| 0 | 0.35 |

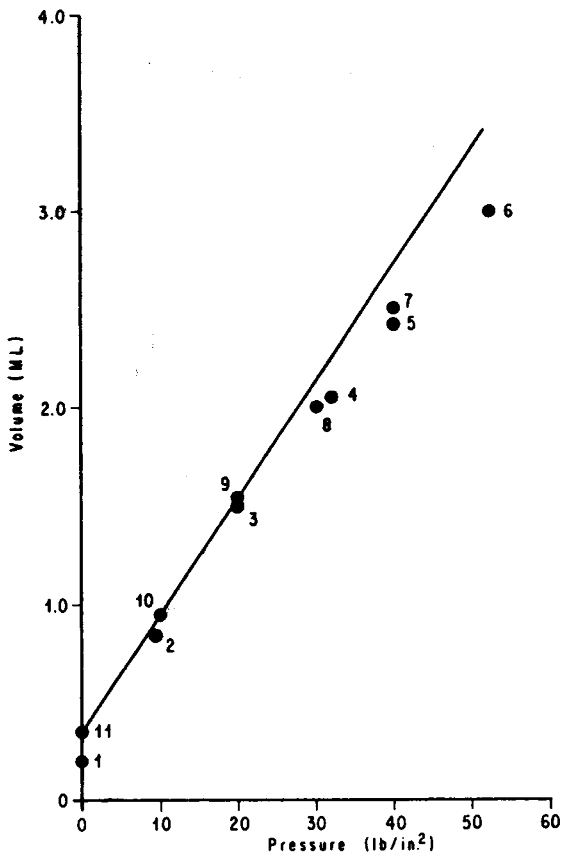


Figure 8. Experimental data.

The internal pressure causes a length change in the threaded rods used to restrain the bellows. Assuming that the bellows carries no axial load and that the rod effective area is the mean cross-sectional area, the length change is $2.34 \times 10^{-4}$ in. Further assuming that the effective area of the bellows is the mean cross-sectional area in the convolutions, the net volume change is 0.0052 in.$^3$.

The bellows internal volume was calculated to be 309.8 in.$^3$ by numerical integration. The volume change due to liquid compression under 50 lb/in.$^2$ pressure is 0.0515 in.$^3$.

## IV. COMPARISON AND CONCLUSIONS

The summary results of the experiment and the analysis are listed below:

**Experiment**

| | |
|---|---|
| Measured volume change | 0.1810 in.$^3$ |
| Measured tare | 0.0153 |
| Calculated effect of length change | 0.0052 |
| Net change in bellows volume | 0.1605 |

**Theory**

| | |
|---|---|
| Symmetric half | 0.0440 |
| Total bellows | 0.0880 |
| Liquid compressibility | 0.0515 |
| Total predicted volume change | 0.1395 |

**Error**

$$100 \times \frac{0.1605 - 0.1395}{0.1605} = 13.1 \text{ percent} .$$

An error of this magnitude, since it does not strictly represent a difference between theory and experiment because several errors are possible in intermediate steps, indicates that the method is probably accurate enough for many applications. It might be desirable to obtain cross sections of the formed convolutions to measure thickness also, since the stiffness terms $D_{11}$, $D_{12}$, and $D_{22}$ are proportional to the thickness cubed. The error in velocity will be only half the error in stiffness.

The axial sonic velocity for water within a line composed of typical segments of the example bellows can be calculated approximating equation (1a) and using data from the previous page. Values are $\rho = 0.935 \times 10^{-4}$ lb sec$^2$/in.$^4$, $\kappa = 0.294 \times 10^6$

lb/in.$^2$, V = 26.36 in.$^3$, $\Delta V$ = 0.0079 in.$^3$, and $\Delta p$ = 50 lb/in.$^2$. This gives a velocity in the bellows of 33,760 in./sec compared to a velocity of 56,080 in./sec in rigid line.

## REFERENCES


1. Paynter, H.M.: Section 20, Fluid Transients in Engineering Systems. From Handbook of Fluid Dynamics, by V.L. Streeter, Ed., McGraw-Hill Book Co., Inc., New York, 1961.

2. Trainer, T.M., Hulbert, L.E., Lestingi, J.F., Keith, R.E., et al.: Final Report on the Development of Analytical Techniques for Bellows and Diaphragm Design. Report AFRPL-TR-68-22, March 1968.

3. Howell, G.A.: A Bibliography of Aerospace Valve and Fluid Component Technology. TRW Systems, Redondo Beach, California, December 1965.

4. Weathers, T.M.: Addendum A to a Bibliography of Aerospace Valve and Fluid Component Technology. TRW Systems, Redondo Beach, California, May 1967.

5. Miyazono, S.: Experimental and Theoretical Studies of the Expansion Joints (Bellows) in the Piping Systems of Nuclear Reactors. Japan Atomic Energy Research Institute, report No. 1130, April 1967.

6. Daniels, V.R.: Dynamic Aspects of Metal Bellows. U.S. Naval Research Laboratory Shock and Vibration Bulletin 35, Part 3, January 1966, pp. 107-124.

7. Adelman, H.M., Catherines, D.S., and Walton, W.C.: A Method for Computation of Vibration Modes and Frequencies of Orthotropic Thin Shells of Revolution Having General Meridional Curvature. NASA TN D-4972, January 1969.

8. Novozhilov, V.P. (P.G. Lowe, translation): Thin Shell Theory. Second Ed., P. Noordhoff Ltd., Groningen, Netherlands, C. 1964.

9. Adelman, Howard M., Catherines, Donnell S., Steeves, Earl C., and Walton, William C., Jr.: User's Manual for a Digital Computer Program for Computing the Vibration Characteristics of Ring-Stiffened Orthotropic Shells of Revolution. NASA TM X-2138, December 1970.

10. Kervick, Richard, Jr.: Cold Bending and Forming Tube and Other Sections. American Society of Tool and Manufacturing Engineers, Dearborn, Michigan, C. 1966.

# APPENDIX A

## ELEMENTS OF MATRICES $A_k$ AND $T_k$

PRECEDING PAGE BLANK NOT FILMED



PAGE 24 INTENTIONALLY BLANK

$$
[A_k] = \begin{bmatrix}
1 & -\varepsilon/2 & \varepsilon^2/4 & -\varepsilon^3/8 & \varepsilon^4/16 & -\varepsilon^5/32 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & -\varepsilon/2 & \varepsilon^2/4 & -\varepsilon^3/8 \\
0 & 1 & \varepsilon & 3\varepsilon^2/4 & \varepsilon^3/2 & 5\varepsilon^4/16 & -1/R_{1,k} & -\varepsilon/2R_{1,k} & -\varepsilon^2/4R_{1,k} & \varepsilon^3/8R_{1,k} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -\varepsilon & 3\varepsilon^2/4 \\
0 & 0 & 2 & -3\varepsilon & 3\varepsilon^2 & -5\varepsilon^3/2 & R'_{1,k}/R^2_{1,k} & a1_k & a2_k & a3_k \\
1 & \varepsilon/2 & \varepsilon^2/4 & \varepsilon^3/8 & \varepsilon^3/16 & \varepsilon^5/32 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & \varepsilon/2 & \varepsilon^2/4 & \varepsilon^3/8 \\
0 & 1 & \varepsilon & 3\varepsilon^2/4 & \varepsilon^3/2 & 5\varepsilon^4/16 & -1/R_{1,k+1} & -\varepsilon/2R_{1,k+1} & -\varepsilon^2/4R_{1,k+1} & -\varepsilon^3/8R_{1,k+1} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \varepsilon & 3\varepsilon^2/4 \\
0 & 0 & 2 & 3\varepsilon & 3\varepsilon^2 & 5\varepsilon^3/2 & R'_{1,k+1}/R^2_{1,k+1} & a1_{k+1} & a2_{k+1} & a3_{k+1}
\end{bmatrix}
$$

where

$$a1_k = -\frac{1}{R_{1,k}}\left(\frac{\varepsilon R'_{1,k}}{2R_{1,k}} + 1\right), \quad a2_k = \frac{\varepsilon}{R_{1,k}}\left(\frac{\varepsilon R'_{1,k}}{4R_{1,k}} + 1\right), \quad a3_k = -\frac{\varepsilon^2}{4R_{1,k}}\left(\frac{\varepsilon R'_{1,k}}{2R_{1,k}} + 3\right),$$

$$a1_{k+1} = +\frac{1}{R_{1,k+1}}\left(\frac{\varepsilon R'_{1,k+1}}{2R_{1,k}} - 1\right), \quad a2_k = \frac{\varepsilon}{R_{1,k+1}}\left(\frac{\varepsilon R'_{1,k+1}}{4R_{1,k+1}} - 1\right),$$

$$a3_k = +\frac{\varepsilon^2}{4R_{1,k+1}}\left(\frac{\varepsilon R'_{1,k+1}}{2R_{1,k+1}} - 3\right)$$

$$
[T_k] = \begin{bmatrix}
\frac{1}{2} & \frac{\varepsilon(5 - t_k)}{32R_{1,k}} & \frac{5\varepsilon}{32} & \frac{\varepsilon^2}{64R_{1,k}} & \frac{\varepsilon^2}{64} & \frac{1}{2} & -\frac{\varepsilon(5 + t_{k+1})}{32R_{1,k+1}} & -\frac{5\varepsilon}{32} & \frac{\varepsilon^2}{64R_{1,k+1}} & \frac{\varepsilon^2}{64} \\
-\frac{15}{8\varepsilon} & \frac{(-7 + t_k)}{16R_{1,k}} & -\frac{7}{16} & -\frac{\varepsilon}{32R_{1,k}} & -\frac{\varepsilon}{32} & \frac{15}{8\varepsilon} & -\frac{(7 + t_{k+1})}{16R_{1,k+1}} & -\frac{7}{16} & \frac{\varepsilon}{32R_{1,k+1}} & \frac{\varepsilon}{32} \\
0 & \frac{(-3 + t_k)}{4\varepsilon R_{1,k}} & -\frac{3}{4\varepsilon} & -\frac{1}{8R_{1,k}} & -\frac{1}{8} & 0 & \frac{(3 + t_{k+1})}{4\varepsilon R_{1,k+1}} & \frac{3}{4\varepsilon} & -\frac{1}{8R_{1,k+1}} & -\frac{1}{8} \\
\frac{5}{\varepsilon^3} & \frac{(5 - t_k)}{2\varepsilon^2 R_{1,k}} & \frac{5}{2\varepsilon^2} & \frac{1}{4\varepsilon R_{1,k}} & \frac{1}{4\varepsilon} & -\frac{5}{\varepsilon^3} & \frac{(5 + t_{k+1})}{2\varepsilon^2 R_{1,k+1}} & \frac{5}{2\varepsilon^2} & -\frac{1}{4\varepsilon R_{1,k+1}} & -\frac{1}{4\varepsilon} \\
0 & \frac{(1 - t_k)}{2\varepsilon^3 R_{1,k}} & \frac{1}{2\varepsilon^3} & \frac{1}{4\varepsilon^2 R_{1,k}} & \frac{1}{4\varepsilon^2} & 0 & -\frac{(1 + t_{k+1})}{2\varepsilon^3 R_{1,k+1}} & -\frac{1}{2\varepsilon^3} & \frac{1}{4\varepsilon^2 R_{1,k+1}} & +\frac{1}{4\varepsilon^2} \\
-\frac{6}{\varepsilon^5} & \frac{(-3 + t_k)}{\varepsilon^4 R_{1,k}} & -\frac{3}{\varepsilon^4} & -\frac{1}{2\varepsilon^3 R_{1,k}} & -\frac{1}{2\varepsilon^3} & \frac{6}{\varepsilon^5} & -\frac{(3 + t_{k+1})}{\varepsilon^4 R_{1,k+1}} & -\frac{3}{\varepsilon^4} & \frac{1}{2\varepsilon^3 R_{1,k+1}} & \frac{1}{2\varepsilon^2} \\
0 & \frac{1}{2} & 0 & \frac{\varepsilon}{8} & 0 & 0 & \frac{1}{2} & 0 & -\frac{\varepsilon}{8} & 0 \\
0 & -\frac{3}{2\varepsilon} & 0 & -\frac{1}{4} & 0 & 0 & \frac{3}{2\varepsilon} & 0 & -\frac{1}{4} & 0 \\
0 & 0 & 0 & -\frac{1}{2\varepsilon} & 0 & 0 & 0 & 0 & \frac{1}{2\varepsilon} & 0 \\
0 & \frac{2}{\varepsilon^3} & 0 & \frac{1}{\varepsilon^2} & 0 & 0 & -\frac{2}{\varepsilon^3} & 0 & \frac{1}{\varepsilon^2} & 0
\end{bmatrix}
$$

where

$$
t_k = \frac{\varepsilon R'_{1,k}}{2R_{1,k}} \quad , \quad t_{k+1} = \frac{\varepsilon R'_{1,k+1}}{2R_{1,k+1}}
$$

## APPENDIX B

## ELEMENTS OF MATRIX [R]

[See equation (19)]

The elements of matrix [r] are as follows:

$$R_{11} = \frac{C_{11}r}{R_1^{\,2}} + 2\,\frac{C_{12}r}{R_1R_2} + \frac{C_{22}r}{R_2^{\,2}}$$

$$R_{12} = R_{21} = -\frac{K_{12}r'}{R_1} - \frac{K_{22}r'}{R_2}$$

$$R_{13} = R_{31} = -\frac{K_{11}r}{R_1} - \frac{K_{12}r}{R_2}$$

$$R_{14} = R_{41} = \frac{C_{12}r'}{R_1} + \frac{C_{22}r'}{R_2} - \frac{K_{11}R_1'r}{R_1^{\,3}} + \frac{K_{12}r'}{R_1^{\,2}} - \frac{K_{12}rR_1'}{R_1^{\,2}\,R_2} + \frac{K_{22}r'}{R_1R_2}$$

$$R_{15} = R_{51} = \frac{C_{11}r}{R_1} + \frac{C_{12}r}{R_2} + \frac{K_{11}r}{R_1^{\,2}} + \frac{K_{12}r}{R_1R_2}$$

$$R_{22} = \frac{D_{22}(r')^2}{r}$$

$$R_{23} = R_{32} + D_{12}r'$$

$$R_{24} = R_{42} = \frac{D_{12}R_1'r'}{R_1^{\,2}} - \frac{D_{22}(r')^2}{rR_1} - \frac{K_{22}(r')^2}{r}$$

$$R_{25} = R_{52} = -\frac{D_{12}r'}{R_1} - K_{12}r'$$

$$R_{33} = D_{11}r$$

$$R_{34} = R_{43} = \frac{D_{11}R'_1 r}{R_1^{\,2}} - K_{12}r' - \frac{D_{12}r'}{R_1}$$

$$R_{35} = R_{53} = - \frac{D_{11}r}{R_1} - K_{11}r$$

$$R_{44} = \frac{C_{22}(r')^2}{r} + \frac{D_{11}(R'_1)^2 r}{R_1^{\,4}} - \frac{2D_{12}R'_1 r'}{R_1^{\,3}} + \frac{D_{22}(r')^2}{rR_1^{\,2}} - \frac{2K_{12}r'R'_1}{R_1^{\,2}} + \frac{2K_{22}(r')^2}{rR_1}$$

$$R_{45} = R_{54} = C_{12}r' - \frac{D_{11}R'_1 r}{R_1^{\,3}} + \frac{D_{12}r'}{R_1^{\,2}} - \frac{K_{11}R'_1 r}{R_1^{\,2}} + \frac{2K_{12}r'}{R}$$

$$R_{55} = C_{11}r + \frac{D_{11}r}{R_1^{\,2}} + \frac{2K_{11}r}{R_1}$$

## APPENDIX C

## COMPUTER PROGRAM SUBROUTINES

| | | |
|---|---|---|
| MAIN PROGRAM SHELL | - | Parameter values set, calls subroutines SHELLS, BANDED, VECTOR, MODE. |
| SHELLS | - | Reads input; calls subroutines CASE, TRAN, FORC, SUMAT, BOUND, and BOUNF, and calculates constant coefficients of $T_{ka}$ and X matrices. |
| TRAN | - | Calculates element transformation matrices $T_k$; calls PEST. |
| SUMAT | - | Calculates element stiffness matrices $S_k$; calls PEST. |
| FORC | - | Calculates element force matrices $G_k$; calls PEST. |
| PEST | - | Calculates all functions of radius. |
| ELIMB | - | Deletes a row and a column from a matrix. |
| CASE | - | Determines rows and columns to be deleted from mass, force, and stiffness matrices to satisfy boundary conditions. |
| BOUN | - | Calls ELIMB. |
| VECTOR | - | Puts boundary zeros in vector, calls BACK. |
| BACK | - | Zeros inserted into vector. |
| MODE | - | Calculates displacements, stresses, and strains along meridian from vector and volume change. |
| BANDED | - | Calculates displacement vector. |
| BOUNF | - | Deletes rows from force matrix column to satisfy boundary conditions. |

The bellows model consists of toroidal segments (ITP=1) and conical segments (ITP=2). Each segment can have an arbitrary number of elements.

FORTRAN PARAMETER values set are:

NSEG - Number of segments

MEL - Total number of elements.

NMAX - Number of equations = 5*MEL + 5

N300 - Total number of output points

= ININ*MEL + 1

where ININ is an integer number of integration points per element.

Input is quite simple and is listed below.

| CARD | FORMAT | QUANTITIES AND DEFINITION |
|---|---|---|
| 1 | 20A4 | Identification |
| 2 | 7I4 | ICASE, identifies boundary conditions. IPRINT, selects items to be printed (0 or 1 for delections only. 2 for above, plus mass and stiffness matrices) ISTRN, set to 1 for strain calculations. ISTRES, set to 1 for stress calculations. |
| 3 | 5E14.8 | SO, coordinate of initial shell edge. RO, reference radius for thickness. |
| NSEG Cards | 2I5,4E15.8 | ITP, segment type, 1 for toroidal, 2 for cone. NEL, number of elements in segment For ITP=1, entries are segment length, major radius, minor radius, and starting X. For ITP = 2, entries are segment length. Starting radius, cos θ, and sin θ. |
| Next | 5E14.8 | Material properties and load $E_1$, $E_2$, $\mu_1$, $\mu_2$, pressure, reference thickness, $G_{12}$. |

```
      PROGRAM SHELL
C     FINITE-ELEMENT METHOD FOR COMPUTING STATIC DEFLECTIONS
C     LARRY KIEFLING, MARSHALL SPACE FLIGHT CENTER
C     ADAPTED FROM NASA TMX-2138,''USER'S MANUAL FOR A
C     DIGITAL COMPUTER PROGRAM FOR COMPUTING THE VIBRATION
C     CHARACTERISTICS OF RING-STIFFENED ORTHOTROPIC SHELLS
C     OF REVOLUTION ''
C
C****SET PARAMETERS IN SUBROUTINE SHELLS
C***NSEG= NO. OF SEGMENTS, MEL = NO. OF ELEMENTS
C*** SET NMAX ' 5 * MEL + 5
C***SET PARAMETER N300 = ININ*MEL + 1 IN SUBROUTINE MODE
C*** SET PARAMETER NSEG IN SUBROUTINE PEST ALSO
      PARAMETER (MEL=79, NMAX=400)
      COMMON /BLK/YOUNG1,XMU1,TH,YOUNG2,XMU2,G12,R0
      COMMON/LIN/ISTRN,ISTRES,ININ,S,E,TRANS,SO,K,KN,NUM,LN,NELIM
      DIMENSION D(9),AM(9),A(55),B(NMAX),EVEC(NMAX),NELIM(8),
     1S(MEL),E(MEL)
      DIMENSION TRANS(10,10)
      DOUBLE PRECISION D,AM,A,B
      CALL SHELLS
      CALL BANDED(9,55,10,KN,19,1,11,12,13,14,D,AM,A,B)
      REWIND 13
      DO 160 I=1,KN
      READ (13)  B(I)
      J=KN-I+1
  160 EVEC(J)= SNGL(B(I))
   61 CALL VECTOR(NUM,KN,NMAX,LN,NELIM,EVEC)
      WRITE (6,1020)
   53 CONTINUE
      WRITE (6,1015)
      WRITE (6,1064)(EVEC(J  ),J=1,LN)
      WRITE (6,1020)
   66 CALL MODE(   ISTRN,ISTRES,ININ,S,E,EVEC,TRANS,SO,K)
 1015 FORMAT(///1X,6HVECTOR,7X,1HW,19X,1HU,18X,4HBETA,15X,7HU PRIME,
     111X,10HBETA PRIME)
 1020 FORMAT(1H1/////)
 1064 FORMAT (1X,5E20.8)
      END
      SUBROUTINE SHELLS
      PARAMETER (NSEG=11, MEL=79, NMAX=400 ,N300=791)
      COMMON/SEG/    ITP,NEL,PAR1,PAR2,PAR3,PAR4
      COMMON/LIN/ISTRN,ISTRES,ININ,S,E,TRANS,SO,K,KN,NUM,LN,NELIM
      COMMON /BLK/YOUNG1,XMU1,TH,YOUNG2,XMU2,G12,R0
      DIMENSION TRANS(10,10),X(5,10),R(10,10),TEP(10,10),SUMS(10,10),
     1IDEN(20),NELIM(8),DST(10),
     2S(MEL),E(MEL),ST(NMAX,10),FORCE(NMAX),
     3ITP(NSEG),NEL(NSEG),PAR1(NSEG),PAR2(NSEG),PAR3(NSEG),PAR4(NSEG)
     *,SUMX(10)
      DOUBLE PRECISION FOR,DST
      ININ=(N300-1)/MEL
      DO 99 I=1,NMAX
   99 FORCE(I)=0.
      MSEG=NSEG
      PI=3.14159265358979
    1 PRINT 1020
      READ(5,1000)IDEN
    3 WRITE(6,1000)IDEN
C     IF IPRINT.EQ.1, STIFFNES MATRIX NOT PRINTED AND MODAL COLUMN
C     PRINTED
C     IF PRINT.EQ.2, STIFFNESS MATRIX PRINTED AND MODAL COLUMN PRINTED
```

```
      READ (5,1001) ICASE,IPRINT,ISTRN,ISTRES
      WRITE(6,1010)
      WRITE(6,1009) ININ,ICASE,IPRINT,ISTRN,ISTRES
  500 READ (5,1002) SO,RO
      DO 501 I=1,NSEG
      READ (5,1011) ITP(I),NEL(I),PAR1(I),PAR2(I),PAR3(I),PAR4(I)
  501 CONTINUE
      K=0
      KK=0
      DO 503 I=1,NSEG
      K = K+NEL(I)
      II=NEL(I)
      DO 504 J=1,II
      KK=KK+1
  504 E(KK)=PAR1(I)/FLOAT(NEL(I))
  503 CONTINUE
      S(1)=SO+.5*E(1)
      IF(K.EQ.1) GO TO 200
      DO 7 I=2,K
      SUM=SO
      II=I-1
      DO 8 J=1,II
    8 SUM=SUM+E(J)
    7 S(I)=SUM+.5*E(I)
  200 WRITE(6,1003)
      DO 4 I=1,K
    4 WRITE(6,1004)I,E(I),S(I)
      READ(5,1002)YOUNG1,YOUNG2,XMU1,XMU2,PRES,TH,G12
      WRITE(6,1019)SO,RO,YOUNG1,YOUNG2,XMU1,XMU2,PRES,TH,G12
C     BOUNDARY CONDITION CODE  (SEE TN FOR DETAILS)
C       ICASE=4  - FREE-SIMPLY SUPPORTED
C       ICASE=5  - SIMPLY SUPPORTED-FREE
C       ICASE=6  - FREE-CLAMPED
C       ICASE=7  - CLAMPED-FREE
C       ICASE=9  - SIMPLY SUPPORTED-SIMPLY SUPPORTED
C       ICASE=10 - CLAMPED-CLAMPED
C       ICASE=11 - FREELY SUPPORTED-SIMPLY SUPPORTED
C       ICASE=12 - FREELY SUPPORTED-CLAMPED
C       ICASE=13 - SIMPLY SUPPORTED-FREELY SUPPORTED
C       ICASE=14 - SIMPLY SUPPORTED-CLAMPED
C       ICASE=15 - CLAMPED-FREELY SUPPORTED
C       ICASE=16 - CLAMPED-SIMPLY SUPPORTED
C       ICASE=17 - SYMMETRIC HALF - CLAMPED
C       ICASE=18 - FLOATING RADIAL SUPPORTS (FRS-FRS)
      IF(ICASE.EQ.4) PRINT 1024
      IF(ICASE.EQ.5) PRINT 1025
      IF(ICASE.EQ.6) PRINT 1026
      IF(ICASE.EQ.7) PRINT 1027
      IF(ICASE.EQ.9) PRINT 1029
      IF(ICASE.EQ.10) PRINT 1030
      IF(ICASE.EQ.11) PRINT 1031
      IF(ICASE.EQ.12) PRINT 1032
      IF(ICASE.EQ.13) PRINT 1033
      IF(ICASE.EQ.14) PRINT 1034
      IF(ICASE.EQ.15) PRINT 1035
      IF(ICASE.EQ.16) PRINT 1036
      IF(ICASE.EQ.17) PRINT 1060
      IF(ICASE.EQ.18) PRINT 1063
      CALL CASE (ICASE,K,NELIM,NUM)
      REWIND 9
C     TRANSFORMATION MATRIX FOR EACH ELEMENT COMPUTED AND WRITTEN ON
```

```
C     FILE 9
      DO 13 I=1,10
      DO 13 J=1,10
   13 TRANS(I,J)=0.
      TRANS(1,1)=.5
      TRANS(1,6)=.5
      TRANS(2,3)=-7./16.
      TRANS(2,8)=-7./16.
      TRANS(3,5)=-1./8.
      TRANS(3,10)=-1./8.
      TRANS(7,2)=.5
      TRANS(7,7)=.5
      TRANS(8,4)=-.25
      TRANS(8,9)=-.25
      DO 14 KK=1,K
      E1=E(KK)
      CALL TRAN(E1,   TRANS,KK)
      WRITE(9)((TRANS(I,J),J=1,10),I=1,10)
   14 CONTINUE
      REWIND 9
      DO 16 I=1,2
      DO 16 J=1,10
   16 X(I,J)=0.
      DO 29 KK=1,K
      E1=E(KK)
      DO 28 I=1,10
   28 SUMX(I  )=0.
      CALL FORC (ININ,E1,PRES,SUMX,KK)
      READ(9)((TRANS(I,J),J=1,10),I=1,10)
   30 DO 31 I=1,10
      TEP(I,1)=0.
      DO 31 IJ=1,10
   31 TEP(I,1)=TEP(I,1)+TRANS(IJ,I)*SUMX(IJ)
      DO 101 I=1,10
      II=(KK-1)*5 +I
  101 FORCE(II)=FORCE(II)+TEP(I,1)
   29 CONTINUE
      REWIND 9
C     A STIFFNESS MATRIX COMPUTED
      KN=5*(K+1)
      DO 5 I=1,KN
      DO 5 J=1,10
    5 ST(I,J)=0.
      DO 11 I=1,5
      DO 11 J=1,10
   11 X(I,J)=0.
      X(1,1)=1.
      X(2,2)=1.
      X(3,3)=2.
      X(4,7)=1.
      X(5,8)=1.
      DO 10 KK = 1,K
      E1=E(KK)
      DO 23 I=1,10
      DO 23 J=1,10
   23 SUMS(I,J)=0.
      CALL SUMAT(ININ,E1,   X,R,TEP,SUMS,KK)
      READ(9)((TRANS(I,J),J=1,10),I=1,10)
      DO 17 I=1,10
      DO 17 J=1,10
      TEP(I,J) = 0.
```

```
      DO 17 IJ=1,10
   17 TEP(I,J)=TEP(I,J) + TRANS(IJ,I) * SUMS(IJ,J)
      DO 18 I=1,10
      DO 18 J=1,10
      SUMS(I,J)=0.
      DO 18 IJ=1,10
   18 SUMS(I,J)=SUMS(I,J)+TEP(I,IJ)*TRANS(IJ,J)
      DO 19 I=1,10
      II = (KK-1)*5 + I
      DO 19 J=I,10
      JJ= J-I+1
   19 ST(II,JJ)=ST(II,JJ)+SUMS(I,J)
   10 CONTINUE
      CON=PI*2.
      DO 6 I=1,KN
      FORCE(I)=CON*FORCE(I)
      DO 6 J=1,10
    6 ST(I,J) = CON*ST(I,J)
C     ROWS AND COLUMNS DELETED FROM STIFFNESS MATRIX TO
C     SATISFY BOUNDARY CONDITION
      CALL BOUN(NUM,KN,NMAX,NELIM,ST)
      CALL BOUNF(NUM,NMAX,NELIM,FORCE)
      REWIND 11
      KNM=KN-9
      DO 150 I=1,KN
      FOR =DBLE(FORCE(I))
      JJ= 10
      IF(I.GT.KNM) JJ=KN-I+1
      DO 151 J=1,JJ
  151 DST(J) =DBLE(ST(I,J))
      WRITE (11) (DST(J),J=1,JJ)
      WRITE (11) FOR
  150 CONTINUE
      WRITE (6,1020)
      WRITE(6,1061)
      WRITE(6,1064)(FORCE(I),I=1,KN)
   44 CONTINUE
      IF(IPRINT.LT.2) GO TO 80
      WRITE(6,1005)
      DO 36 I=1,KN
      WRITE(6,1007)I
      JJ=10
      IF(I.GT.KNM) JJ=KN-I+1
   36 WRITE(6,1008)(ST(I,J),J=1,JJ)
   80 CONTINUE
      REWIND 11
      REWIND 12
      REWIND 13
      REWIND 14
 1000 FORMAT(20A4)
 1001 FORMAT(10I4)
 1002 FORMAT(5E14.8)
 1003 FORMAT(///14X,11HEPSILON (K),10X,5HS (K))
 1004 FORMAT(4X,I4,2(2X,E16.8))
 1005 FORMAT(//4X,16HSTIFFNESS MATRIX/)
 1007 FORMAT(2X,3HROW,I3)
 1008 FORMAT(8E16.8)
 1009 FORMAT(10I10)
 1010 FORMAT(50H       ININ      ICASE     IPRINT      ISTRN     ISTRES  )
 1011 FORMAT (2I5,4E15.8)
 1019 FORMAT(//11X,9HS0,  R0 =,2E16.8/2X,18HYOUNGS MODULUS 1 =,
```

```
     1                                                       E16.8/2X,18HY Y
     2OUNGS MODULUS 2 =,E16.8/2X,18HPOISSONS RATIO 1 =,
     3                                                   E16.8/2X,18HPOISSON
     4S RATIO 2 =,E16.8/15X,5HPRES=,E16.8/9X,11HTHICKNESS =,
     5                                                     E16.8/10X,10HG
     6SUB 12 =,E16.8)
1020 FORMAT(1H1/////)
1024 FORMAT(//2X,'FREE-SIMPLY SUPPORTED BOUNDARY CONDITION - (5K+1),
     1 (5K+2) ROWS AND COLUMNS DELETED')
1025 FORMAT(//2X,'SIMPLY SUPPORTED-FREE BOUNDARY CONDITION - 1,2,
     1  ROWS AND COLUMNS DELETED')
1026 FORMAT(//2X,'FREE-CLAMPED BOUNDARY CONDITION - (5K+1),(5K+2),(5K+
     1 3), ROWS AND COLUMNS DELETED')
1027 FORMAT(//2X, 'CLAMPED-FREE BOUNDARY CONDITION - FIRST3 ROWS AND C
     1OLUMNS DELETED')
1029 FORMAT(//2X,'SIMPLY SUPPORTED-SIMPLY SUPPORTED BOUNDARY CONDITION
     1 - 1,2, (5K+1), (5K+2) ROWS AND COLUMNS DELETED')
1030 FORMAT(//2X,  'CLAMPED-CLAMPED BOUNDARY CONDITION - FIRST 3 AND (5
     1K+1),(5K+2),(5K+3)         ROWS AND COLUMNS DELETED')
1031 FORMAT(//2X,  'FREELY SUPPORTED-SIMPLY SUPPORTED BOUNDARY CONDITIO
     1N - 1,  (5K+1),(5K+2)          ROWS AND COLUMNS DELETED')
1032 FORMAT(//2X,'FREELY SUPPORTED-CLAMPED BOUNDARY CONDITION - 1,
     1 AND (5K+1),(5K+2),(5K+3) ROWS AND COLUMNS DELETED')
1033 FORMAT(//2X,'SIMPLY SUPPORTED-FREELY SUPPORTED BOUNDARY CONDITION
     1 - FIRST 2, (5K+1) ROWS AND COLUMNS DELETED')
1034 FORMAT(//2X,  'SIMPLY SUPPORTED-CLAMPED BOUNDARY CONDITION - FIRST
     1 2 AND (5K+1),(5K+2),(5K+3)          ROWS AND COLUMNS DELETED')
1035 FORMAT(//2X,'CLAMPED-FREELY SUPPORTED BOUNDARY CONDITION - FIRST
     1 3 AND (5K+1) ROWS AND COLUMNS DELETED')
1036 FORMAT(//2X,'CLAMPED-SIMPLY SUPPORTED BOUNDARY CONDITION - FIRST
     1 3 AND(5K+1),(5K+2) ROWS AND COLUMNS DELETED')
1060 FORMAT(//2X,'SYMMETRIC HALF-CLAMPED - 2,3,(5K+1),(5K+2),
     1 AND (5K+3) ROWS AND COLUMNS DELETED')
1061 FORMAT (///15H FORCE MATRIX  )
1063 FORMAT(//2X,'FLOATING RADIAL SUPPORTS - 2,3, (5K+2), AND
     1 (5K+3) ROWS AND COLUMNS DELETED')
1064 FORMAT (1X,5E20.8)
     GO TO 2000
2001 FORMAT (13H ERROR IN ROW,I5,11H OF INVERSE)
2000 CONTINUE
     RETURN
     END
     SUBROUTINE TRAN(E1,   TRANS,KK)
C    COMPUTATION OF TRANSFORMATION MATRIX
     DIMENSION TRANS(10,10)
     E2=E1*E1
     E3=E1*E2
     E4=E1*E3
     E5=E1*E4
     TRANS(1,3)=5.*E1/32.
     TRANS(1,5)=E2/64.
     TRANS(1,8)=-5.*E1/32.
     TRANS(1,10)=E2/64.
     TRANS(2,1)=-15./(8.*E1)
     TRANS(2,5)=-E1/32.
     TRANS(2,6)=15./(8.*E1)
     TRANS(2,10)=E1/32.
     TRANS(3,3)=-.75/E1
     TRANS(3,8)=.75/E1
     TRANS(4,1)=5./E3
     TRANS(4,3)=2.5/E2
```

```
      TRANS(4,5)=.25/E1
      TRANS(4,6)=-5./E3
      TRANS(4,8)=2.5/E2
      TRANS(4,10)=-.25/E1
      TRANS(5,3)=.5/E3
      TRANS(5,5)=.25/E2
      TRANS(5,8)=-.5/E3
      TRANS(5,10)=.25/E2
      TRANS(6,1)=-6./E5
      TRANS(6,3)=-3./E4
      TRANS(6,5)=-.5/E3
      TRANS(6,6)=6./E5
      TRANS(6,8)=-3./E4
      TRANS(6,10)=.5/E3
      TRANS(7,4)=E1/8.
      TRANS(7,9)=-E1/8.
      TRANS(8,2)=-1.5/E1
      TRANS(8,7)=1.5/E1
      TRANS(9,4)=-.5/E1
      TRANS(9,9)=.5/E1
      TRANS(10,2)=2./E3
      TRANS(10,4)=1./E2
      TRANS(10,7)=-2./E3
      TRANS(10,9)=1./E2
      X1=.5*E1
      CALL PEST(3,0,-X1,FR1,KK)
      CALL PEST(5,0,-X1,PR1,KK)
      CALL PEST(3,0,X1,FR2,KK)
      CALL PEST(5,0,X1,PR2,KK)
      FF1=.5*E1*PR1*FR1
      FF2=.5*E1*PR2*FR2
      TRANS(1,2)=E1*FR1*(5.-FF1)/32.
      TRANS(1,4)=E2*FR1/64.
      TRANS(1,7)=-E1*FR2*(5.+FF2)/32.
      TRANS(1,9)=E2*FR2/64.
      TRANS(2,2)=FR1*(-7.+FF1)/16.
      TRANS(2,4)=-E1*FR1/32.
      TRANS(2,7)=-FR2*(7.+FF2)/16.
      TRANS(2,9)=E1*FR2/32.
      TRANS(3,2)=FR1*(-3.+FF1)/(4.*E1)
      TRANS(3,4)=-FR1/8.
      TRANS(3,7)=FR2*(3.+FF2)/(4.*E1)
      TRANS(3,9)=-FR2/8.
      TRANS(4,2)=FR1*(5.-FF1)/(2.*E2)
      TRANS(4,4)=FR1/(4.*E1)
      TRANS(4,7)=FR2*(5.+FF2)/(2*E2)
      TRANS(4,9)=-FR2/(4.*E1)
      TRANS(5,2)=FR1*(1.-FF1)/(2.*E3)
      TRANS(5,4)=FR1/(4.*E2)
      TRANS(5,7)=-FR2*(1.+FF2)/(2.*E3)
      TRANS(5,9)=FR2/(4.*E2)
      TRANS(6,2)=FR1*(-3.+FF1)/E4
      TRANS(6,4)=-FR1/(2.*E3)
      TRANS(6,7)=-FR2*(3.+FF2)/E4
      TRANS(6,9)=FR2/(2.*E3)
      RETURN
      END
      SUBROUTINE FORC (ININ,E1,PRES,SUMN,KK)
C     ELEMENT MASS MATRIX COMPUTED BY NUMERICAL INTEGRATION USING THE
C     TRAPEZOIDAL RULE
      DIMENSION Y( 10),     TEP(10),SUMN(10)
```

```
      FININ=FLOAT(ININ)
      DEL=E1/FININ
      NN=ININ+1
      Y(1  ) = 1.
      DO 2 I=7,10
    2 Y(I)=0.
      DO 1 IN=1,NN
      X1=-.5*E1+DEL*FLOAT(IN-1)
      Y(  2)=X1
      Y(  3)=X1*X1
      Y(  4)=X1*Y(  3)
      Y(  5)=X1*Y(  4)
      Y(  6)=X1*Y(  5)
      CALL PEST(2,0,X1,FR1,KK)
      R=PRES*FR1
      DO 3 I=1,10
    3 TEP(I  )=          Y(   I)*R
    6 CON=DEL
      IF((IN.EQ.1).OR.(IN.EQ.NN)) CON=.5*DEL
      DO 8 I=1,10
    8 SUMN(I  )=SUMN(I  )+CON*TEP(I  )
    1 CONTINUE
      RETURN
      END
      SUBROUTINE SUMAT(ININ,E1,    X,R,TEP,SUMS,KK)
C     ELEMENT STIFFNESS MATRIX COMPUTED BY NUMERICAL INTEGRATION USING
C     THE TRAPEZOIDAL RULE
      DIMENSION X(5,10),R(10,10),TEP(10,10),SUMS(10,10)
      INTG=ININ
      FINTG=FLOAT(INTG)
      DEL=E1/FINTG
      NN=INTG+1
      DO 6 IN=1,NN
      X1=-.5*E1+DEL*FLOAT(IN-1)
      X2=X1*X1
      X3=X1*X2
      X4=X1*X3
      X5=X1*X4
      X(1,2)=X1
      X(1,3)=X2
      X(1,4)=X3
      X(1,5)=X4
      X(1,6)=X5
      X(2,3)=2.*X1
      X(2,4) = 3. * X2
      X(2,5)=4.*X3
      X(2,6)=5.*X4
      X(3,4)=6.*X1
      X(3,5)=12.*X2
      X(3,6)=20.*X3
      X(4,8)=X1
      X(4,9)=X2
      X(4,10)=X3
      X(5,9)=2.*X1
      X(5,10)=3.*X2
      INT=0
      DO 7 I = 1,5
      DO 7 J = I,5
      INT=INT+1
      R(I,J)=0.
      CALL PEST(1,INT,X1,R(I,J),KK)
```

```
      IF(I.EQ.J) GO TO 7
      R(J,I)=R(I,J)
    7 CONTINUE
      DO 8 I=1,5
      DO 8 J=1,10
      TEP(J,I)=0.
      DO 8 IJ=1,5
    8 TEP(J,I)=TEP(J,I)+X(IJ,J)*R(IJ,I)
      DO 9 I=1,10
      DO 9 J=1,10
      R(I,J)=0.
      DO 9 IJ=1,5
    9 R(I,J)=R(I,J)+TEP(I,IJ)*X(IJ,J)
      CON=DEL
      IF((IN.EQ.1).OR.(IN.EQ.NN)) CON=.5*DEL
      DO 12 I=1,10
      DO 12 J=1,10
   12 SUMS(I,J)=SUMS(I,J)+CON*R(I,J)
    6 CONTINUE
      RETURN
      END
      SUBROUTINE BOUN(NUM,N,NMAX,NROW,ST)
C     ROWS AND COLUMNS DELETED TO SATISFY BOUNDARY CONDITION
      DIMENSION NROW(8),ST(NMAX,10)
      NN=0
      DO 1 K=1,NUM
      NE=NROW(K)-NN
      CALL ELIMB(NE,N,NMAX,10,ST)
      NN=NN+1
      N=N-1
    1 CONTINUE
      RETURN
      END
      SUBROUTINE BOUNF(NUM,NMAX,NELIM,FORCE)
      DIMENSION NELIM(8),FORCE(NMAX),NE(8)
      DO 1 K=1,NUM
    1 NE(K)=NELIM(K)
      DO 2 K=1,NUM
      DO 6 I=1,NMAX
      IF(I.NE.NE(K)) GO TO 5
      NNMAX=NMAX-1
      DO 3 J=I,NNMAX
    3 FORCE(J)=FORCE(J+1)
      K1=K+1
      DO 4 J=K1,NUM
    4 NE(J)=NE(J)-1
      GO TO 2
    5 CONTINUE
    6 CONTINUE
    2 CONTINUE
      RETURN
      END
      SUBROUTINE PEST(ICODE,INT,S,RR,KK)
      PARAMETER(NSEG=11)
      COMMON /BLK/YOUNG1,XMU1,THO,YOUNG2,XMU2,G12,R0
      COMMON /STR/R1,R2,R1P,R,RP,C11,C12,C22,D11,D12,D22,K11,K12,K22
      COMMON/SEG/    ITP,NEL,PAR1,PAR2,PAR3,PAR4
      REAL K11,K12,K22
      DIMENSION ITP(NSEG),NEL(NSEG),PAR1(NSEG),PAR2(NSEG),
     1PAR3(NSEG),PAR4(NSEG)
C     FUNCTIONS DESCRIBING GEOMETRICALLY EXACT ELEMENT USED TO COMPUTE
```

```
C     MATRIX R
      J=0
      DO 500 I=1,NSEG
      J=J+NEL(I)
      IF(KK.LE.J) GO TO 501
  500 CONTINUE
  501 FN = FLOAT(KK-J+NEL(I)-1)
      FNEL=NEL(I)
      II=ITP(I)
      GO TO (101,102),II
C        ITP=1   TOROIDAL SEGMENT, PARAMETERS ARE
C        LENGTH, MAJOR RADIUS, MINOR RADIUS, STARTING X
C        MINOR RADIUS IS NEGATIVE FOR INNER PART
  101 SS=S+(FN+.5)*PAR1(I)/FNEL+PAR4(I)
      CZ=COS(SS/PAR3(I))
      R=PAR2(I)+PAR3(I)*CZ
      RP=-SIN(SS/PAR3(I))
      R1=1./PAR3 (I)
      R2=CZ/R
      GO TO 150
C        ITP=2   CONICAL SEGMENT   PARAMETERS ARE
C        LENGTH, STARTING RADIUS, COS THETA, SIN THETA
  102 SS=S+(FN+.5)*PAR1(I)/FNEL
      R=PAR2(I)+SS*PAR4(I)
      RP=PAR4(I)
      R1=0
      R2=PAR3(I)/R
  150 R1P =0.
      TH=THO*RO/R
      IF(ICODE.EQ.1) GO TO 29
      IF(ICODE.EQ.2) GO TO 30
      IF(ICODE.EQ.4) GO TO 29
      IF(ICODE.EQ.5) GO TO 32
      RR=R1
      RETURN
   30 RR=R
      RETURN
   32 RR = R1P
      RETURN
   29 CONTINUE
      C11=YOUNG1*TH/(1.-XMU1**2)
      C12=XMU1*C11
      C22=C11
      D11=YOUNG1*TH**3/(12.*(1.-XMU1**2))
      D12=XMU1*D11
      D22=D11
      K11=0.
      K12=0.
      K22=0.
      IF(ICODE.EQ.1) GO TO 31
      RR=0.
      RETURN
C     ELEMENTS OF R MATRIX ARE FUNCTIONS OF THE MERIDIONAL COORDINATE
   31 GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15),INT
    1 RR = C11*R*R1**2+2.*C12*R*R1*R2+C22*R*R2**2
      RETURN
    2 RR = -K12*RP*R1-K22*RP*R2
      RETURN
    3 RR = -K11*R*R1-K12*R*R2
      RETURN
    4 RR = C12*RP*R1+C22*RP*R2 + K22*RP*R1*R2
```

```
      RR = RR-K11*R1P*R*R1**3+K12*RP*R1**2-K12*R*R1P*R1**2*R2
      RETURN
    5 RR = C11*R*R1+C12*R*R2 +K11*R*R1**2+K12*R*R1*R2
      RETURN
    6 RR = D22*RP**2/R
      RETURN
    7 RR=D12*RP
      RETURN
    8 RR = D12*R1P*RP*R1**2-D22*RP**2*R1/R -K22*RP**2/R
      RETURN
    9 RR=-D12*RP*R1 -K12*RP
      RETURN
   10 RR=D11*R
      RETURN
   11 RR=D11*R1P*R*R1**2-D12*RP*R1 -K12*RP
      RETURN
   12 RR=-D11*R*R1 -K11*R
      RETURN
   13 RR=C22*RP**2/R+D11*R1P**2*R*R1**4-2.*D12*R1P*RP*R1**3
      RR = RR-K12*RP*R1P*R1**2*2.+2.*K22*RP**2*R1/R+D22*RP**2*R1**2/R
       RETURN
   14 RR=C12*RP-D11*R1P*R*R1**3+D12*RP*R1**2
      RR=RR-K11*R*R1P*R1**2+2.*K12*RP*R1
      RETURN
   15 RR=C11*R+D11*R*R1**2 +K11*R*R1*2.
      RETURN
      END
      SUBROUTINE ELIMB(NE,N,NMAX,NB,A)
C *** ROW AND COLUMN DELETED FROM BANDED MATRIX A      26 JANUARY 1972
C *** NE=ROW AND COLUMN ELIMINATED      N=SIZE OF MATRIX A (ROWS)
C *** NB=SEMI-BAND WIDTH (COLUMNS)      NMAX=MAXIMUM SIZE OF MATRIX A
      DIMENSION A(NMAX,NB)
      M=N-1
      IF (NE.GT.M) GO TO 2
      DO 1 I=NE,M
      DO 1 J=1,NB
    1 A(I,J)=A(I+1,J)
    2 L=NB-1
      DO 4 K=2,L
      I=NE-K+1
      IF (I.LE.0) RETURN
      DO 3 J=K,L
    3 A(I,J)=A(I,J+1)
    4 A(I,NB)=0
      RETURN
      END
      SUBROUTINE BANDED (II1,II2,II3,NIN,M,NRHS,NNIT,NOT,NANST,NMT,D,AM,
     1A,B)
C   ARGUMENTS...
C               M=BANDWITH.
C             II1=(M-1)/2,DIMENSION OF D AND AM ARRAYS. (NDM)
C             II2=(M+1)*(M+3)/8,DIMENSION OF A ARRAY. (NT)
C             II3=(M+1)/2, ROW DIMENSION OF B. (NDMP1)
C             NIN=NO. OF EQUATIONS.
C            NRHS=NO. OF RIGHT HAND SIDES.
C            NNIT=INPUT TAPE NO.  EACH RECORD MUST BE A ROW OF COEFF. OF
C                 THE EQ.  THOSE COEFF. STARTING WITH THE DIAGONAL,OUT TO
C                 THE END OF THE BAND ARE ENTERED. (M+1)/2 ELEMENTS ARE
C                 ENTERED. A SEPARATE RECORD CONTAINING THE NRHS CONSTANT
C                 S FOLLOWS EACH ROW. PREFIX WITH (-) FOR CHECKOUT OUTPUT
C             NOT=TAPE NO. ON WHICH THE TRIANGULARIZED MATRIX IS TO BE
```

```
C                 STORED WITH THE MODIFIED R.H.S.,IF ANY
C           NANST=TAPE NO. ON WHICH THE SOLUTIONS ARE TO BE WRITTEN. EACH
C                 RECORD WILL CONTAIN THE NRHS SOLUTIONS FOR THE VARIABLE
C                 IN QUESTION.
C             NMT=TAPE NO. ON WHICH THE MULTIPLYING FACTORS MAY BE STORED
C                 THE (M-1)/2 FACTORS ARE STORED AS A RECORD FOR EACH ROW
C                 THE 1ST (M-1)/2 ROWS WILL HAVE ONLY I-1 FACTORS,WHERE I
C                 IS THE ROW NUMBER.  ITFOLLOWS THAT NONE ARE STORED FOR
C                 THE 1ST ROW.
C            D(I)=STORAGE FOR THOSE DIAG. ELEMENTS NEEDED IN TRIANGULAR-
C                 IZATION OF A PARTICULAR ROW.
C           AM(I)=STORAGE FOR THE M(I,J) FOR THE ROW BEING OPERATED ON.
C            A(J)=STORAGE FOR THAT TRIANGULAR MATRIX NEEDED WHEN OPERAT-
C                 ING ON A PARTICULAR ROW.
C          B(K,L)=STORAGE FOR THE L R.H.S. FOR THE K VARIABLES NEEDED AT
C                 ONE TIME.  THE R.H.S. ARE OPERATED ON AT THE SAME TIME
C                 THE TRIANGULARIZATION TAKES PLACE
C        NOTE.....ALL TAPES MUST BE READY TO USE,I.E.,NO REWINDING WILL
C        BE DONE AT THE OUTSET.  PROGRAM WILL RETURN WITH SOLUTIONS ON
C        TAPE NANST READY TO READ THE NRHS VALUES OF THE NTH UNKNOWN.
      DIMENSION D(II1),AM(II1),A(II2),B(II3,NRHS)
      DOUBLE PRECISION D,AM,A,B

      NIT=IABS(NNIT)
        N=IABS(NIN)
   20 IF(NIT.NE.5.AND.NIT.NE.6.AND.NOT.NE.5.AND.NOT.NE.6.AND.NANST.NE.5.
     1AND.NANST.NE.6.AND.NMT.NE.5.AND.NMT.NE.6.AND.N.GT.M.AND.MOD(M,2).N
     2E.0) GO TO 40
   30 WRITE (6,5000) IERR
                 CALL EXIT
                 STOP
   40   NDM=(M-1)/2
      NDMP1=NDM+1
          NT=(M+1)*(M+3)/8
        NL1=NDM*(NDM+1)/2
         NL=NL1+1
       NDM1=NDM-1
        NT1=NT-1
        NL2=NT-M+1
        LLM=M-3
        LLT=LLM/2
       NNDM=N-NDM1
        NNN=N-2*NDM
C READ 1ST ROW FROM TAPE (NIT)
      READ (NIT) D(1),(A(I),I=NL2,NL1)
CHECK IF DIAG. ELEMENT IS 0
                                                         IERR=2
                                                 IF(D(1)) 50,30,50
   50 KBIG=1
C WRITE OUT 1ST ROW IF REQUESTED
                                                         IERR=3
                                                 IF(NNIT) 60,30,70
   60 WRITE (6,5010) KBIG,D(1),(A(I),I=NL2,NL1)
C READ R.H.S. FROM TAPE (NIT),WRITE R.H.S ON TAPE (NOT), IF NRHS NOT 0.
   70                                                    IERR=4
                                                 IF(NRHS) 30,80,90
C WRITE ALTERED ROW ON TAPE (NOT) IF NRHS=0.
   80 WRITE (NOT) D(1),(A(I),I=NL2,NL1)
                                                       GO TO 120
   90 READ (NIT) (B(1,I),I=1,NRHS)
C WRITE ALTERED ROW AND R.H.S. ON TAPE (NOT) IF NRHS NOT ZERO
```

```
      WRITE (NOT) D(1),(A(I),I=NL2,NL1),(B(1,I),I=1,NRHS)
C SHIFT DOWN R.H.S. IF NRHS NOT ZERO
                              DO 100 J=1,NRHS
  100 B(2,J)=B(1,J)
C WRITE OUT INPUT R.H.S. IF REQUESTED AND IF NRHS NOT ZERO
                                                   IF(NNIT) 110,30,120
  110 WRITE (6,5020) KBIG,(B(1,J),J=1,NRHS)
C*****ALTER ROWS 2 TO (M-1)/2 IF M GREATER THAN 3  *********************
  120                                              IF(NDM1) 30,380,130
  130 JO=NL2
      LO=NL1-NL2
                              DO 370 K=1,NDM1
      KBIG=KBIG+1
C READ ROW K+1 FROM TAPE (NIT)
      READ (NIT) (A(I),I=NL,NT)
CHECK IF DIAG. IS ZERO
                                                               IERR=5
                                                  IF(A(NL)) 140,30,140
C WRITE OUT INPUT ROW IF REQUESTED
  140                                              IF(NNIT) 150,30,160
  150 WRITE (6,5010) KBIG,(A(I),I=NL,NT)
COMPUTE THE M(I,J)
  160 L=LO+1
      J=JO
                              DO 170 I=1,K
      AM(I)=-A(J)/D(I)
      J=J+L
  170 L=L+1
      JO=JO-LO
      LO=LO-1
                                                   IF(NNIT) 180,30,190
  180 WRITE (6,5030) K,KBIG,(AM(I),I=1,K)
COMPUTE NEW ELEMENTS FOR THIS ROW
  190 K1=NT1
      M1=NL2
      M2=LLM
      L=K
                              DO 210 J=1,K
                              DO 200 I=NL,K1
      A(I)=A(I)+AM(L)*A(M1)
  200 M1=M1+1
      K1=K1-1
      M1=M1-M2-1
      M2=M2-2
  210 L=L-1
C WRITE OUT ALTERED ROW IF REQUESTED
                                                   IF(NNIT) 220,30,230
  220 WRITE (6,5040) KBIG,(A(I),I=NL,NT)
C ATTEND TO R.H.S. IF NRHS NOT ZERO.
  230                                              IF(NRHS) 30,240,250
C WRITE ALTERED ROW ON TAPE (NOT) IF NRHS=0.
  240 WRITE (NOT) (A(I),I=NL,NT)
                                                           GO TO 320
C READ R.H.S. FROM TAPE (NIT)
  250 READ (NIT) (B(1,J),J=1,NRHS)
C WRITE OUT INPUT R.H.S. IF REQUESTED
                                                   IF(NNIT) 260,30,270
  260 WRITE (6,5020) KBIG,(B(1,J),J=1,NRHS)
COMPUTE NEW R.H.S
  270                         DO 280 J=1,NRHS
                              DO 280 I=1,K
```

```
  280 B(1,J)=B(1,J)+AM(I)*B(I+1,J)
C WRITE ALTERED ROW AND R.H.S. ON TAPE (NOT) IF NRHS   NOT 0.
      WRITE (NOT) (A(I),I=NL,NT),(B(1,J),J=1,NRHS)
C WRITE OUT ALTERED R.H.S. IF REQUESTED
                                                   IF(NNIT) 290,30,300
  290 WRITE (6,5050) KBIG,(B(1,J),J=1,NRHS)
C SHIFT R.H.S. DOWN
  300                      DO 310 J=1,NRHS
  310 B(K+2,J)=B(1,J)
C WRITE M(I,J) ON TAPE (NMT) IF REQUESTED
                                                                IERR=6
  320                                               IF(NMT) 30,340,330
  330 WRITE (NMT) (AM(I),I=1,K)
C SHIFT ALTERED DIAGONAL ELEMENT
  340 D(K+1)=A(NL)
C SHIFT ELEMENTS TOWARDS TOP OF TRIANGULAR ARRAY FOR NEXT ROW OPERATION
      K1=NDMP1-K
      I1=NDM-K
      M1=LLT-K
      M2=M1
      M1=M1*(M1+1)/2+1
      M2=M2+M1
                           DO 360 I=I1,NDM
                           DO 350 J=M1,M2
      L=K1+J
  350 A(J)=A(L)
      K1=K1+1
      M1=M2+1
  360 M2=M1+I
  370                      CONTINUE
C*****OPERATE ON ROWS (M-1)/2+1 TO N-(M-1)/2 (FULL BAND WIDTH) *********
  380                      K=0
  390                      K=K+1
      KBIG=KBIG+1
C READ ROW (M-1)/2+K FROM TAPE (NIT)
      READ (NIT) (A(I),I=NL,NT)
CHECK IF DIAG. ELEMENT IS ZERO
                                                                IERR=7
                                                  IF(A(NL)) 400,30,400
C WRITE OUT INPUT ROW IF REQUESTED
  400                                              IF(NNIT) 410,30,420
  410 WRITE (6,5010) KBIG,(A(I),I=NL,NT)
COMPUTE THE M(I,J)
  420 J=1
                           DO 430 I=1,NDM
      AM(I)=-A(J)/D(I)
  430 J=J+I
C WRITE OUT THE M(I,J) IF REQUESTED
                                                   IF(NNIT) 440,30,450
  440 WRITE (6,5030) NDM,KBIG,(AM(I),I=1,NDM)
COMPUTE NEW ELEMENTS FOR THIS ROW
  450 M1=0
       L=0
                           DO 460 I=NL,NT1
      L=L+1
      M1=M1+L
      M2=M1
                           DO 460 J=L,NDM
      A(I)=A(I)+AM(J)*A(M2)
  460 M2=M2+J
C WRITE OUT ALTERED ROW IF REQUESTED
```

```
                                                      IF(NNIT) 470,30,480
  470 WRITE (6,5040) KBIG,(A(I),I=NL,NT)
C ATTEND TO R.H.S. IF NRHS NOT ZERO.
  480                                                 IF(NRHS) 30,490,500
C WRITE ALTERED ROW ON TAPE (NOT) IF NRHS=0.
  490 WRITE (NOT) (A(I),I=NL,NT)
                                                               GO TO 580
C READ R.H.S. FROM TAPE (NIT)
  500 READ (NIT) (B(1,J),J=1,NRHS)
C WRITE OUT R.H.S. INPUT IF REQUESTED
                                                      IF(NNIT) 510,30,520
  510 WRITE (6,5020) KBIG,(B(1,J),J=1,NRHS)
COMPUTE NEW R.H.S.
  520                        DO 530 J=1,NRHS
                             DO 530 I=1,NDM
  530 B(1,J)=B(1,J)+AM(I)*B(I+1,J)
C WRITE ALTERED ROW AND R.H.S. ON TAPE (NOT) IF NRHS NOT ZERO
      WRITE (NOT) (A(I),I=NL,NT),(B(1,J),J=1,NRHS)
C WRITE OUT ALTERED R.H.S. IF REQUESTED
                                                      IF(NNIT) 540,30,550
  540 WRITE (6,5050) KBIG,(B(1,J),J=1,NRHS)
C SHIFT R.H.S. UP
  550                        DO 570 J=1,NRHS
                             DO 560 I=1,NDM1
  560 B(I+1,J)=B(I+2,J)
  570 B(NDMP1,J)=B(1,J)
C WRITE THE M(I,J) ON TAPE (NMT) IF REQUESTED
  580                                                  IF(NMT) 30,600,590
  590 WRITE (NMT) (AM(I),I=1,NDM)
C SHIFT DIAG. ELEMENTS FOR NEXT ROW OPERATION
  600                        DO 610 I=1,NDM1
  610 D(I)=D(I+1)
      D(NDM)=A(NL)
C SHIFT ELEMENTS TOWARDS TOP OF TRIANGULAR ARRAY FOR NEXT ROW OPERATION
      K1=2
      M1=1
      M2=1
                             DO 630 I=1,NDM
                             DO 620 J=M1,M2
      L=K1+J
  620 A(J)=A(L)
      K1=K1+1
      M1=M2+1
  630 M2=M1+I
                               IF(K-NNN) 390,640,30
C*****OPERATE ON LAST (M-1)/2 ROWS (NOT FULL BANDWIDTH) ****************
  640 LAST=NT
      ILA=NDMP1
                             DO 900 K=1,NDM
      KBIG=KBIG+1
      ILA=ILA-1
      LAST=LAST-1
C READ ROW N-(M-1)/2+K FROM TAPE (NIT)
      READ (NIT) (A(I),I=NL,LAST)
CHECK IF DIAGONAL ELEMENT IS ZERO
                                                                  IERR=8
                                                     IF(A(NL)) 650,30,650
C WRITE OUT INPUT ROW IF REQUESTED
  650                                                 IF(NNIT) 660,30,670
  660 WRITE (6,5010) KBIG,(A(I),I=NL,LAST)
COMPUTE THE M(I,J)
```

```
  670 J=1
                              DO 680 I=1,NDM
      AM(I)=-A(J)/D(I)
  680 J=J+I
C WRITE OUT THE M(I,J) IF REQUESTED
                                                                IERR=9
                                                    IF(NNIT) 690,30,700
  690 WRITE (6,5030) NDM,KBIG,(AM(I),I=1,NDM)
COMPUTE NEW ELEMENTS FOR THIS ROW
  700 M1=0
      L=0
                              DO 710 I=NL,LAST
      L=L+1
      M1=M1+L
      M2=M1
                              DO 710 J=L,NDM
      A(I)=A(I)+AM(J)*A(M2)
  710 M2=M2+J
C WRITE OUT ALTERED ROW IF REQUESTED
                                                    IF(NNIT) 720,30,730
  720 WRITE (6,5040) KBIG,(A(I),I=NL,LAST)
C ATTEND TO R.H.S. IF NRHS NOT ZERO
  730                                               IF(NRHS) 30,740,750
C WRITE ALTERED ROW ON TAPE (NOT) IF NRHS=0.
  740 WRITE (NOT) (A(I),I=NL,LAST)
                                                              GO TO 830
C READ R.H.S. FROM TAPE (NIT)
  750 READ (NIT) (B(1,I),I=1,NRHS)
C WRITE OUT INPUT R.H.S. IF REQUESTED
                                                    IF(NNIT) 760,30,770
  760 WRITE (6,5020) KBIG,(B(1,J),J=1,NRHS)
COMPUTE NEW R.H.S.
  770                         DO 780 J=1,NRHS
                              DO 780 I=1,NDM
  780 B(1,J)=B(1,J)+AM(I)*B(I+1,J)
C WRITE ALTERED ROW AND R.H.S. ON TAPE (NOT) IF NRHS NOT ZERO
      WRITE (NOT) (A(I),I=NL,LAST),(B(1,J),J=1,NRHS)
C WRITE OUT ALTERED R.H.S. IF REQUESTED
                                                    IF(NNIT) 790,30,800
  790 WRITE (6,5050) KBIG,(B(1,J),J=1,NRHS)
C SHIFT UP R.H.S.
  800                         DO 820 J=1,NRHS
                              DO 810 I=1,NDM1
  810 B(I+1,J)=B(I+2,J)
  820 B(NDMP1,J)=B(1,J)
C WRITE THE M(I,J) ON TAPE (NMT) IF REQUESTED
  830                                                IF(NMT) 30,850,840
  840 WRITE (NMT) (AM(I),I=1,NDM)
C SHIFT DIAGONAL ELEMENTS FOR NEXT ROW OPERATION (IF IT EXISTS)
  850                                              IF(K-NDM) 860,900,30
  860                         DO 870 I=1,NDM1
  870 D(I)=D(I+1)
      D(NDM)=A(NL)
C SHIFT ELEMENTS TOWARDS TOP OF TRIANGULAR ARRAY FOR NEXT ROW OPERATION
      K1=2
      M1=1
      M2=1
                              DO 890 I=1,NDM
                              DO 880 J=M1,M2
      L=K1+J
  880 A(J)=A(L)
```

```
      K1=K1+1
      M1=M2+1
  890 M2=M1+I
  900                      CONTINUE
C****************       END OF TRIANGULARIZATION             **************
C     ***********               BACK SUBSTITUTION               ***********
  920                                                 IF(NRHS) 30,1070,925
  925 KBIG=N+1
      BACKSPACE NOT
                           K=0
  930                      K=K+1
      KBIG=KBIG-1
                                                     IF(K-NDM) 934,934,935
  934 M2=K
      K2=K+1
  935                                              IF(K-NDMP1) 940,940,950
  940 LAST=K
      K1=LAST-1
  950                                                 IF(NRHS) 30,955,960
  955 READ (NOT) (A(I),I=1,LAST)
                                                              GO TO 970
  960 READ (NOT) (A(I),I=1,LAST),(B(1,J),J=1,NRHS)
COMPUTE UNKNOWNS
  970 BACKSPACE NOT
      BACKSPACE NOT
                           DO 1000 J=1,NRHS
                                                     IF(K-1) 30,1000,980
  980                      DO 990 I=1,K1
  990 B(1,J)=B(1,J)-B(I+1,J)*A(I+1)
 1000 B(1,J)=B(1,J)/A(1)
                                                   IF(NNIT) 1010,30,1020
C WRITE OUT SOLUTIONS IF REQUESTED
 1010 WRITE (6,5070) KBIG,(B(1,J),J=1,NRHS)
 1020                      DO 1030 J=1,NRHS
      M1=K2
                           DO 1030 I=1,M2
      B(M1,J)=B(M1-1,J)
 1030 M1=M1-1
C WRITE SOLUTIONS ON TAPE (NANST)
      WRITE (NANST) (B(1,J),J=1,NRHS)
                    IF(K-N) 930,1060,30
C****************           END OF BACK SUBSTTUTION           **************
 1060 REWIND NANST
 1070 RETURN
 5000 FORMAT(//16H  FAULTY DATA AT,1I4)
 5010 FORMAT(//12H  INPUT ROW ,1I5/(1P,4D25.15))
 5020 FORMAT(  26H  INPUT CONSTANTS FOR ROW ,1I5/(1P,4D25.15))
 5030 FORMAT(6H  THE ,1I5,'  COMPUTED M(I,J) FOR ROW',1I5/(1P,4D25.15))
 5040 FORMAT(  14H  ALTERED ROW ,1I5/(1P,4D25.15))
 5050 FORMAT(  28H  ALTERED CONSTANTS FOR ROW ,1I5/(1P,4D25.15))
 5070 FORMAT(/ 19H  COMPUTED UNKNOWN ,1I5/(1P,4D25.15))
      END
      SUBROUTINE VECTOR(NUM,N,NMAX,M,NROW,A)
C     ROWS DELETED TO SATISFY BOUNDARY CONDITION REPLACED BY ZEROS IN
C     VECTOR
      DIMENSION NROW(8),A(NMAX,1)
      M=N
      DO 1 K=1,NUM
      CALL BACK(NROW(K),N,M,NMAX,A)
      M=M+1
    1 CONTINUE
```

```
      RETURN
      END
      SUBROUTINE MODE(ISTRN,ISTRES,INR,SK,EPSIL,EVEC,TRANS,SO,K)
      PARAMETER(N300=791)
C     DEFLECTIONS, STRAINS, AND STRESSES COMPUTED, PRINTED AND PLOTTED
      COMMON /BLK/YOUNG1,XMU1,TH,YOUNG2,XMU2,G12,RO
      COMMON /STR/R1,R2,R1P,R,RP,C11,C12,C22,D11,D12,D22,K11,K12,K22
      DIMENSION X(N300),W(N300),WP(N300),WPP(N300),U(N300),UP(N300),
     1E1(N300),E2(N300),X1(N300),X2(N300),CE1(N300),CE1N(N300),
     2CE2(N300),CE2N(N300),T1(N300),T2(N300),XM1(N300),XM2(N300),
     3SIG1(N300),SIG1N(N300),SIG2(N300),SIG2N(N300),
     4TRANS(10,10),EVEC(1),A(10),SK(1),EPSIL(1)
      REAL K11, K12, K22
      CON1=YOUNG1/(1.-XMU1*XMU2)
      CON2=YOUNG2/(1.-XMU1*XMU2)
      VI=0
   31 IK=0
      REWIND 9
      EBEG=0.
      ELAST=EPSIL(1)
      I5=-1
      IFIRST=1
C     IK IS LOOP ON ELEMENT (K TOTAL ELEMENTS)
   40 IK=IK+1
      IF(IK.GT.K) GO TO 90
      IF(IK.EQ.1) GO TO 50
      EBEG=EBEG+EPSIL(IK-1)
      ELAST=EBEG+EPSIL(IK)
C     TRANSFORMATION MATRIX FOR ELEMENT IK READ FROM FILE 9
   50 READ(9)((TRANS(I,J),J=1,10),I=1,10)
      I5=I5+1
      I6 = 5*I5
      DO 10 I1=1,10
      A(I1) = 0.
      DO 10 I3=1,10
      I4=I6+I3
C     TRANSFORMATION MATRIX * PROPER BLOCK OF NUMBERS OF VECTOR GIVES
C     THE COEFFICIENTS
   10 A(I1) = A(I1) + TRANS(I1,I3) * EVEC(I4)
      IF (IK.NE.1) GO TO 70
      S=0.
      II=1
      GO TO 110
   70 EINT=ELAST-EBEG
      IFIRST=0
      DEL=EINT/FLOAT(INR)
      S=EBEG
      STT=-EINT/2.
      INRP=INR + 1
      DO 200 I=1,INRP
      S1=STT
      S2=S1*S1
      S3=S2*S1
      S4=S3*S1
      S5=S4*S1
      WW=A(1)+A(2)*S1+A(3)*S2+A(4)*S3+A(5)*S4+A(6)*S5
      CON =6.28318
      IF((I.EQ.1).OR.(I.EQ.INRP)) CON=3.14159
      CALL PEST (2,0,S1,R,IK)
      STT=STT+DEL
  200 VI = VI + CON*WW*R*DEL
```

```
      WRITE(6,1010) IK, VI
   30 S=S+DEL
      IF(S.GT.ELAST) GO TO 20
      II=II+1
  110 S1=S-(SK(IK)-S0)
      S2=S1**2
      S3=S1*S2
      S4=S1*S3
      S5=S1*S4
C     MODE SHAPES
      W(II)=A(1)+A(2)*S1+A(3)*S2+A(4)*S3+A(5)*S4+A(6)*S5
      WP(II)=A(2)+2.*A(3)*S1+3.*A(4)*S2+4.*A(5)*S3+5.*A(6)*S4
      WPP(II)=2.*A(3)+6.*A(4)*S1+12.*A(5)*S2+20.*A(6)*S3
      U(II)=A(7)+A(8)*S1+A(9)*S2+A(10)*S3
      UP(II)=A(8)+2.*A(9)*S1+3.*A(10)*S2
      X(II)=S
C     STRAINS
      IF(ISTRN.EQ.0) GO TO 60
      ARG=SK(IK)-EPSIL(IK)/2.+S
      CALL PEST (4,0,S1,RR,IK)
      E1(II)=UP(II)+W(II)*R1
      E2(II) = RP*U(II)/R+W(II)*R2
      X1(II)=-WPP(II)+UP(IT)*R1-U(II)*R1P*R1**2
      X2(II) = (-RP*WP(II)+RP*U(II)+R1)/R
      CE1(II)=(E1(II)+.5*TH*X1(II))/(1.+.5*TH*R1)
      CE1N(II)=(E1(II)-.5*IH*X1(II))/(1.-.5*TH*R1)
      CE2(II)=(E2(II)+.5*TH*X2(II))/(1.+.5*TH*R2)
      CE2N(II)=(E2(II)-.5*TH*X2(II))/(1.-.5*TH*R2)
C     STRESSES
      IF(ISTRES.EQ.0) GO TO 60
      SIG1(II)=CON1*(CE1(II)+XMU2*CE2(II))
      SIG1N(II)=CON1*(CE1N(II)+XMU2*CE2N(II))
      SIG2(II)=CON2*(CE2(II)+XMU1*CE1(II))
      SIG2N(II)=CON2*(CE2N(II)+XMU1*CE1N(II))
      T1(II)=C11*E1(II)+C12*E2(II)+K11*X1(II)+K12*X2(II)
      T2(II)=C12*E1(II)+C22*E2(II)+K12*X1(II)+K22*X2(II)
      XM1(II)=D11*X1(II)+D12*X2(II)+K11*E1(II)+K12*E2(II)
      XM2(II)=D12*X1(II)+D22*X2(II)+K12*E1(II)+K22*E2(II)
   60 IF(IFIRST.EQ.1) GO TO 70
      GO TO 30
   20 CONTINUE
      GO TO 40
   90 CONTINUE
      II2 = 0
      WRITE(6,1001)
      DO 80 I=1,II
   80 WRITE(6,1002)X(I),W(I),U(I)
      IF(ISTRN.EQ.0) GO TO 100
      WRITE(6,1003)
      DO 160 I=1,II
  160 WRITE(6,1004)X(I),E1(I),E2(I),X1(I),X2(I)
      WRITE(6,1005)
      DO 170 I=1,II
  170 WRITE(6,1004)X(I),CE1(I),CE1N(I),CE2(I),CE2N(I)
      IF(ISTRES.EQ.0) GO TO 100
      WRITE(6,1007)
      DO 190 I=1,II
  190 WRITE(6,1004)X(I),SIG1(I),SIG1N(I),SIG2(I),SIG2N(I)
      WRITE(6,1006)
      DO 180 I=1,II
  180 WRITE(6,1004)X(I),T1(I),T2(I),XM1(I),XM2(I)
```

```
 100 CONTINUE
1010 FORMAT (' CUMULATIVE VOLUME CHANGE THRU SEGMENT ',I2, 1H=,
    1E16.8)
1001 FORMAT(1H1///22X,10HMODE SHAPE//12X,1HX,19X,1HW,19X,1HU)
1002 FORMAT(4(4X,F16.8))
1003 FORMAT(1H1///25X,'MIDDLE SURFACE STRAINS AND CHANGES IN CURVATURE'
    1 //10X,1HX,17X,2HE1,16X,2HE2,15X,2HX1,16X,2HX2)
1004 FORMAT(7(2X,E16.8))
1005 FORMAT(1H1///41X,'EXTREME FIBER STRAINS' //
    1                                              10X,1HX,12X,10HE1POSITIVE
    2,7X,11HE1 NEGATIVE ,7X,11HE2 POSITIVE ,7X,11HE2 NEGATIVE)
1006 FORMAT(1H1///35X,28HSTRESS AND MOMENT RESULTANTS
    1//10X,1HX,16X,2HT1,16X,2HT2,12X,2HM1,16X,2HM2)
1007 FORMAT(1H1///40X,22HEXTREME FIBER STRESSES
    1//10X,1HX,12X,11HSIGMA SUB 1,7X,11HSIGMA SUB 1,7X,11HSIGMA SUB 2,
    27X,11HSIGMA SUB 2/24X,10H(POSITIVE)
    3 ,8X,10H(NEGATIVE),8X,10H(POSITIVE),8X,10H(NEGATIVE))
     RETURN
     END
     SUBROUTINE BACK(NE,N,M,NMAX,A)
C    ZERO INSERTED INTO PROPER ROW OF VECTOR
     DIMENSION A(NMAX,1)
     MP1=M+1
     IF(NE.GT.1) GO TO 30
         J = 1
     DO 10 I=2,MP1
     II=MP1+2-I
  10 A(II,J)=A(II-1,J)
  20 A(1,J)=0.
     RETURN
  30 IF(NE.NE.MP1) GO TO 50
            J = 1
  40 A(MP1,J)=0.
     RETURN
  50 NEP1=NE+1
     J = 1
     DO 60 I=NEP1,MP1
     II=MP1+NEP1-I
  60 A(II,J)=A(II-1,J)
  70 A(NE,J)=0.
     RETURN
     END
     SUBROUTINE CASE(ICASE,K,NELIM,NUM)
C    ROW AND COLUMN NUMBERS TO BE DELETED TO SATISFY BOUNDARY CONDITION
C    STORED IN ARRAY NELIM  (MAXIMUM OF 8 NUMBERS)
     DIMENSION NELIM(8)
     NUM=0
     IF(ICASE.EQ.17) GO TO 40
     IF(ICASE.EQ.18) GO TO 40
     IF(ICASE.EQ.4) GO TO 20
     IF(ICASE.EQ.6) GO TO 30
     NELIM(1)=1
     NUM=NUM+1
     IF(ICASE.EQ.11) GO TO 20
     IF(ICASE.EQ.12) GO TO 30
     NELIM(2)=2
     NUM=NUM+1
     IF(ICASE.EQ.5) RETURN
     IF(ICASE.EQ.9) GO TO 20
     IF(ICASE.EQ.13) GO TO 10
     IF(ICASE.EQ.14) GO TO 30
```

```
      NELIM(3)=3
      NUM=NUM+1
      IF(ICASE.EQ.7) RETURN
      IF(ICASE.EQ.10) GO TO 30
      IF(ICASE.EQ.15) GO TO 10
      IF(ICASE.EQ.16) GO TO 20
   10 NELIM(NUM+1) = 5*K+1
      NUM = NUM+1
      RETURN
   20 DO 1 I=1,2
    1 NELIM(NUM+I)=5*K+I
      NUM=NUM+2
      RETURN
   30 DO 2 I=1,3
    2 NELIM(NUM+I)=5*K+I
      NUM=NUM+3
      RETURN
   40 NUM=2
      NELIM(1)=2
      NELIM(2)=3
      IF(ICASE.EQ.18) GO TO 50
      GO TO 30
   50 NUM=4
      NELIM(3)=5*K+2
      NELIM(4)=5*K+3
      RETURN
      END
```

# APPENDIX D

## SAMPLE INPUT

```
//HLAK196T JOB (6ED553590034),CONVERSIONS,CLASS=X,MSGLEVEL=(1,1),
//    TIME=0025
//SPARD PROC P=P
//   EXEC   PGM=&P,REGION=4000K,COND=(4,LT)
//STEPLIB  DD  DSNAME=HLAK196.SPAR.LOAD,DISP=SHR
//FT05F001  DD DDNAME=SYSIN
//FT06F001    DD  SYSOUT=X
//FT07F001  DD DUMMY
//FT09F001  DD DSNAME=HLAK196.NAS9.DATA,DISP=SHR
//FT11F001  DD DSNAME=HLAK196.NAS11.DATA,DISP=SHR
//FT12F001  DD DSNAME=HLAK196.NAS12.DATA,DISP=SHR
//FT13F001  DD DSNAME=HLAK196.NAS13.DATA,DISP=SHR
//FT14F001  DD DSNAME=HLAK196.NAS14.DATA,DISP=SHR
// PEND
//STEPY EXEC SPARD,P=SHELL
//SYSIN    DD *
 BELLOWS RECOMPILATION CHECK 1/89
  17   1   0   0
 0.              2.0
    1    5     .470          2.843        .326       0.
    2    4     .608          2.884        .125       -.99216
    1   10     .760          2.312       -.263       -.380
    2    4     .608          2.279        .125        .99216
    1   10     .940          2.843        .326       -.470
    2    4     .608          2.884        .125       -.99216
    1   10     .760          2.312       -.263       -.380
    2    4     .608          2.279        .125        .99216
    1   10     .982          2.843        .326       -.470
    1    8    1.253          2.843       -.798      -1.253
    2   10    1.5            2.045       1.           .0
       .28E+08        .28E+08  .3          .3        50.
     .037          .1077E+08
/*
//
```

(NOTE: FOR IBM, FORTRAN FILE 12 MUST BE SEQUENTIAL, OTHERS DIRECT ACCESS.)

APPROVAL

# PRESSURE-VOLUME PROPERTIES OF METALLIC BELLOWS

By Larry Kiefling

The information in this report has been reviewed for technical content. Review of any information concerning Department of Defense or nuclear energy activities or programs has been made by the MSFC Security Classification Officer. This report, in its entirety, has been determined to be unclassified.

JAMES C. BLAIR
Director, Structures and Dynamics Laboratory

☆U.S. GOVERNMENT PRINTING OFFICE: 1989 - 631-056/00105