STRESSES AND FLEXIBILITIES FOR PRESSURE VESSEL ATTACHMENTS

by

FRANK MICHAEL GERARD WONG

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS OF THE
DEGREES OF

BACHELOR OF SCIENCE IN NUCLEAR ENGINEERING

and

MASTER OF SCIENCE IN NUCLEAR ENGINEERING

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1984

Signature of Author ______________________________
Department of Nuclear Engineering
May 11, 1984

Certified by ______________________________
Professor John E. Meyer
Thesis Supervisor

Accepted by ______________________________
Professor Allan F. Henry
Departmental Graduate Committee
Department of Nuclear Engineering

STRESSES AND FLEXIBILITIES FOR PRESSURE VESSEL ATTACHMENTS

by

FRANK MICHAEL GERARD WONG

Submitted to the Department of Nuclear Engineering on
May 11, 1984 in partial fulfillment of the requirements for
the Degrees of Bachelor of Science in Nuclear Engineering
and Master of Science in Nuclear Engineering

ABSTRACT

The Fourier analysis method, developed primarily by Bijlaard, modelled pressure vessels as continuous simply-connected surfaces with central patch loads. Through the use of superposition, my work has extended the technique to include solutions for loads placed at any position along the cylinder. The technique allows placement of tractions with combinations of radial, shear, and axial components. In addition, the stiffness and enhanced load carrying capacity that internal pressure gives to thin vessels can be simulated. Numerical convergence problems encountered in this method are reduced by an improved displacement-load algorithm, and by using load sites that allow the circular functions to be compactly grouped.

Using this approach, a variety of loading distributions are analyzed including large and small nozzles near and away from centerlines. Both rectangular and circular attachments are simulated. When large attachments are involved, local lateral loads are modelled as a combination of tractions along the intersecting boundaries. Stiff attachments are simulated through a set of loads along the vessel/attachment intersection determined by colocation. These loads are chosen so that a rigid body displacement field is produced within that boundary. Through this same technique, multiple attachments with their own loads may be examined. The attachments to the vessel may be either rigid or soft.

In comparison with experimental results, the calculated values are in generally good agreement. Variations between experimental and calculated results are primarily caused by assuming a simply-supported base in the calculation, whereas in the experimental test, the base is more nearly fixed. The stiffening effect of the welds on an attachment was also shown to cause discrepancies between experimental and calculated values.

Thesis Supervisor: Dr. John E. Meyer

Title: Professor of Nuclear Engineering

DEDICATION

This work is dedicated to my mother, Shirley. Without her years of generous care, faith, and understanding, none of this would have been possible.

ACKNOWLEDGEMENTS

This thesis was performed at Stone & Webster Engineering Corporation as part of the M.I.T. Engineering Internship Program. I would like to properly acknowledge the help and assistance of a few individuals.

I am very grateful to Professor John E. Meyer for his invaluable insight, guidance, and encouragement as my thesis advisor, as well as my advisor during my undergraduate years.

My sincere thanks are due to Dr. William J. Craft for his indispensable advice, patience, and support. His contributions in creating some of the early source code are greatly appreciated.

I would also like to thank Dr. George H. East, Jr. for his help and direction during my work assignments at Stone & Webster.

The assistance of George Bushnell, my supervisor at Stone & Webster, is gratefully appreciated.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 SCOPE

This thesis describes a method to calculate vessel stresses and flexibilities for attachments on cylindrical vessels. Determining deformations, stresses, and flexibilities in vessels due to external loadings has not been an easy task. In some cases, finite element analysis may be used, but it is neither convenient nor economical. It has primarily been used for geometries beyond the scope of the Welding Research Council method (Ref. 1), which is the most commonly used approach.

The method presented here extends and advances Bijlaard's technique (Ref. 2 and 3) for determining stresses at attachment-shell junctions through double-sum Fourier analysis. These Fourier series are constructed so that they fulfill the boundary conditions of simple support and the equilibrium equations of thin-shell theory. The major features include:

- Ability to place attachments anywhere on the cylindrical shell, even near the end;
- Interaction between multiple attachments;
- Incorporation of any combination of external forces and moments, as well as internal pressure;
- Incorporation of rectangular and circular attachments, both solid and tubular;
- Designation of large or small attachments, as well as rigid or soft ones; and
- Evaluation of stress at more than the four cardinal points on intersection provided by finite element analysis and the Welding Research Council method.

The method employs the following assumptions:

- Vessel is a right circular cylinder of constant thickness;
- Cylindrical surface is simply-connected (i.e. there is a continuous shell at all attachments);

- Vessel circumference is simply supported at each end (i.e. zero radial and circumferential deflection, zero shell axial moment and axial normal force); and

- No thermal expansions are considered

By specifying shell dimensions and loading conditions, the method is able to compute deflections and stresses in the cylindrical shell.

1.2 BACKGROUND

1.2.1 BIJLAARD'S WORK

Two of Bijlaard's papers (Ref. 2 and 3) were used extensively to develop the background theory for this project. "Stresses Fom Local Loadings in Cylindrical Pressure Vessels" presents the basic theory for analyzing stresses at attachment-shell intersections through Fourier analysis. By starting with the partial differential equations of thin-shell theory, which govern the displacements in the X, Y(ø), and Z directions for a given pad load P, Bijlaard reduces this system of three partial differential equations into a single eigth-order equation. This single eighth-order differential equation is known as Donnell's equation. However, in deriving Donnell's equation, Bijlaard neglected terms that contained $t^2/12a^2$, because in considering terms of t/a (thickness-to-radius) smaller than 0.1, these $t^2/12a^2$ terms are less than 0.001. As a result of this approximation, Bijlaard says that results may be up to 25 per cent too small. Greater discrepancies occur for shells with larger thickness-to-radius ratios.

Bijlaard now proceeds to express the displacements and loads as double-sum Fourier series that are functions of the axial coordinate, x, and the circumferential coordinate, ø. Having employed this solution technique, Bijlaard solves the equation for several loading conditions which include external radial forces and longitudinal and circumferential moments.

In his second paper, "Stresses From Radial Load and External Moments In Cylindrical Pressure Vessels," Bijlaard presents some numerical data for his solution for many load cases due to radial loads and external moments, including the effects of internal pressure.

There are two essential shortcomings in Bijlaard work: the omission of higher order terms in the governing differential equation and the restricted positions of load application. As mentioned earlier, the omission of higher order terms in the differential equation will lead to discrepancies in results, unless shells are sufficiently thin. Bijlaard's solution was also restricted to locations that were far removed from the ends of the cylinder. As a result, large discrepancies will occur as one moves the location of the attachment toward the end of a finite cylinder. The thesis work includes provisions to eliminate these inaccuracies.

1.2.2 WELDING RESEARCH COUNCIL BULLETIN 107

To solve attachment-shell problems, industry presently employs a technique that is based on Welding Research Council Bulletin 107, "Local Stresses in Spherical and Cylindrical Shells due to External Loadings," by Wichman, Hopper, and Mershon (Ref. 1). This bulletin contains a "lookup graph" method for calculating these stresses due to external loadings. It contains Bijlaard's curves for these stresses, which have been plotted with a given set of dimensionless parameters.

In short, the method that Bulletin 107 presents is quite straight-forward. Given an initial set of parameters of a shell problem with external loadings, one can look up the values for the bending and membrane forces, M and N respectively. One also needs to look up the values of two stress concentration factors, Kn and Kb, which are a function of the attachment dimensions. Once these values are known, the stresses are calculated through a typical compound stress formula:

$$\sigma = Kn(N/T) \pm Kb(6M/T^2)$$

The procedure in the WRC Bulletin 107 contains the same shortcomings as Bijlaard's original work. In addition, no provision is made for obtaining deflections. As a result, one cannot use this method for calculating stiffnesses. It also has no capability to analyze interaction between multiple attachments. Because the curves in this bulletin have been plotted for only a given set of initial parameters, any data which does not lie on these curves must be interpolated or extrapolated to the nearest curve.

The thesis addresses the pitfalls in Bulletin 107. Since the thesis work does calculate deflections, stiffnesses as well as stresses are obtained. In addition, because the thesis uses a closed-form solution, interpolation and extrapolation is not necessary.

2. SOLUTION FUNDAMENTALS

2.1 GEOMETRY

Coordinate System

The vessel is modeled as a right-circular-cylindrical shell (See Figure 2.1). A conventional orthogonal coordinate system is used to define the cylindrical geometry (cylinder has radius a, thickness t, and length L):

- Coordinates:

  x = axial coordinate $0 \leq x \leq L$

  ϕ = circumferential coordinate $0 \leq \phi \leq 2\pi$

- Displacements:

  u = axial displacement

  v = circumferential displacement

  w = radial displacement (positive in the negative r-direction)

- Surface loadings (force per unit area):

  P_x = axial loading (shear)

  P_ϕ = circumferential loading (shear)

  P_r = radial loading (positive radially inward)

Boundary Conditions

The cylinder is assumed to be simply-supported. The boundary conditions for simple support are:

At $x = 0$ and L : $w = 0$

$v = 0$

Moment $M_x = 0$

Membrane force $N_x = 0$

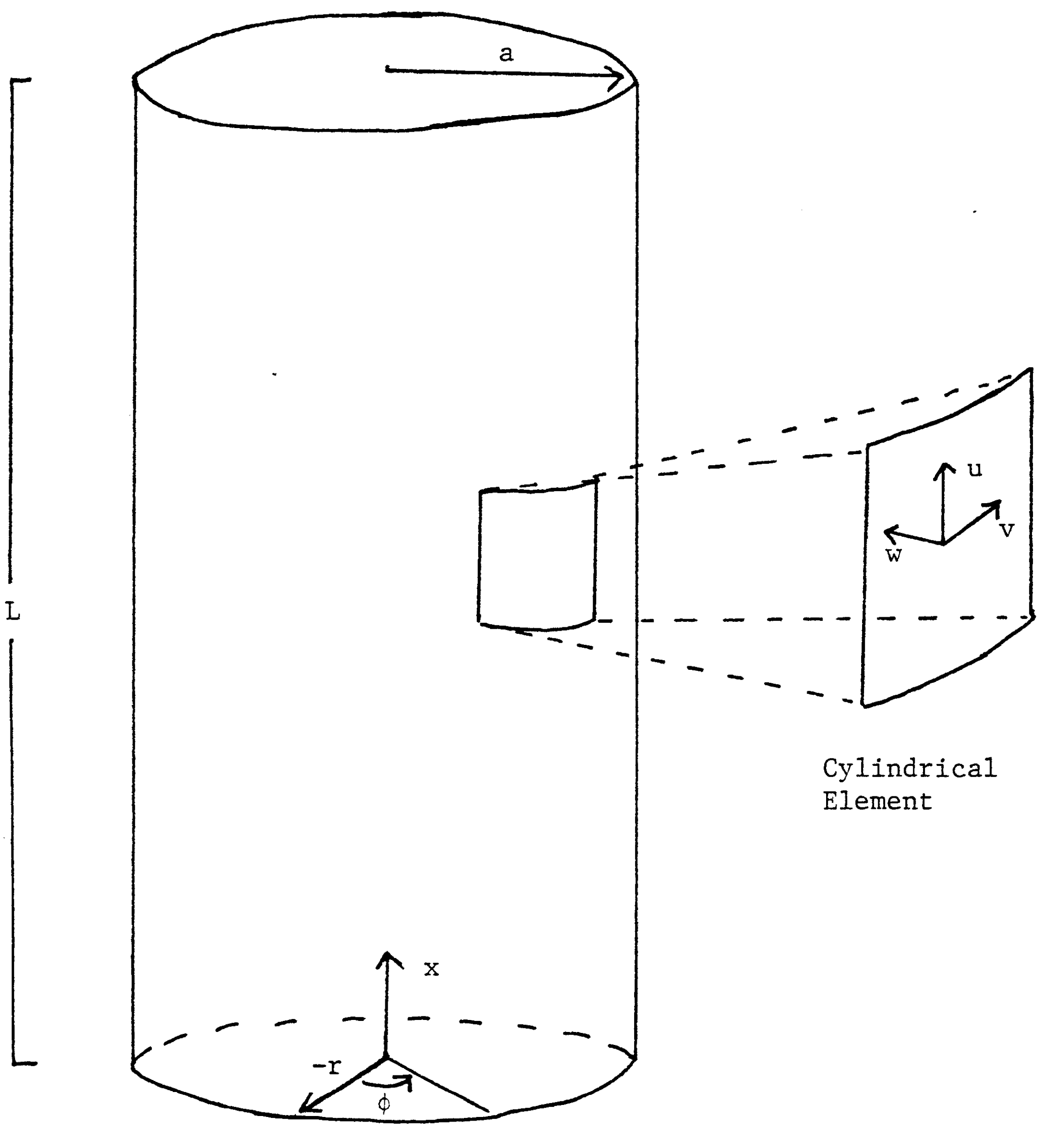


Figure 2.1 Cylindrical Geometry and Coordinate Definition

Equilibrium Equations

The governing equations for the displacements in thin-walled circular cylindrical shells under general loading are (Ref. 4):

$$\frac{\partial^2 u}{\partial x^2} + \frac{1 - \nu}{2a^2}\frac{\partial^2 u}{\partial \phi^2} + \frac{1 + \nu}{2a}\frac{\partial^2 v}{\partial x\, \partial \phi} - \frac{\nu}{a}\frac{\partial w}{\partial x} = -P_x\frac{(1 - \nu^2)}{Et} \qquad (2.1a)$$

$$\frac{1 + \nu}{2a}\frac{\partial^2 u}{\partial x\, \partial \phi} + \frac{1 - \nu}{2}\frac{\partial^2 v}{\partial x^2} + \frac{1}{a^2}\frac{\partial^2 v}{\partial \phi^2} - \frac{1}{a^2}\frac{\partial w}{\partial \phi} + \frac{t^2}{12a^2}\left[\frac{\partial^3 w}{\partial x^2\, \partial \phi} + \frac{1}{a^2}\frac{\partial^3 w}{\partial \phi^3}\right] + \frac{t^2}{12a^2}\left[(1 - \nu)\frac{\partial^2 v}{\partial x^2} + \frac{1}{a^2}\frac{\partial^2 v}{\partial \phi^2}\right] = -P_\phi\frac{(1 - \nu^2)}{Et} \qquad (2.1b)$$

$$\frac{\nu}{a}\frac{\partial u}{\partial x} + \frac{1}{a^2}\frac{\partial v}{\partial \phi} - \frac{w}{a^2} - \frac{t^2}{12}\left[\frac{\partial^4 w}{\partial x^4} + \frac{2}{a^2}\frac{\partial^4 w}{\partial x^2\, \partial \phi^2} + \frac{1}{a^4}\frac{\partial^4 w}{\partial \phi^4}\right] - \frac{t^2}{12a^2}\left[\frac{\partial^3 v}{\partial x^2\, \partial \phi} + \frac{1}{a^2}\frac{\partial^3 v}{\partial \phi^3}\right] = -P_r\frac{(1 - \nu^2)}{Et} \qquad (2.1c)$$

Attachment Geometry

The area of intersection between the attachment and the cylindrical shell is modeled as an area partitioned into rectangular and triangular elements.

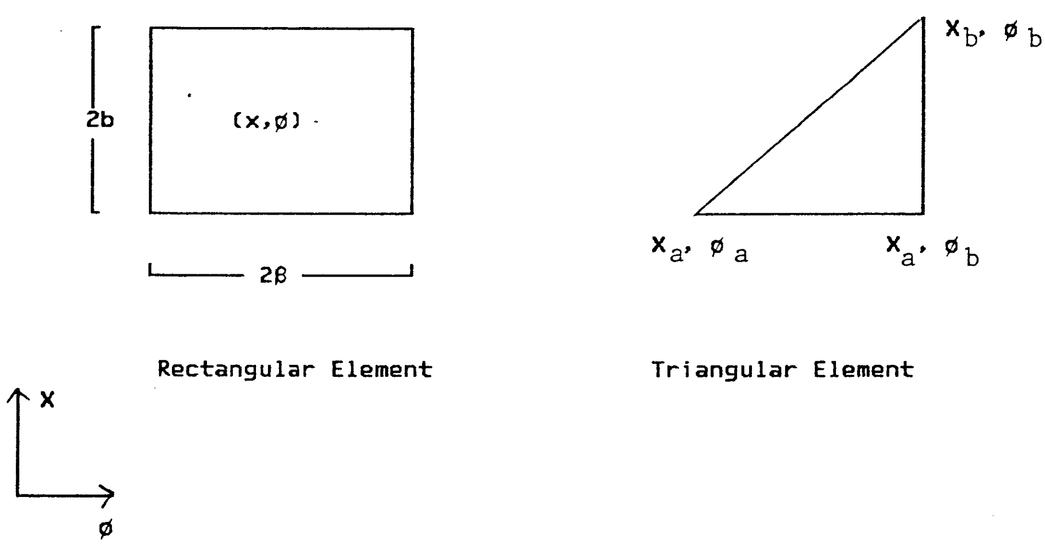


Figure 2.2 Rectangular and Triangular Elements for the Area of Intersection Between Attachment and Cylindrical Shell

Through a combination of rectangular elements, a solid or hollow rectangular attachment may be modeled. Through a combination of rectangular and triangular elements, a circular pipe or solid may be simulated.

2.2 ELEMENT PARTITIONING OF ATTACHMENT

Rectangular Attachments

For rectangular attachments, the area of intersection between the attachment and cylindrical shell is split into four regions. The partitioning of the this area is determined by the initial dimensions of the attachment.

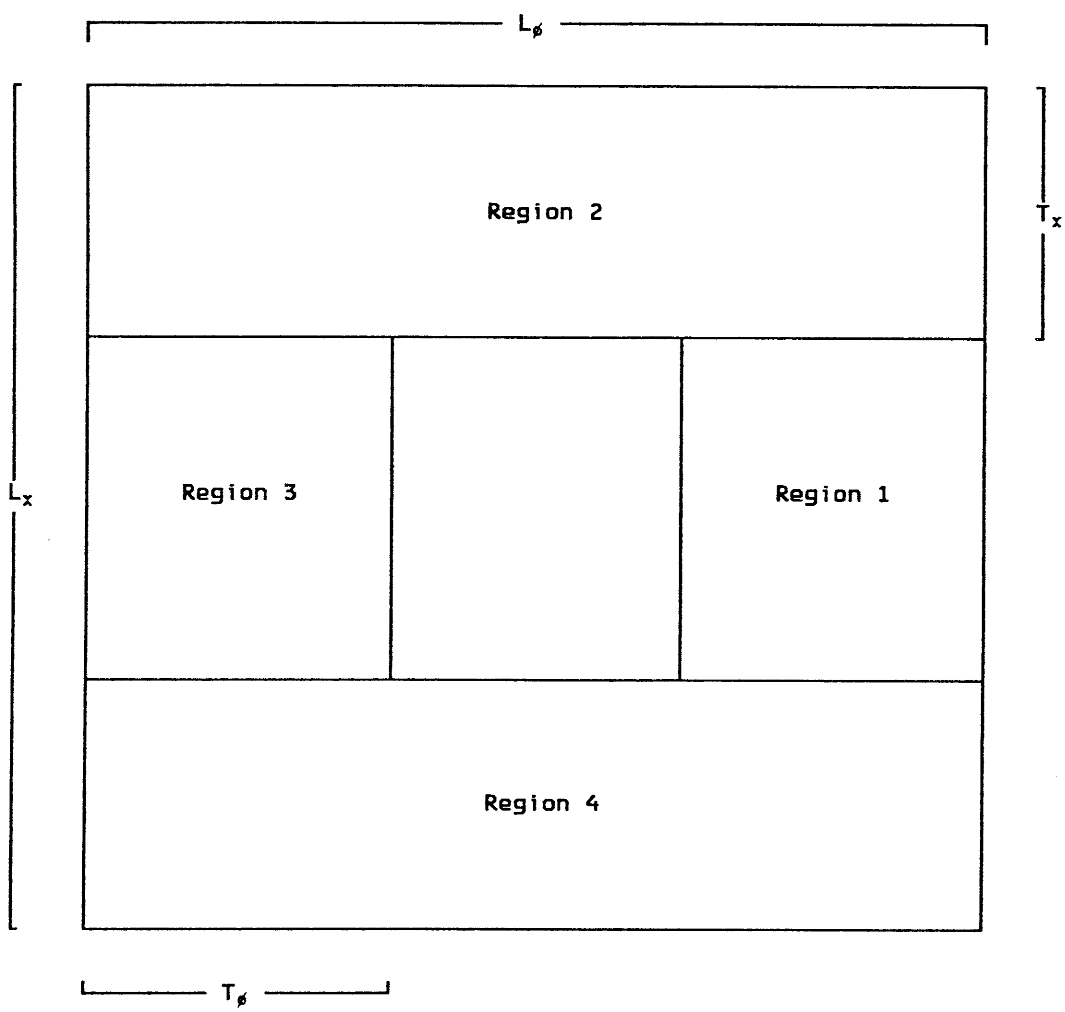


Figure 2.3 Patch Area Partition of a General Rectangular Tube.

A solid rectangular attachment may be modelled when $T_\phi = L_\phi/2$ and $T_x = L_x/4$. This partition is shown in Figure 2.4.

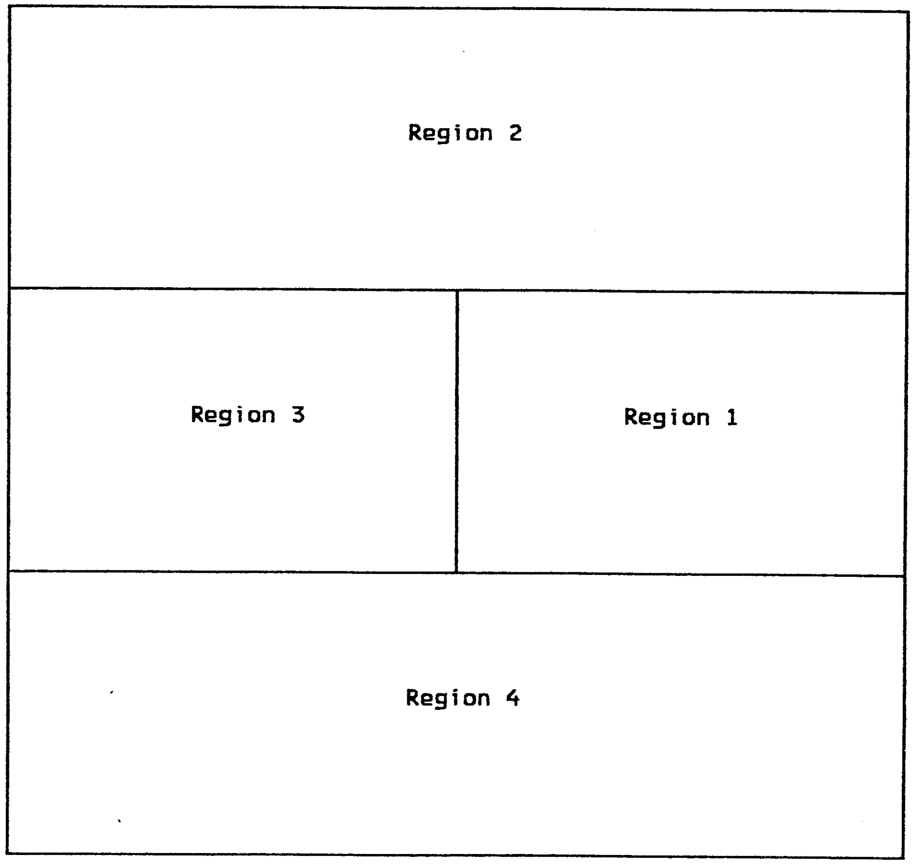


Figure 2.4 Partition of a Solid Rectangular Attachment

These four patch areas are further divided into elements. The maximum subdivision of elements is a 4 x 4 partition in each region, and the minimum partition is 1 x 1. Figure 2.5 illustrates the element subdivision for a 3 x 3 partition. The surface loadings are applied on these individual elements to simulate a uniform radial force, longitudinal and circumferential moments, and a rigid plug displacement.

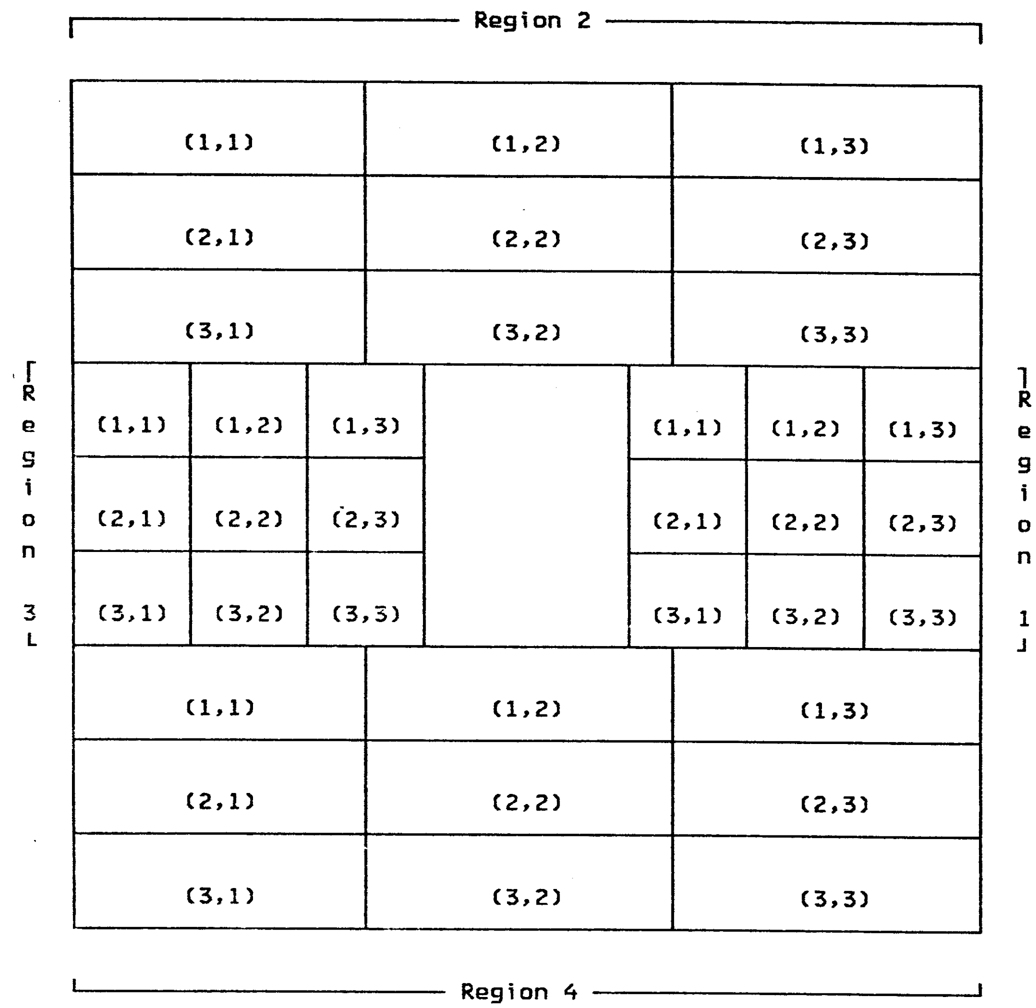


Figure 2.5 3 x 3 Element Partitioning of a Hollow Rectangular Attachment

Circular Attachments

For circular attachments, the area of intersection between the attachment and cylindrical shell is split into four quadrants. Each quadrant is then split into rectangles and triangles to approximate one quarter of a circle (See Figures 2.6 and 2.7). Each circular solid is split into 12 elements, and each circular tube is split into 28 elements.

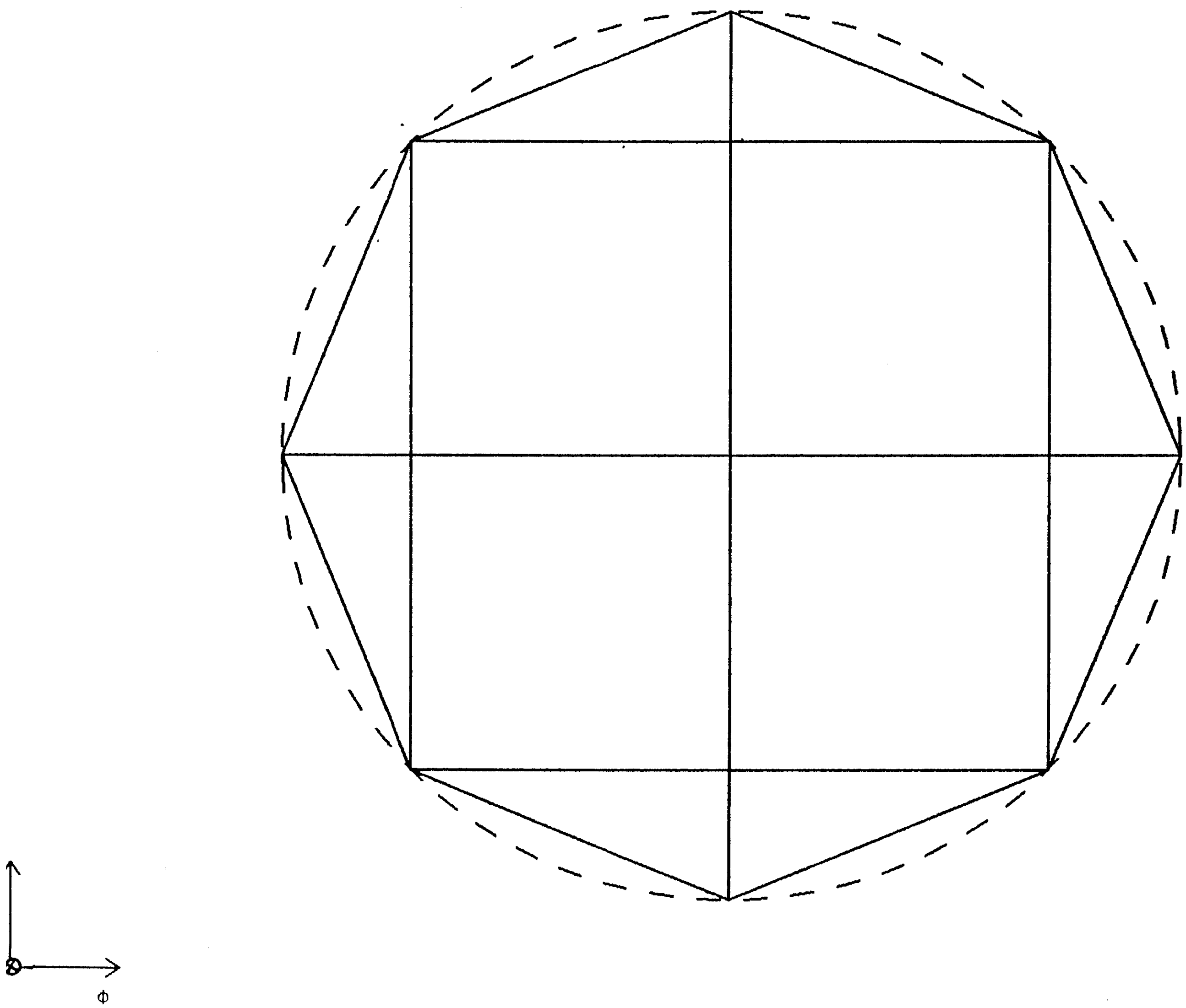


FIGURE 2.6 ELEMENT PARTITION OF CIRCULAR SOLID

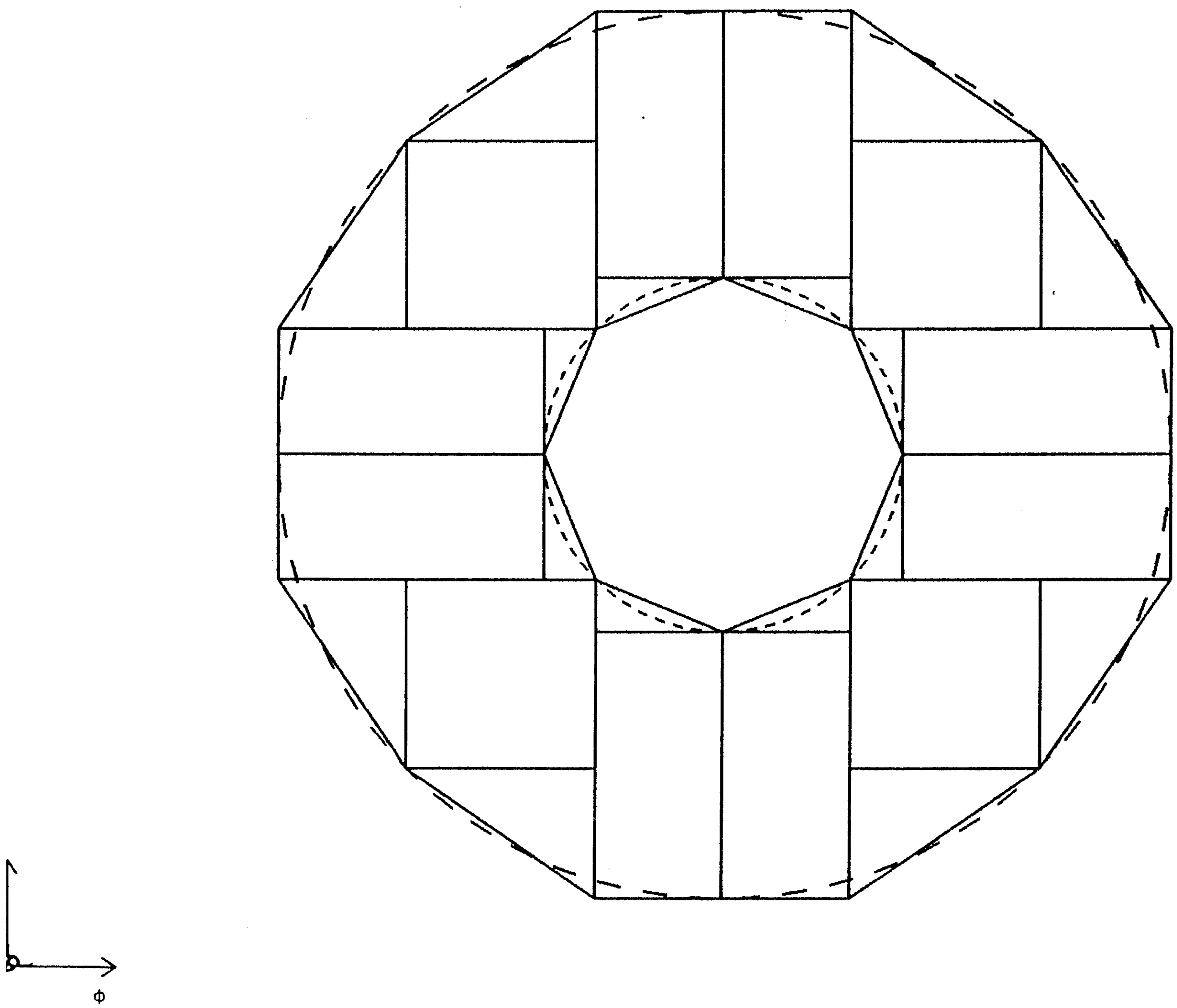


FIGURE 2.7 ELEMENT PARTITION OF CIRCULAR TUBE

3. LOADS AND MOMENTS

3.1 RADIAL LOADS

A radial load is simulated so that all elements of force are parallel to the centerline of the attachment. The user-specified force on the attachment is divided by the attachment area to give a pressure; this pressure is then applied uniformly over all elements in the attachment-shell intersection. This loading is decomposed into Fourier harmonics. The analysis unrolls the cylindrical surface, and then applies the P_r and P_ϕ everywhere inside the individual elements.

If the attachment-shell intersecting area is large compared to the total circumference of the cylinder, then shear loads are added to ensure that the applied loads are parallel to the attachment centerline.

Since all the loads are represented as Fourier series, superposing loads only requires adding the specified series together before the displacement solution is found.

3.2 EXTERNAL BENDING MOMENTS

A bending moment is represented as a series of radial loads distributed over the patch area. The magnitude of these loads varies as a linear function of their distance from the neutral axis. The magnitude of the loads is zero at the neutral axis, and increases linearly farther from the neutral axis. A longitudinal moment acts along the axis of the attachment and varies linearly in the x-direction of the shell. A circumferential moment also acts along the axis of the attachment, but varies linearly in the ϕ-direction of the shell.

For a longitudinal moment, the pressure loads assume the function

$$\sigma = \frac{Mx'}{I_x}$$

Where, x' = Axial distance of an element to the neutral axis

I_x = Moment of inertia of the actual attachment about the ϕ-axis

M = Externally applied longitudinal moment

For a circumferential moment, the pressure loads assume the function

$$\sigma = \frac{M\phi'}{I_\phi}$$

Where, ϕ' = Circumferential distance of an element to the neutral axis. Since this distance must be projected onto a flat surface, it reads as a function of the angle as ϕ' = Rsin(angular distance from neutral axis)

I_ϕ = Moment of inertia of the actual attachment about the x-axis

M = Externally applied circumferential moment

Figures 3.2A and 3.2B illustrate the interpretation of a moment.

3.3 MULTIPLE ATTACHMENTS

Multiple attachment can be easily simulated with Fourier analysis. Since the displacements over the entire cylinder may be determined from any given loading condition, multiple attachments involve superposing a given number of loading conditions. The Fourier series for each independent loading condition is developed. Then, these individual Fourier series are superposed on one another. Since the origins of these Fourier series are the same, superposing these series only involves adding the series together term by term. This produces a new Fourier series that contains multiple loading conditions. It is this new Fourier series that is used in determining the displacements over the cylindrical surface. The displacements that result will reflect the

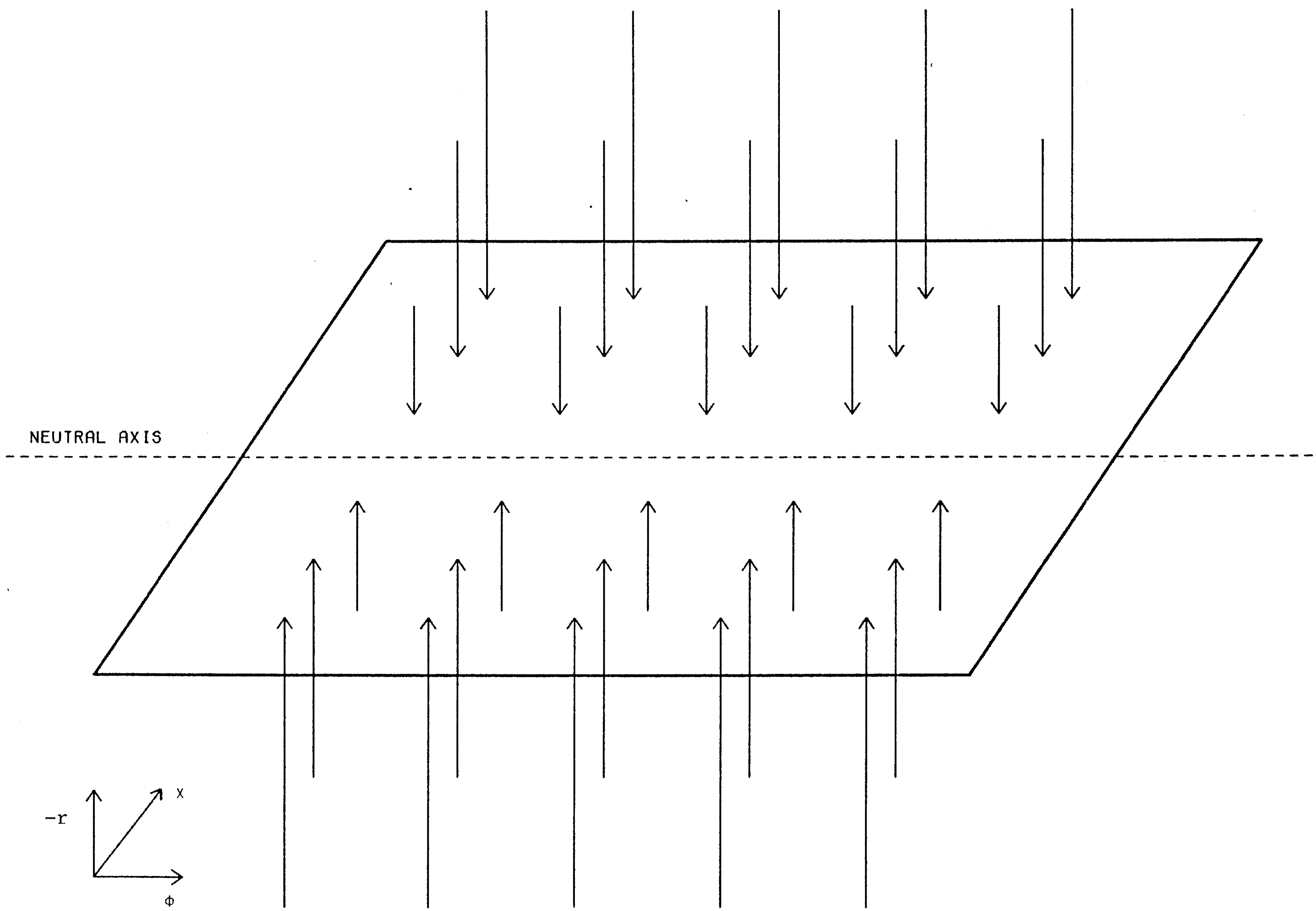


FIGURE 3.2A EXTERNAL LONGITUDINAL MOMENT

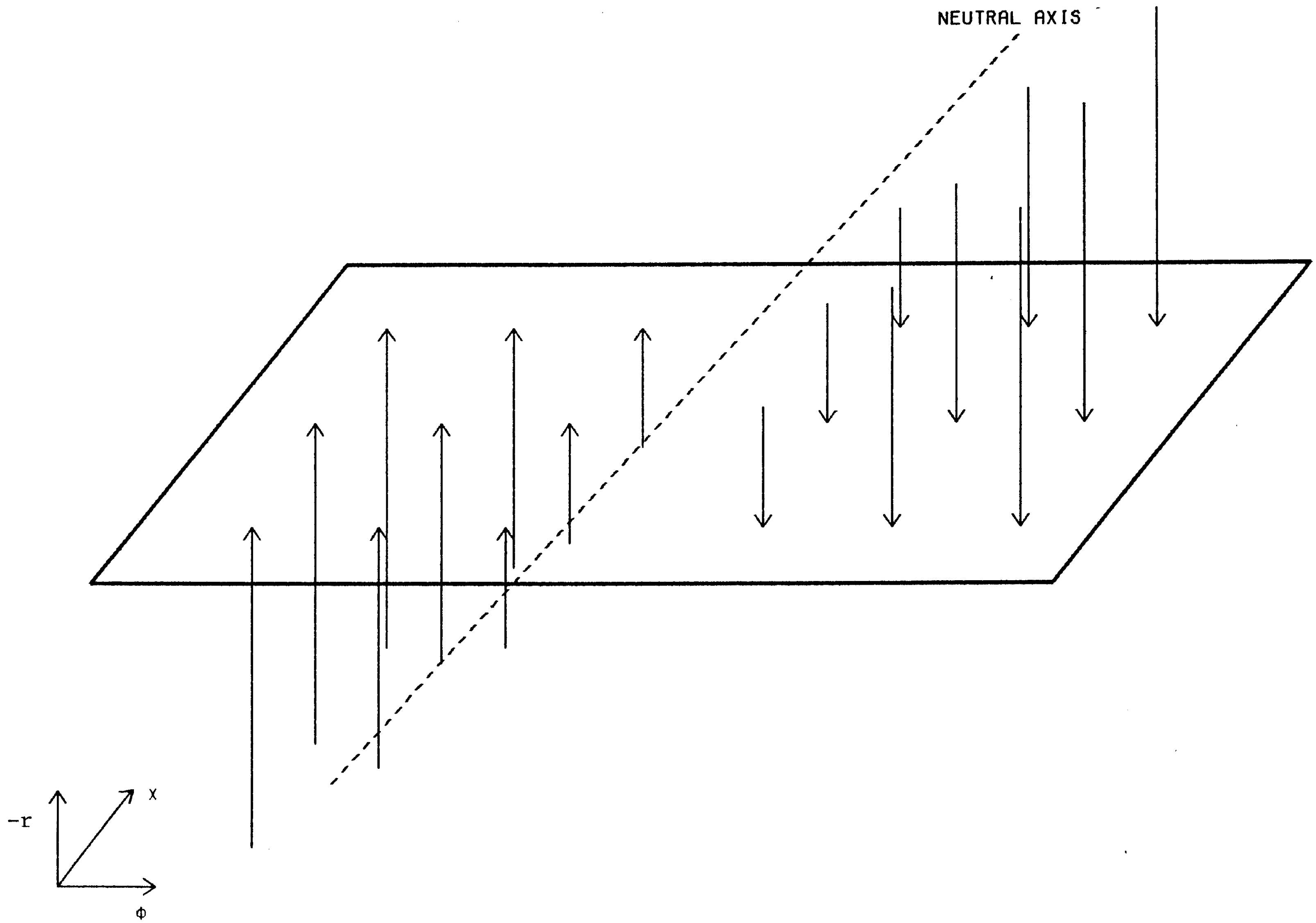


FIGURE 3.2B   EXTERNAL CIRCUMFERENTIAL MOMENT

original multiple loading conditions. This approach allows one to construct the multiple loading conditions, and then designate the flexibility of the attachment.

4. FOURIER ANALYSIS

4.1 EVEN AND ODD FOURIER SOLUTIONS

A double-sum Fourier series is constructed that satifies both the boundary conditions of simple-support and the equilibrium equations (2.1a - 2.1c). If one constructs the double-sum Fourier series for the radial displacement first, then the double-sum Fourier series for the shear and axial displacements are also determined. The total Fourier solution will be the sum of an even and an odd series in ϕ. A solution is said to be even if it corresponds to the even solution in ϕ, $\cos(m\phi)$, of the radial displacement. A solution is said to be odd if it corresponds to the odd solution in ϕ, $\sin(m\phi)$, of the radial displacement. The even and odd series in the axial direction are embedded in the terms $\sin(n\pi x/L)$ and $\cos(n\pi x/L)$, which each contain both even and odd solutions. The loading functions in the general equilibrium equations, P_x, P_ϕ, and P_r are also represented by double-sum Fourier series to be consistent with the displacement solutions.

The double-sum Fourier series for the radial displacement which satisfies the boundary conditions and the equations of equilibrium is:

$$w = \frac{1}{2} \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} \left[w^{e}_{m,n} \cos(m\phi) + w^{o}_{m,n} \sin(m\phi) \right] \sin(n\pi x/L) \qquad (4.1a)$$

Having designated the radial displacement solution, the v and u displacement solutions must be:

$$v = \frac{1}{2} \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} \left[v^{e}_{m,n} \sin(m\phi) + v^{o}_{m,n} \cos(m\phi) \right] \sin(n\pi x/L) \qquad (4.1b)$$

$$u = \frac{1}{2} \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} \left[u^{e}_{m,n} \cos(m\phi) + u^{o}_{m,n} \sin(m\phi) \right] \cos(n\pi x/L) \qquad (4.1c)$$

where the superscript e denotes an even term and o, an odd term.

Having determined the Fourier series of the displacement solutions, the loading functions which would allow the equilibrium equations to be solved by Fourier analysis are:

$$P_r = \frac{1}{2}\sum_{n=1}^{\infty}\sum_{m=0}^{\infty}\left[P_{r\,m,n}^{e}\cos(m\phi) + P_{r\,m,n}^{o}\sin(m\phi)\right]\sin(n\pi x/L) \qquad (4.1d)$$

$$P_\phi = \frac{1}{2}\sum_{n=1}^{\infty}\sum_{m=0}^{\infty}\left[P_{\phi\,m,n}^{e}\sin(m\phi) + P_{\phi\,m,n}^{o}\cos(m\phi)\right]\sin(n\pi x/L) \qquad (4.1e)$$

$$P_x = \frac{1}{2}\sum_{n=1}^{\infty}\sum_{m=0}^{\infty}\left[P_{x\,m,n}^{e}\cos(m\phi) + P_{x\,m,n}^{o}\sin(m\phi)\right]\cos(n\pi x/L) \qquad (4.1f)$$

4.2 FOURIER COEFFICIENTS

4.2.1 FOURIER COEFFICIENTS FOR LOADING FUNCTIONS OF RECTANGULAR AREAS

To determine the Fourier coefficients of the loading functions P_x, P_ϕ, and P_r , these functions are separated into even and odd solutions.

EVEN

$$P_r = \sum_{n=1}^{\infty}\sum_{m=0}^{\infty} P_{r\,m,n}^{e}\cos(m\phi)\sin(n\pi x/L)$$

$$P_\phi = \sum_{n=1}^{\infty}\sum_{m=0}^{\infty} P_{\phi\,m,n}^{e}\sin(m\phi)\sin(n\pi x/L)$$

$$P_x = \sum_{n=1}^{\infty}\sum_{m=0}^{\infty} P_{x\,m,n}^{e}\cos(m\phi)\cos(n\pi x/L)$$

ODD

$$P_r = \sum_{n=1}^{\infty}\sum_{m=0}^{\infty} P_{r\,m,n}^{o}\sin(m\phi)\sin(n\pi x/L)$$

$$P_\phi = \sum_{n=1}^{\infty}\sum_{m=0}^{\infty} P_{\phi\,m,n}^{o}\cos(m\phi)\sin(n\pi x/L)$$

$$P_x = \sum_{n=1}^{\infty}\sum_{m=0}^{\infty} P_{x\,m,n}^{o}\sin(m\phi)\cos(n\pi x/L)$$

Through the use of orthogonality, expressions for the Fourier coefficients $P_{x\,m,n}$, $P_{\phi\,m,n}$, and $P_{r\,m,n}$ can found. This analysis is shown for P_r as an example. Multiplying by $\cos(m'\phi)\sin(n'\pi x/L)$ and integrating,

$$\sum\sum\int_0^{2\pi}\int_0^{L} P_{r\,m,n}^{e}\cos(m\phi)\cos(m'\phi)\sin(n\pi x/L)\sin(n'\pi x/L)\,dx\,d\phi$$
$$= \int_0^{2\pi}\int_0^{L} P_r^{e}(x,\phi)\cos(m'\phi)\sin(n'\pi x/L)\,dx\,d\phi$$

Using orthogonality (Appendix B),

$$P_{r\,m,n}^{e}\,\pi(1+\delta_{m,0})L/2 = \int_0^{2\pi}\int_0^{L} P_r^{e}(x,\phi)\cos(m'\phi)\sin(n'\pi x/L)\,dx\,d\phi$$

Let

$$P_r(x,\phi) = P_r \text{ for } x_0 - b \le x \le x_0 + b \text{ and } \phi_0 - \beta \le \phi \le \phi_0 + \beta$$

$$= 0 \text{ otherwise}$$

Now integrate over attachment area (Figure 2.2) using integral formulas in Appendix B,

$$P_{r\,m,n}^{e} = \frac{2P_r}{\pi L(1+\delta_{m,0})}\int_{\phi_0-\beta}^{\phi_0+\beta}\cos(m'\phi)\,d\phi\int_{x_0-b}^{x_0+b}\sin(n'\pi x/L)\,dx$$

Let $m = m'$ and $n = n'$

$$P_{r\,m,n}^{e} = \frac{8P_r}{\pi^2 L(1+\delta_{m,0})mn}\cos(m\phi)\sin(m\beta)\sin(n\pi x/L)\sin(n\pi b/L)$$

This solution must be doubled to account for the mirror value of $-\phi$; that is, to account for the integral with limits $x_0 - b \le x \le x_0 + b$ and $-\phi_0 - \beta \le \phi \le -\phi_0 + \beta$.

Then finally,

$$P_{r\,m,n}^{e} = \frac{16P_r^{e}}{mn\pi^2 L(1+\delta_{m,0})}\cos(m\phi)\sin(m\beta)\sin(n\pi x/L)\sin(n\pi b/L)$$

A similar analysis is followed for all even and odd loading functions. The following Fourier loading coefficients result:

EVEN

$$P_{r\,m,n}^{e} = \frac{16P_r^{e}}{mn\pi^2 L(1+\delta_{m,0})}\cos(m\phi)\sin(m\beta)\sin(n\pi x/L)\sin(n\pi b/L)$$

$$P_{\phi\,m,n}^{e} = \frac{16P_\phi^{e}}{mn\pi^2}\sin(m\phi)\sin(m\beta)\sin(n\pi x/L)\sin(n\pi b/L)$$

$$P_{x\,m,n}^{e} = \frac{16P_x^{e}}{mn\pi^2(1+\delta_{m,0})(1+\delta_{n,0})}\cos(m\phi)\sin(m\beta)\cos(n\pi x/L)\sin(n\pi b/L)$$

ODD

$$P_{r\,m,n}^{o} = \frac{16P_r^{o}}{mn\pi^2}\sin(m\phi)\sin(m\beta)\sin(n\pi x/L)\sin(n\pi b/L)$$

$$P_{\phi\,m,n}^{o} = \frac{16P_\phi^{o}}{mn\pi^2 L(1+\delta_{m,0})}\cos(m\phi)\sin(m\beta)\sin(n\pi x/L)\sin(n\pi b/L)$$

$$P_{x\,m,n}^{o} = \frac{16P_x^{o}}{mn\pi^2 L(1+\delta_{n,0})}\sin(m\phi)\sin(m\beta)\cos(n\pi x/L)\sin(n\pi b/L)$$

4.2.2. FOURIER COEFFICIENTS FOR LOADING FUNCTIONS OF TRIANGULAR AREAS

The Fourier coefficients that correspond to loading functions in triangular areas are determined through a procedure similar to the coefficients in rectangular areas. It is after the use of orthogonality that the procedures differ. For example, in determining the Fourier loading coefficient P_r, one must now evaluate this integral over a triangular element (Figure 2.2):

$$P^e_{r\,m,n} = \frac{2P^e_r}{\pi L(1+\delta_{m,0})} \int_{\phi_a}^{\phi_b} \int_{x_a}^{A\phi+B} \cos(m\phi)\sin(n\pi x/L)\, dx\, d\phi$$

where,

$$A = \frac{x_b - x_a}{\phi_b - \phi_a} \quad \text{and} \quad B = \frac{x_a\,\phi_b - x_b\,\phi_a}{\phi_b - \phi_a}$$

Now, integrating with respect to x first,

$$\int_{\phi_a}^{\phi_b} \int_{x_a}^{A\phi+B} \cos(m\phi)\sin(n\pi x/L)\, dx\, d\phi =$$

$$\frac{L}{n\pi} \int_{\phi_a}^{\phi_b} \cos(m\phi) \left[\cos(n\pi x_a/L) - \cos(n\pi(A\phi + B)/L) \right] d\phi$$

Using the identity for cos(X + Y) in Appendix B,

$$= \frac{L}{n\pi} \int_{\phi_a}^{\phi_b} \cos(m\phi) \left[\cos(n\pi x_a/L) - \cos(n\pi A\phi/L)\cos(n\pi B/L) + \sin(n\pi A\phi/L)\sin(n\pi B/L) \right] d\phi$$

Integrating with respect to ϕ, setting $k = n\pi A/L$, and using trignometric integrals in Appendix B,

$$\int_{\phi_a}^{\phi_b} \cos(m\phi)\, d\phi = 1/m \sin(m\phi) \Big|_{\phi_a}^{\phi_b}$$

$$\int_{\phi_a}^{\phi_b} \cos(m\phi)\cos(k\phi)\, d\phi = \frac{1}{2} \left[\frac{\sin(m+k)\phi}{m+k} + \frac{\sin(m-k)\phi}{m-k} \right]_{\phi_a}^{\phi_b}$$

$$\int_{\phi_a}^{\phi_b} \cos(m\phi)\sin(k\phi)\, d\phi = \frac{1}{2} \left[\frac{\cos(m+k)\phi}{m+k} + \frac{\cos(m-k)\phi}{m-k} \right]_{\phi_a}^{\phi_b}$$

Finally,

$$P^e_{r_{m,n}} = \frac{4P^e_r}{n\pi^2(1+\delta_{m,0})}\left[\cos(n\pi x_a/L)\frac{1}{m}\sin(m\phi) - \frac{1}{2}\cos(n\pi B/L)\left[\frac{\sin(m+k)\phi}{m+k}\right.\right.$$

$$\left.\left.+\frac{\sin(m-k)\phi}{m-k}\right] + \frac{1}{2}\sin(n\pi B/L)\left[-\frac{\cos(m+k)\phi}{m+k} + \frac{\cos(m-k)\phi}{m-k}\right]\right]_{\phi_a}^{\phi_b}$$

A similar process is followed for all even and odd loading functions.

The following Fourier coefficients for loadings in triangular areas result:

EVEN

$$P^e_{r_{m,n}} = \frac{4P^e_r}{n\pi^2(1+\delta_{m,0})}\left[\cos(n\pi x_a/L)\frac{1}{m}\sin(m\phi) - \frac{1}{2}\cos(n\pi B/L)\left[\frac{\sin(m+k)\phi}{m+k}\right.\right.$$

$$\left.\left.+\frac{\sin(m-k)\phi}{m-k}\right] + \frac{1}{2}\sin(n\pi B/L)\left[-\frac{\cos(m+k)\phi}{m+k} + \frac{\cos(m-k)\phi}{m-k}\right]\right]_{\phi_a}^{\phi_b}$$

$$P^e_{\phi_{m,n}} = \frac{4P^e_{\phi}}{n\pi^2}\left[-\cos(n\pi x_a/L)\frac{1}{m}\cos(m\phi) - \frac{1}{2}\cos(n\pi B/L)\left[-\frac{\cos(m+k)\phi}{m+k}\right.\right.$$

$$\left.\left.-\frac{\cos(m-k)\phi}{m-k}\right] + \frac{1}{2}\sin(n\pi B/L)\left[-\frac{\sin(m+k)\phi}{m+k} + \frac{\sin(m-k)\phi}{m-k}\right]\right]_{\phi_a}^{\phi_b}$$

$$P^e_{x_{m,n}} = \frac{4P_x{}^e}{n\pi^2(1+\delta_{m,0})(1+\delta_{n,0})}\left[-\sin(n\pi x_a/L)\frac{1}{m}\sin(m\phi)\right.$$

$$+\frac{1}{2}\cos(n\pi B/L)\left[-\frac{\cos(m+k)\phi}{m+k} + \frac{\cos(m-k)\phi}{m-k}\right]$$

$$\left.+\frac{1}{2}\sin(n\pi B/L)\left[\frac{\sin(m+k)\phi}{m+k} + \frac{\sin(m-k)\phi}{m-k}\right]\right]_{\phi_a}^{\phi_b}$$

ODD

$$P_{r\,m,n}^{\,O} = \frac{4P_r^{\,O}}{n\pi^2}\left[-\cos(n\pi x_a/L)\,\frac{1}{m}\cos(m\phi) - \frac{1}{2}\cos(n\pi B/L)\left[-\frac{\cos(m+k)\phi}{m+k} - \frac{\cos(m-k)\phi}{m-k} \right] + \frac{1}{2}\sin(n\pi B/L)\left[-\frac{\sin(m+k)\phi}{m+k} + \frac{\sin(m-k)\phi}{m-k} \right]\right]_{\phi_a}^{\phi_b}$$

$$P_{\phi\,m,n}^{\,O} = \frac{4P_\phi^{\,O}}{n\pi^2(1+\delta_{m,0})}\left[\cos(n\pi x_a/L)\,\frac{1}{m}\sin(m\phi) - \frac{1}{2}\cos(n\pi B/L)\left[\frac{\sin(m+k)\phi}{m+k} + \frac{\sin(m-k)\phi}{m-k} \right] + \frac{1}{2}\sin(n\pi B/L)\left[-\frac{\cos(m+k)\phi}{m+k} + \frac{\cos(m-k)\phi}{m-k} \right]\right]_{\phi_a}^{\phi_b}$$

$$P_{x\,m,n}^{\,O} = \frac{4P_x^{\,O}}{n\pi^2(1+\delta_{n,0})}\left[\sin(n\pi x_a/L)\,\frac{1}{m}\cos(m\phi) - \frac{1}{2}\cos(n\pi B/L)\left[-\frac{\sin(m+k)\phi}{m+k} + \frac{\sin(m-k)\phi}{m-k} \right] + \frac{1}{2}\sin(n\pi B/L)\left[-\frac{\cos(m+k)\phi}{m+k} - \frac{\cos(m-k)\phi}{m-k} \right]\right]_{\phi_a}^{\phi_b}$$

4.3 DISPLACEMENT SOLUTION

To obtain the displacement solutions, one first separates the double-sum Fourier series for both displacements and loads into even and odd components. The even solutions are then substituted into equations 2.1a - 2.1c. A matrix equation results that gives the Fourier displacement coefficients $u_{m,n}$, $v_{m,n}$, $w_{m,n}$ in terms of the Fourier loading coefficients $P_{x_{m,n}}$, $P_{\phi_{m,n}}$, and $P_{r_{m,n}}$. The procedure is repeated for the odd coefficients, which will produce a matrix that determines the odd Fourier displacement coefficients. This procedure is shown for the even solutions as an example.

One begins with the even displacement solutions:

$$u^e = \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} u^e_{m,n} \cos(m\phi)\cos(n\pi x/L)$$

$$v^e = \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} v^e_{m,n} \sin(m\phi)\sin(n\pi x/L)$$

$$w^e = \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} w^e_{m,n} \cos(m\phi)\sin(n\pi x/L)$$

Now these solutions are substituted into the equilibrium equation (2.1a):

$$\frac{\partial^2 u}{\partial x^2} = -\frac{n^2\pi^2}{L^2} \sum\sum u^e_{m,n} \cos(m\phi)\cos(n\pi x/L)$$

$$\frac{(1-\nu)}{2a^2}\frac{\partial^2 u}{\partial \phi^2} = -\frac{m^2(1-\nu)}{2a^2} \sum\sum u^e_{m,n} \cos(m\phi)\cos(n\pi x/L)$$

$$\frac{(1+\nu)}{2a}\frac{\partial^2 v}{\partial x\partial \phi} = \frac{(1+\nu)}{2a}\frac{mn\pi}{L} \sum\sum v^e_{m,n} \cos(m\phi)\cos(n\pi x/L)$$

$$\frac{\nu}{a}\frac{\partial w}{\partial x} = -\frac{\nu}{a}\frac{n\pi}{L} \sum\sum w^e_{m,n} \cos(m\phi)\cos(n\pi x/L)$$

Assume that the external loading can be represented by a Fourier series,

$$P_x^e = \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} P_{x_{m,n}}^e \cos(m\phi)\cos(n\pi x/L)$$

Collecting terms,

$$-\left[\frac{n^2\pi^2}{L^2} + \frac{m^2(1-\nu)}{2a^2}\right] u^e_{m,n} + \left[\frac{(1+\nu)\,mn\pi}{2aL}\right] v^e_{m,n} - \left[\frac{\nu n\pi}{aL}\right] w^e_{m,n} = -\frac{1-\nu^2}{Et} P_{x_{m,n}}^e$$

Similarly, the even solutions of $u^e_{m,n}$, $v^e_{m,n}$, and $w^e_{m,n}$ are substituted into the equations for P_ϕ and P_r. The result is a system of three equations which may be written in matrix form.

$$\begin{bmatrix} C^e(1,1) & C^e(1,2) & C^e(1,3) \\ C^e(2,1) & C^e(2,2) & C^e(2,3) \\ C^e(3,1) & C^e(3,2) & C^e(3,3) \end{bmatrix} \begin{bmatrix} u^e_{m,n} \\ v^e_{m,n} \\ w^e_{m,n} \end{bmatrix} = \begin{bmatrix} D^e(1) \\ D^e(2) \\ D^e(3) \end{bmatrix}$$

Where,

$$C^e(1,1) = -\left[\frac{n^2\pi^2}{L^2} + \frac{m^2(1-\nu)}{2a^2}\right]$$

$$C^e(1,2) = \frac{(1+\nu)mn\pi}{2aL}$$

$$C^e(1,3) = -\frac{\nu n\pi}{aL}$$

$$C^e(2,1) = \frac{(1+\nu)mn\pi}{2aL}$$

$$C^e(2,2) = -\left[\frac{(1-\nu)n^2\pi^2}{2L^2} + \frac{m^2}{a^2} + \frac{t^2(1-\nu)n^2\pi^2}{12a^2L^2} + \frac{t^2m^2}{12a^4}\right]$$

$$C^e(2,3) = \left[\frac{t^2m^3}{12a^4} + \frac{mn^2\pi^2t^2}{12a^2L^2} + \frac{m}{a^2}\right]$$

$$C^e(3,1) = -\frac{n\pi\nu}{aL}$$

$$C^e(3,2) = \left[\frac{t^2m^3}{12a^4} + \frac{t^2mn^2\pi^2}{12a^2L^2} + \frac{m}{a^2}\right]$$

$$C^e(3,3) = -\left[\frac{1}{a^2} + \frac{t^2n^4\pi^4}{12L^4} + \frac{2t^2m^2n^2\pi^2}{12a^2L^2} + \frac{t^2m^4}{12a^4}\right]$$

$$D^e(1) = -\frac{(1-\nu^2)}{Et} P_{x\,m,n}^{\,e}$$

$$D^e(2) = -\frac{(1-\nu^2)}{Et} P_{\phi\,m,n}^{\,e}$$

$$D^e(3) = -\frac{(1-\nu^2)}{Et} P_{r\,m,n}^{\,e}$$

A similar set of matrix equations are also calculated by using the following odd solutions for $u^o_{m,n}$, $v^o_{m,n}$, and $w^o_{m,n}$. The odd solutions are:

$$u^o = \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} u^o_{m,n} \sin(m\phi)\cos(n\pi x/L)$$

$$v^o = \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} v^o_{m,n} \cos(m\phi)\sin(n\pi x/L)$$

$$w^o = \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} w^o_{m,n} \sin(m\phi)\sin(n\pi x/L)$$

FOURIER COEFFICIENTS FOR DISPLACEMENT SOLUTIONS

Through the use of Fourier analysis, two sets of 3 x 3 matrix equations result, one for even and one for odd solutions. These matrices may now be solved individually for the even and odd Fourier displacement coefficients. Once these coefficients have been determined, the total displacement, u, v, and w, may be computed from equations 2.1d - 2.1f.

A conventional row-manipulation technique is employed to solved for Fourier displacement coefficients. For example, consider the even matrix equation:

$$\begin{bmatrix} C^e(1,1) & C^e(1,2) & C^e(1,3) \\ C^e(2,1) & C^e(2,2) & C^e(2,3) \\ C^e(3,1) & C^e(3,2) & C^e(3,3) \end{bmatrix} \begin{bmatrix} u^e_{m,n} \\ v^e_{m,n} \\ w^e_{m,n} \end{bmatrix} = \begin{bmatrix} D^e(1) \\ D^e(2) \\ D^e(3) \end{bmatrix}$$

To solve for the even displacement coefficients, $u^e_{m,n}$, $v^e_{m,n}$, and $w^e_{m,n}$, the coefficient matrix is converted into an upper-triangular matrix.

$$\begin{bmatrix} C'(1,1) & C'(1,2) & C'(1,3) \\ 0 & C'(2,2) & C'(2,3) \\ 0 & 0 & C'(3,3) \end{bmatrix} \begin{bmatrix} u^e_{m,n} \\ v^e_{m,n} \\ w^e_{m,n} \end{bmatrix} = \begin{bmatrix} D'(1) \\ D'(2) \\ D'(3) \end{bmatrix}$$

The displacement coefficients may then computed for by simply "back substituting."

4.4 STRESS SOLUTION

Once the displacements of the shell have been determined, the bending and membrane forces, M and N, can be computed. The stress-resultant displacement relations are:

$$N_x = \frac{Et}{1 - \nu^2}\left[\frac{\partial u}{\partial x} + \nu\left[\frac{1}{a}\frac{\partial v}{\partial \phi} - \frac{w}{a}\right]\right] \tag{4.4a}$$

$$N_\phi = \frac{Et}{1 - \nu^2}\left[\frac{1}{a}\frac{\partial v}{\partial \phi} - \frac{w}{a} + \frac{\nu \partial u}{\partial x}\right] \tag{4.4b}$$

$$N_{x\phi} = \frac{Et}{2(1 + \nu)}\left[\frac{\partial v}{\partial x} + \frac{1}{a}\frac{\partial u}{\partial \phi}\right] \tag{4.4c}$$

$$M_x = -D\left[\frac{\partial^2 w}{\partial x^2} + \frac{\nu}{a^2}\left[\frac{\partial v}{\partial \phi} + \frac{\partial^2 w}{\partial \phi^2}\right]\right] \tag{4.4d}$$

$$M_\phi = -D\left[\frac{1}{a^2}\left[\frac{\partial v}{\partial \phi} + \frac{\partial^2 w}{\partial \phi^2}\right] + \frac{\nu \partial^2 w}{\partial x^2}\right] \tag{4.4e}$$

$$M_{x\phi} = -D(1 - \nu)\frac{1}{a}\left[\frac{\partial v}{\partial x} + \frac{\partial^2 w}{\partial x\, \partial \phi}\right] \tag{4.4f}$$

$$D = \frac{Et^3}{12(1 + \nu)}$$

The compound stresses in the shell may be calculated next:

$$\sigma_x = \frac{N_x}{t} + \frac{12M_x z}{t^3} \qquad \sigma_\phi = \frac{N_\phi}{t} + \frac{12M_\phi z}{t^3}$$

$$\tau_{x\phi} = \frac{N_{x\phi}}{t} + \frac{12M_{x\phi} z}{t^3} \qquad -\frac{t}{2} \le z \le \frac{t}{2}$$

4.5 INTERNAL PRESSURE

To solve for the Fourier series that represents internal pressure, which contains no axial component, one starts with the original radial loading series. Beginning with the even series,

$$P_r^e = \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} P_{r\,m,n}^e \cos(m\phi)\sin(n\pi x/L)$$

Using orthogonality,

$$P_{r\,m,n}^{e}\,\pi(1 + \delta_{m,0})L = \int_0^{2\pi}\int_0^{L} P_r(x,\phi)\cos(m'\phi)\sin(n'\pi x/L)\,dx\,d\phi \qquad (4.5a)$$

Now let

$$P_r(x,\phi) = P_r \quad \text{for} \quad \begin{array}{l} 0 \le x \le L \\ 0 \le \phi \le 2\pi \end{array}$$

Evaluating the double integral for m = 0,

$$P_{r\,m,n}^{e} = \frac{4P_r}{n\pi}[\,1 - \cos(n\pi)\,]$$

$$= \frac{4P_r}{n\pi}[1 - (-1)^n]$$

for m = 0 and n > 0

Similarly for the odd solution,

$$P_r^o = \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} P_{r\,m,n}^{o} \sin(m\phi)\sin(n\pi x/L)$$

Evaluating the double integral over the same limits produces,

$$P_r^o = \frac{2P_r}{\pi^2 n}[1-\cos(n\pi)][1-\cos(2m\pi)]$$

$$= 0 \quad \text{for all m and n}$$

Thus, in general,

$$P_r = \frac{1}{2} \sum_{n=1}^{\infty} \sum_{m=0}^{\infty} \left[P_{r\,m,n}^{e} \cos(m\phi) + P_{r\,m,n}^{o} \sin(m\phi) \right] \sin(n\pi x/L)$$

Since $P_{r\,m,n}^{o}$ is 0 for all m, n,

$$P_r = \frac{1}{2} \sum_{n=1}^{\infty} P_{r\,m,n}^{e} \sin(n\pi x/L) \quad \text{for m = 0} \qquad (4.5b)$$

Finally,

$$P_r = \sum_{n=1}^{\infty} \frac{2P}{n\pi}[1 - (-1)^n]\sin(n\pi x/L)$$

This Fourier series represents a uniform radially outward pressure over the entire surface of the cylinder. In order to account for the effect of internal pressure, this outward radial pressure is superposed on the external forces and moments. Mathematically, this is accomplished by merely adding the above Fourier series to the Fourier series representing the external loads.

Hydrostatic Pressure

To account for hydrostatic pressure, in a tank with the x-direction vertically upward, one integrates equation 4.5a, but does not assume that $P_r(x,\phi)$ is constant for x from 0 to L. Instead, the following expression is used:

$$P_r(x,\phi) = \rho g(x_o - x) \quad \text{for} \quad 0 < x \leq x_o$$
and
$$P_r(x,\phi) = 0 \quad \text{for} \quad x_o < x \leq L$$

where x_o = Fluid fill level from vessel bottom
ρ = Fluid density

Now, evaluating equation 4.5a for m = 0,

$$P^e_{r_{m,n}} = \frac{4\rho g}{L}\left[\frac{x_o L}{n\pi}\left[1 - \cos(fx_o)\right] - \left[\frac{1}{f^2}\sin(fx_o) - \frac{x_o}{f}\cos(fx_o)\right]\right]$$

where,

$$f = \frac{n\pi x}{L}$$

Like the uniform, radial outward pressure, the odd terms of this series are also zero. Thus, substituting this new expression for $P^e_{r_{m,n}}$ into equation 4.5b, the Fourier series becomes:

$$P_r = \sum_{n=1}^{\infty} \frac{2\rho g}{L}\left[\frac{x_o L}{n\pi}\left[1 - \cos(fx_o)\right] - \left[\frac{1}{f^2}\sin(fx_o) - \frac{x_o}{f}\cos(fx_o)\right]\right]\sin(n\pi x/L)$$

4.6 SUMMARY

In this chapter, the loading Fourier coefficients, $P_{x_{m,n}}$, $P_{\phi_{m,n}}$, and $P_{r_{m,n}}$, for both even and odd solutions have been determined by orthgonality. Once these coefficients have been found, equations 4.1d - 4.1f may be used to determine the loads, P_x , P_ϕ, and P_r at any location on the cylindrical surface. Using equations 4.1a - 4.1c, the displacements u, v, and w can also be expressed as Fourier series. The Fourier series for the displacements and the loadings may now be substituted into the general equilibrium equations 2.1a - 2.1c. Equations 2.1a - 2.1c may now be written as two matrix equations, one for even solutions and one for odd solutions. These matrix equations are then used to obtain the displacement Fourier coefficients $u_{m,n}$, $v_{m,n}$, and $w_{m,n}$. Once the displacement Fourier coefficients have been computed, the displacement solutions, equations 4.1a - 4.1c, may be used to determine the deflection of the shell at any location on the cylinder. Using these displacement solutions, the bending and membrane forces may be calculated from equations 4.4a - 4.4f. The compound and shear stresses may then be computed.

Section 4.5 developed the Fourier loading series that would simulate internal and hydrostatic pressure. To account for internal or hydrostatic pressure, these series would be added to the existing Fourier series which represent the external loads.

Since the Fourier series for displacements and loadings were developed so that they fulfill the boundary conditions of simple-support, we now have a solution to the equilibrium equations for a simply-supported cylindrical shell.

5. THE METHOD OF COLOCATION

5.1 BASIC THEORY

The method of colocation is a technique that is used to solve for forces that produce a specified displacement field for a designated patch area. To accomplish this, a two step process is involved. First, using the previously mentioned Fourier analysis with unit loadings, one can obtain displacements. These displacements are effectively the elements of a flexibility matrix. Consider an n x n subdivision of elements, the displacements are found to be:

$$
\begin{aligned}
w(1) &= C(1,1)P(1) + C(1,2)P(2) + C(1,3)P(3) + \ldots + C(1,4n^2)P(4n^2) \\
w(2) &= C(2,1)P(1) + C(2,2)P(2) + C(2,3)P(3) + \ldots + C(2,4n^2)P(4n^2) \\
w(3) &= C(3,1)P(1) + C(3,2)P(2) + C(3,3)P(3) + \ldots + C(3,4n^2)P(4n^2) \\
&\vdots \\
w(4n^2) &= C(4n^2,1)P(1) + C(4n^2,2)P(2) + C(4n^2,3)P(3) + \ldots + C(4n^2,4n^2)P(4n^2)
\end{aligned}
$$

To obtain the values of the coefficient (or flexibility) matrix, a unit loading is applied for P(1) and all other loadings are zero. Thus,

$$
\begin{aligned}
w(1) &= C(1,1)P(1) \\
w(2) &= C(2,1)P(1) \\
w(3) &= C(3,1)P(1) \\
&\vdots \\
w(4n^2) &= C(4n^2,1)P(1)
\end{aligned}
$$

But, since P(1) is a unit loading, the coefficients are just equal to the displacements.

$$
\begin{aligned}
C(1,1) &= w(1)\\
C(2,1) &= w(2)\\
C(3,1) &= w(3)\\
&\vdots\\
C(4n^2,1) &= w(4n^2)
\end{aligned}
$$

To obtain the next set of coefficents, P(2) becomes a unit loading and all other loadings are zero. This process is repeated iteratively for all loadings in all directions. Thus, the final matrix is comprised of four types of coefficients: from radial displacements due to radial loadings, radial displacements due to shear loadings, shear displacements due to radial loadings, and shear displacements due to shear loadings.

The final matrix equation for colocation takes the form of

$$
\begin{bmatrix} w(1) \\ \vdots \\ w(4n^2) \\ v(1) \\ \vdots \\ v(4n^2) \end{bmatrix}
=
\begin{bmatrix}
w_r(1,1) & \dots & w_r(1,4n^2) & w_\phi(1,1) & \dots & w_\phi(1,4n^2) \\
\vdots & & \vdots & \vdots & & \vdots \\
w_r(4n^2,1) & \dots & w_r(4n^2,4n^2) & w_\phi(4n^2,1) & \dots & w_\phi(4n^2,4n^2) \\
v_r(1,1) & \dots & v_r(1,4n^2) & v_\phi(1,1) & \dots & v_\phi(1,4n^2) \\
\vdots & & \vdots & \vdots & & \vdots \\
v_r(4n^2,1) & \dots & v_r(4n^2,4n^2) & v_\phi(4n^2,1) & \dots & v_\phi(4n^2,4n^2)
\end{bmatrix}
\begin{bmatrix} P_r(1) \\ \vdots \\ P_r(4n^2) \\ P_\phi(1) \\ \vdots \\ P_\phi(4n^2) \end{bmatrix}
\tag{5.1}
$$

Where, the elements of the flexibility matrix are:

$w_r(n,n)$ = The coefficients of a radial displacement due to a radial loading.

$w_\phi(n,n)$ = The coefficients of a radial displacement due to a shear loading.

$v_r(n,n)$ = The coefficients of a shear displacement due to a radial loading.

$v_\phi(n,n)$ = The coefficients of a shear displacement due to a shear loading.

If the displacements are specified on the left-hand side of this matrix equation, the coefficient matrix can be inverted to obtain a stiffness matrix, and the loadings that will produce the specified displacment field may be solved.

5.2 UNIFORM DISPLACEMENTS

In order to allow the plug to move uniformly as a rigid body, w(n) and v(n) must be determined to ensure this motion. From Figure 5.1, one can see that a uniform unit lateral displacement, parallel to $\phi = 0$, is guaranteed if w(n) and v(n) satisfy the following:

$$w(n)\sin\phi = v(n)\cos\phi$$

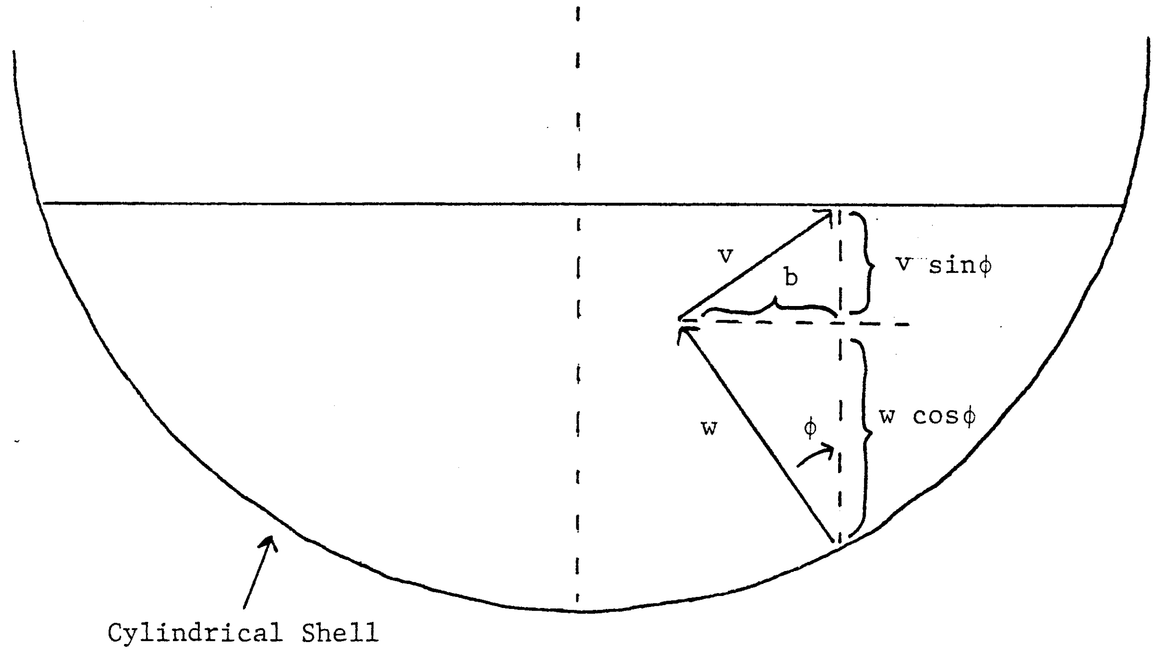
$$w(n)\cos\phi + v(n)\sin\phi = 1$$

Solving for w(n) and v(n):

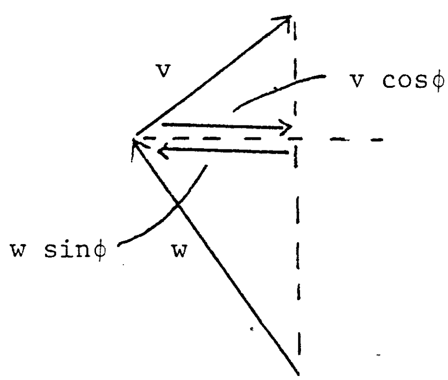
$$w(n) = \cos\phi$$

$$v(n) = \sin\phi$$

These expressions for w(n) and v(n) are substituted into the general matrix equation 5.1, and then this equation is solved for the corresponding loadings that will produce a uniform, rigid body displacement. Having solved for the required forces from a known set of displacements, the stiffness may be computed.



Detail of b:



Conditions for uniform displacment:

$$w \sin\phi = v \cos\phi$$

$$w \cos\phi + v \sin\phi = 1$$

Solving for w and v:

$$w = \cos\phi$$

$$v = \sin\phi$$

Figure 5.1 Conditions For Uniform Displacement

5.3 AN EXAMPLE OF COLOCATION

As an example, the colocation method was used to calculate the required forces necessary for a unit radially inward displacement of a rectangular attachment. The cylinder used in this example had the following parameters:

Radius = 1.0 in.
Length = 8.0 in.
Thickness = 0.01 in.
E = 1.0 psi.
ν = 0.3

The rectangular attachment had the following dimensions:

L_x = 0.5 in.
L_ϕ = 0.5 in.
T_x = 0.125 in.
T_ϕ = 0.25 in.

The attachment was placed two inches from the bottom of the cylinder.

The calculated radial and shear forces that were required to produce a unit radially inward displacement of the rectangular attachment are shown in Figure 5.2. It is interesting to note that very large shear forces (twice as large as radial forces) are necessary in regions 1 and 3 to obtain a unit displacement. These large shear forces are expected, because in simulating the uniform displacement of a rigid attachment, the curvature of the shell-attachment intersection must be maintained. This is not true for a soft attachment in which the curvature of the shell-attachment intersection would tend to flatten. Since the attachment is located near the bottom of the cylinder, the radial force is inward in region 4 and outward in region 2. This combination of forces was expected because it would ensure a uniform displacement for an attachment near the bottom of a simply-supported cylinder, since the shell becomes stiffer near the ends.

By loading each the four regions with the corresponding radial and shear forces in Figure 5.2, one may calculate the membrane and bending stresses for this rigid attachment in the usual manner.

<table>
<tr><td colspan="2">Region 2: Radial Force = −1.920
Shear Force = 8.818×10^{-4}</td></tr>
<tr><td>Region 3:
Radial Force = 5.228
Shear Force = 10.48</td><td>Region 1:
Radial Force = 5.228
Shear Force = −10.48</td></tr>
<tr><td colspan="2">Region 4: Radial Force = 0.1766
Shear Force = 9.055×10^{-4}</td></tr>
</table>

Figure 5.2 Calculated Radial And Shear Forces That Produce A Unit Radially Inward Displacement Of The Rectangular Attachment (10^{-4} lbs.)

6. DISCUSSION OF RESULTS

6.1 COMPARISON WITH BIJLAARD'S WORK

In his paper "Stresses From Radial Loads And External Moments In Cylindrical Pressure Vessels" (Ref. 3), Bijlaard includes tabulated results of calculations. As a first comparison, the method presented here is used to calculate the same problems examined by Bijlaard. The geometry and properties of the cylinder are chosen to aid in non-dimensionalizing the results, since all of Bijlaard's tables are non-dimensionalized. The parameters of the cylinder are:

Radius = 1.0
Length = 8.0
ν = 0.3

The tables included the following values:

- M x N = Number of terms in the Fourier series
- t = Thickness of shell
- W = Radial displacement
- M_ϕ = Circumferential bending moment per unit length
- M_x = Longitudinal bending moment per unit length
- $-N_\phi$ = Circumferential membrane force per unit length
- $-N_x$ = Longitudinal membrane force per unit length
- P = Surface loading
- β = One-half the length of the attachment-cylinder intersecting area in the x and ϕ directions

All attachments were of square geometry. Bijlaard usually used a 41 x 61 term Fourier series. However, as the shell became thinner, his value of N exceeded 100.

Radial Loads

Tables A, B, and C reference Table 9 in Bijlaard's paper. The three tables display the non-dimensionalized deflections, membrane forces per unit length, and bending moments per unit length using a varying number of terms in the Fourier series.

TABLE A – Radial Loading with $t = 1/15$ and $\beta = 1/4$

M X N	t	β	W/(P/EA)	M_ϕ/P	M_x/P	$-N_\phi/(P/A)$	$-N_x/(P/A)$
Bijlaard	1/15	1/4	423	0.077	0.0485	1.820	2.23
10 x 10	1/15	1/4	437	0.072	0.0324	0.993	2.09
25 x 25	1/15	1/4	457	0.084	0.0530	1.754	2.30
50 x 50	1/15	1/4	457	0.084	0.0517	1.749	2.30
75 x 75	1/15	1/4	457	0.084	0.0511	1.746	2.30
100 x 100	1/15	1/4	457	0.084	0.0515	1.746	2.30

TABLE B – Radial Loading with $t = 0.01$ and $\beta = 1/8$

M X N	t	β	W/(P/EA)	M_ϕ/P	M_x/P	$-N_\phi/(P/A)$	$-N_x/(P/A)$
Bijlaard	.01	1/8	30,136	0.0716	0.0394	9.792	14.192
10 x 10	.01	1/8	26,675	0.0391	0.0138	2.006	9.791
25 x 25	.01	1/8	29,396	0.0675	0.0291	6.380	13.372
50 x 50	.01	1/8	29,889	0.0707	0.0379	9.478	14.141
75 x 75	.01	1/8	29,913	0.0718	0.0389	9.640	14.163
100 x 100	.01	1/8	29,896	0.0709	0.0370	9.485	14.155

TABLE C - Radial Loading with t = 0.003 and β = 1/16

M X N	t	β	W/(P/EA)	M_ϕ/P	M_x/P	$-N_\phi/(P/A)$	$-N_x/(P/A)$
Bijlaard	0.003	1/16	370,000	0.095	0.0455	34.5	47.0
10 x 10	0.003	1/16	296,550	0.023	0.0076	2.7	21.8
25 x 25	0.003	1/16	354,310	0.059	0.0217	10.6	36.4
50 x 50	0.003	1/16	367,760	0.079	0.0348	22.3	43.3
75 x 75	0.003	1/16	371,670	0.083	0.0429	30.1	45.3
100 x 100	0.003	1/16	372,730	0.084	0.0469	32.7	45.7

One can see that our results are in good agreement with Bijlaard's. As expected, more terms in the Fourier series are needed for convergence when t and β become small. In this situation, about 75 x 75 terms are required for good convergence. For larger values of t and β, approximately 50 x 50 terms are required for good convergence. Since Bijlaard truncated the governing equilibrium equation, our results are not expected to match Bijlaard's exactly.

External Moments

Table D references Tables 2 and 5 in Bijlaard's paper. The geometry and properties of the cylinder were the same as for the radial loadings. The thickness was t = 0.01. Our number of patch divisions varied to allow greater refinement within the patch area.

TABLE D - External Moment Loading with
$t = 0.01$ and $\beta = 1/8$

TYPE	M x N	PATCH DIVISION	M_ϕ	M_x	$-N_\phi$	$-N_x$
Longitudinal						
	Bijlaard		0.199	0.270	85.28	32.8
	50 x 50	1 x 1	0.157	0.205	67.18	27.1
	75 x 75	1 x 1	0.177	0.219	70.18	27.4
	75 x 75	3 x 3	0.198	0.258	82.54	32.1
Circumferential						
	Bijlaard		0.629	0.306	32.72	60.08
	50 x 50	1 x 1	0.255	0.146	16.60	25.97
	75 x 75	3 x 3	0.611	0.314	31.55	56.29

In regard to moment loadings, our results are in good agreement with Bijlaard's. It is quite evident that as more elements in the attachment-cylinder intersection are used, the agreement becomes better. When more elements are used, there are more elements on either side of the neutral axis. As a result, the moment that is applied may be simulated more accurately. Since t and β are small, approximately 75 x 75 terms are needed for good convergence.

6.2 COMPARISON WITH AN API-650 "LOW TYPE" NOZZLE ON A STORAGE TANK

Ref. 5 contains results that were obtained by applying separate external moments and radial forces on an API-650 "Low Type" 24-in.-dia nozzle. The applied loads were:

Longitudinal Moment = 1800×10^3 in-lbs.
Circumferential Moment = 2724×10^3 in-lbs.
Radial Thrust = 40.8×10^3 lbs.

To simulate the presence of a reinforcing pad on the tank, the loads were distributed over a reinforcing pad which had dimensions:

Inner Radius = 12.0 in.
Outer Radius = 27.0 in.

The locations of the dial gauges are shown in Figure 6.1. The shell deflections due to various loadings are shown in Table E. The calculations used 50 x 50 Fourier series, except where indicated. The parameters that were used to calculate shell deflections and stiffnesses were:

Radius = 2070 in. (172.5 ft.)
Height = 576 in. (48 ft.)
Tank Thickness = 1.345 in.
$E = 30 \times 10^6$ psi.
$\nu = 0.3$

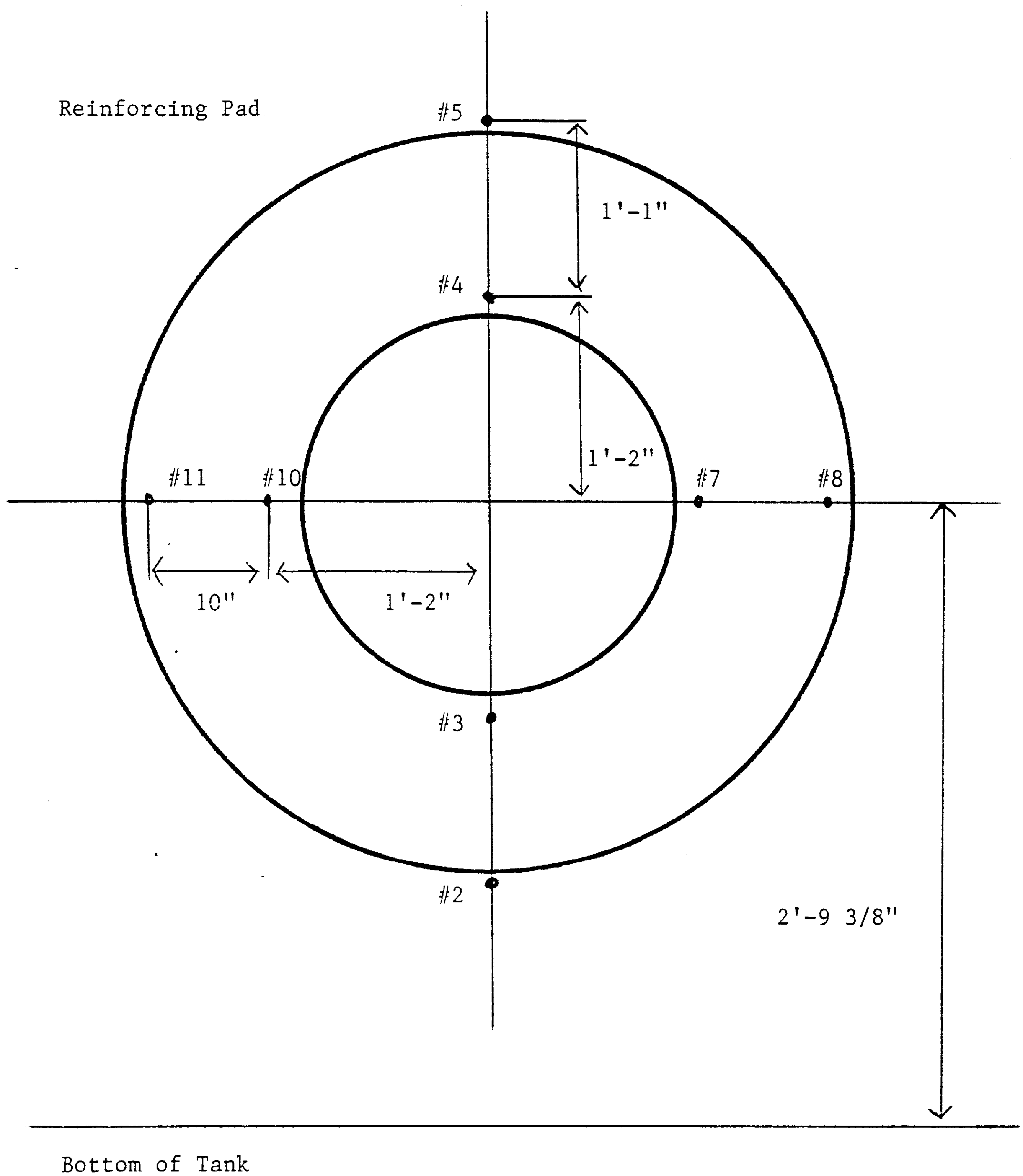


Figure 6.1 Location of Dial Gauges on Reinforcing Pad
(From Reference 5)

TABLE E - Shell Deflections (in.) near an API-650
Nozzle Due to Various External Loads

LOAD	Dial Gauges NO. 2	NO. 3	NO. 4	NO. 5	NO. 7	NO. 8	NO. 10	NO. 11
Longitudinal Moment								
Experiment	0.020	0.099	0.409	0.471	0.246	0.222	0.223	0.231
Calculated	0.033	0.128	0.432	0.504	0.278	0.267	0.278	0.267
Circumfer. Moment								
Experiment	0.012	0.028	0.041	0.033	0.170	0.210	-0.093	-0.137
Calculated	0.0	0.0	0.0	0.0	0.068	0.141	-0.068	-0.141
Calculated (75 x 75)	0.0	0.0	0.0	0.0	0.153	0.249	-0.153	-0.249
Radial Thrust								
Experiment	0.078	0.152	0.350	0.380	0.250	0.222	0.235	0.229
Calculated	0.086	0.246	0.427	0.432	0.359	0.345	0.363	0.351

The calculations for the shell deflections are larger than the experimental values, because the cylindrical model is simply-supported, and the actual tank is fixed at the base.

The Fourier series of 50 x 50 terms provided good convergence for the calculated values. However, when a circumferential moment was applied, a series of 75 x 75 terms was necessary to ensure good convergence.

Using the above data, stiffness coefficients may be determined (Ref.5). A comparison of calculated and experimental values is shown in Table F.

TABLE F - Comparison of Stiffness Coefficients

Stiffness Coefficient	Experimental	Calculated
K Longitundinal (in-lbs/rad.)	163×10^6	166×10^6
K Circumferential (in-lbs/rad.)	290×10^6	249×10^6
K Radial (lbs/in)	131×10^3	111×10^3

The calculated stiffness coefficients are in good agreement with the experimental values. Since the calculated shell deflections are generally larger that the experimentally measured values, this discrepancy will cause smaller values for stiffness coefficients.

Hydrostatic Pressure

Ref. 5 also contains measured deflections due to hydrostatic pressure. A comparison of calculated and experimental values is shown in Table G.

TABLE G - Shell Deflections Due to Hydrostatic Pressure

Tank Water Level	Dial Gauges No. 3	No. 4	No. 5
24'-0"			
Measured (in.)	0.178	0.385	0.480
Calculated (in.)	0.417	0.781	0.860
44'-0"			
Measured	0.498	1.151	1.380
Calculated	0.824	1.583	1.791

The calculated deflections are larger by a factor of two, when the tank is only half full. However, the calculated values are in better agreement with the measured deflections when the tank is filled to near capacity. Two reasons account for this discrepancy. First, calculations are based on the assumption that the shell-to-bottom junction is free to rotate. In reality, the bottom of the tank is fixed. As a result, calculated values can be expected to differ reatly from measured deflections near the bottom of the tank. Second, when the tank is nearly full, the hydrostatic load becomes predominant over the rigidity of the bottom-to-shell junction. This is not the case when the filling level in the tank is low. Thus, better calculated deflections can be expected when the filling level is high and the attachment is near the bottom.

6.3 COMPARISON WITH THE PRESSURE VESSEL ATTACHMENTS OF WELDING RESEARCH COUNCIL BULLETIN NO. 60

In Welding Research Council Bulletin No. 60 (Ref. 6), Cranch presents experimental data that was obtained by independently loading five attachments on a pressure vessel. Only two of the five attachments were used for this comparison (Difficulties were encountered in accurately simulating a third attachment, see Appendix E). Due to the many ambiguities of the Bulletin's presentation of the experiment, it should be kept in mind that this comparison should not treated as numerically exact one. Rather, it should reflect the general trend of the solution presented in this thesis.

In attempting to accurately simulate the Bulletin's experiments on the attachments, some major ambiguities of and inconsistencies in the tests are causes for discrepancies between experimental and calculated values. The Bulletin does not describe the method in which the vessel is supported. As seen in the last comparison, the boundary conditions at the ends of the cylinder are very important. The welds that connect the attachments to the vessel are inconsistent, because the width of the weld changes as one moves around the circumference of the attachment. The treatment in the Bulletin of deflections is incomplete, because one cannot tell what deflections were measured with respect to. Taken together, these factors are most likely to affect the agreement between calculated and experimental values.

The Bulletin also presents a theoretical value for every experimental value. These theoretical values were based on Bijlaard's original method for determining stresses at attachment-shell intersections. However, the Bulletin does not explain how circular attachments were simulated, since Bijlaard's method only accounted for rectangular attachment-shell intersections.

The parameters of the pressure vessel used in simulating the experiments were (See Figure 6.2):

Radius = 24.0 in.
Length = 104.0 in.
Vessel thickness = 0.625 in.
$E = 29.5 \times 10^6$ psi.
$\nu = 0.3$

The dimensions used to simulate the attachments were (See Figures 6.3 and 6.4):

TABLE H – Summary of Attachment Dimensions (in.)

Attachment	Inner Radius	Outer Radius	Weld Thickness
No. 1	3.033	3.844	17/32
No. 3	3.033	5.810	9/16

To be consistent with the methods in the Bulletin, the loading area on the cylinder included the thickness of the welds. For Attachment 3, the load was applied over the reinforcing pad and its welds. The bending and membrane stresses were then calculated at the edge of each attachment. For Attachment 3, the stresses were calculated at the edge of the reinforcing pad. The experimental values in the Bulletin were obtained by extrapolating the experimentally determined stresses to the edge of the attachments. As in the Bulletin, values along the meridian were those along the circumferential axis, and values along the generator were those along the longitudinal axis.

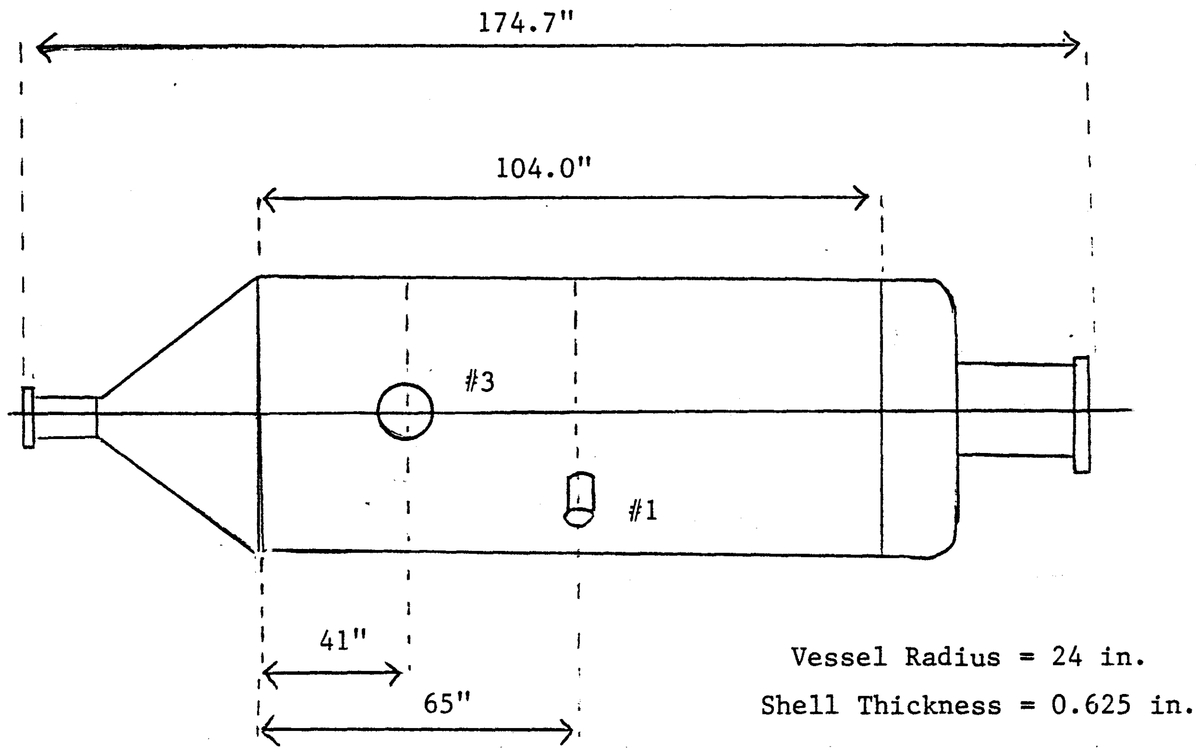


Figure 6.2 Pressure Vessel Geometry
(From Reference 6)

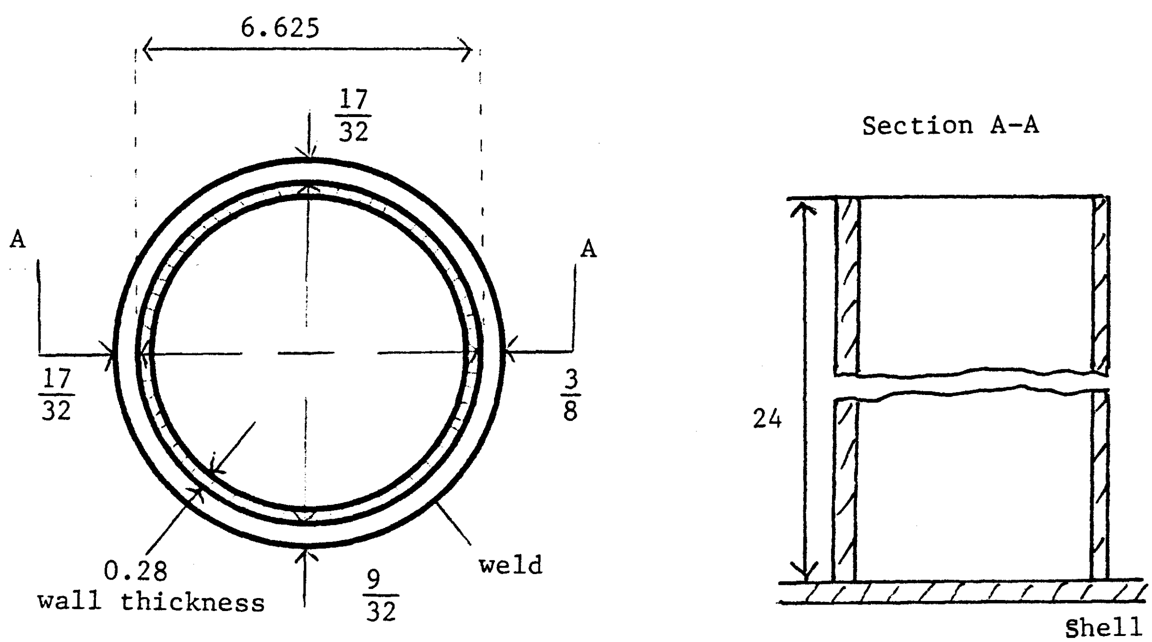


Figure 6.3 Attachment 1 Geometry, ins. (From Reference 6)

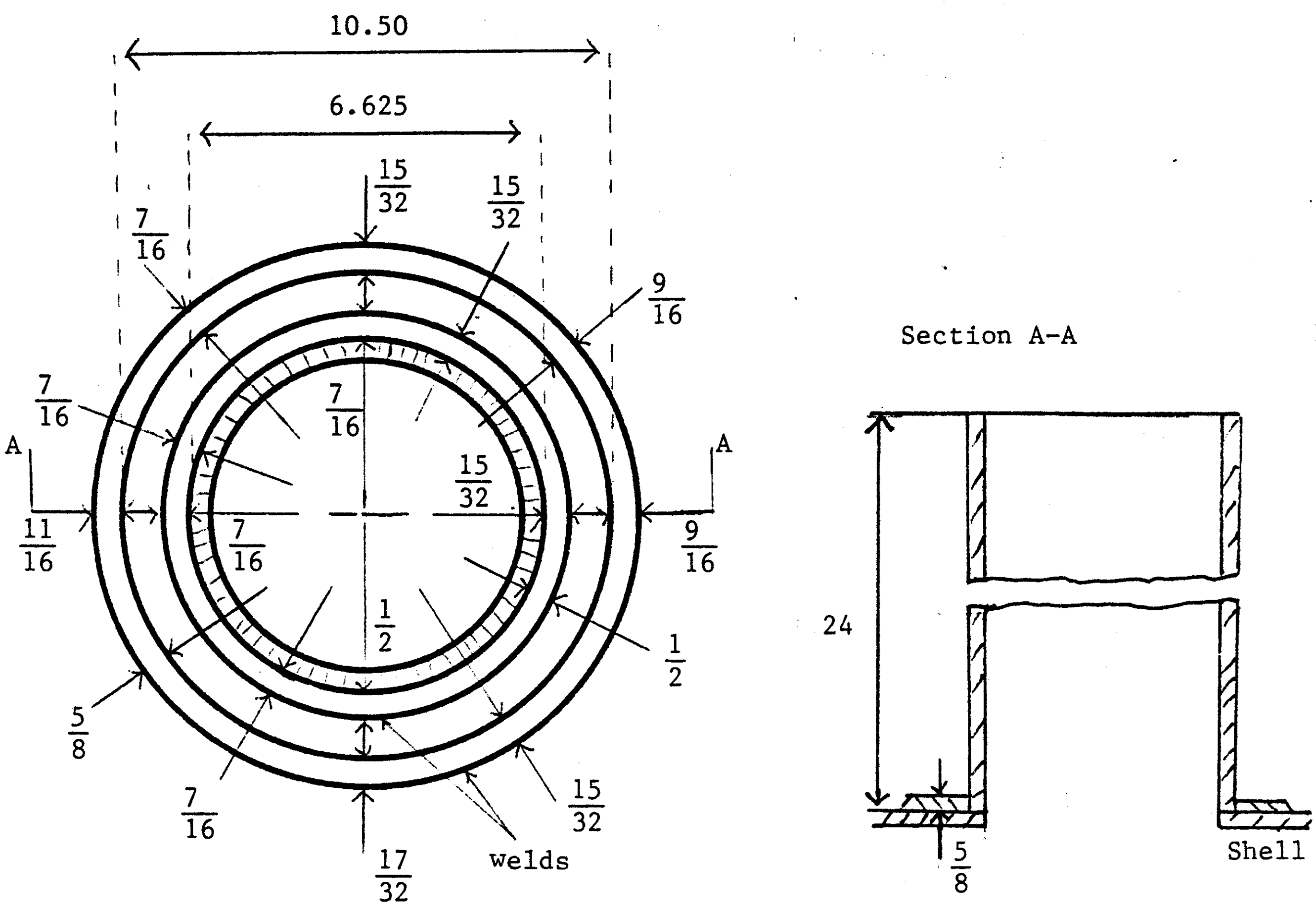


Figure 6.4 Attachment 3 Geometry, ins.
(From Reference 6)

6.3.1 BENDING AND MEMBRANE STRESSES

Tables I and J contain bending and membrane stress per unit force for a radial load. My calculated values are shown along with the Bulletin's experimental and theoretical values. For Attachment 1, the calculated values are in good agreement with the Bulletin's theoretical results. This agreement is expected since the theoretical values are based on Bijlaard's method for determining stresses in cylindrical shells. The agreement between experimental and calculated stresses is consistent and within the same order of magnitude. Since Attachment 3 involves a configuration for which no detailed theory exists, the theoretical values in the Bulletin were determined from certain recommendations that Bijlaard made using spherical shell theory. As a result, my calculated values, which were based on cylindrical shell theory, will tend to disagree with the theoretical values. However, the calculated values are consistent in many respects to the experimental values, and the stresses are within the correct order of magnitude.

TABLE I - Attachment No. 1, Radial Load, Bending and Membrane Stress per Unit Force (psi./lbs.), P = 16,600 lbs.

Method	N_x/Pt	$6M_x/Pt^2$	N_ϕ/Pt	$6M_\phi/Pt^2$
Meridian:				
Experiment	-0.21	-0.81	-0.43	-1.80
Calculated	-0.42	-0.67	-0.33	-1.26
Theoretical	-0.38	-1.01	-0.33	-1.49
Generator:				
Experiment	-0.27	-0.71	-0.07	-0.49
Calculated	-0.45	-0.85	-0.27	-1.34
Theoretical	-0.38	-1.01	-0.33	-1.49

TABLE J - Attachment No. 3, Radial Load, Bending and Membrane Stress per Unit Force (psi./lbs.), P = 19,500 lbs.

Method	N_x/Pt	$6M_x/Pt^2$	N_ϕ/Pt	$6M_\phi/Pt^2$
Meridian:				
Experiment	-0.19	-0.47	-0.30	-1.20
Calculated	-0.23	-0.25	-0.17	-0.45
Theoretical	-0.30	-0.53	-0.23	-0.96
Generator:				
Experiment	-0.23	-0.59	+0.01	-0.39
Calculated	-0.29	-0.40	-0.12	-0.76
Theoretical	-0.30	-0.53	-0.23	-0.91

Tables K and L contain bending and membrane stress per unit couple for longitudinal and circumferential couples. Unlike the case involving radial loads, the Bulletin's theoretical values for both attachments were obtained using Bijlaard's method for determining stresses with cylindrical shell theory.

Examining the stresses due to a longitudinal couple, my calculated values are in good agreement with the theoretical values. This agreement was expected. The calculated values are also very consistent with the experimental results, and in certain cases, the calculated stresses are closer to the experimental results than the theoretical values.

For the circumferential couple, calculated stresses are in good agreement with the theoretical values, as they should be. As before, the calculated results are consistent with experimental values, and in some cases, the calculated values represent a better estimate of the experimental stresses than the theoretical values.

Thus, although there were ambiguities in simulating the experimental procedure in this Bulletin, my calculated values were in consistent and in good agreement with the experimental and theoretical values. The discrepancies arose primarily because of ambiguities in the treatment of the welds of the attachments and in the method of support for the vessel.

TABLE K - Longitudinal Couple, Bending and Membrane Stress per Unit Couple (psi./in-lbs)

Method (P = in-lbs.)	N_x/Pt	$6M_x/Pt^2$	N_ϕ/Pt	$6M_\phi/Pt^2$
Attachment No. 1 ($P = 1.344 \times 10^5$)				
Generator:				
Experiment	0.019	0.27	0.036	0.15
Calculated	0.039	0.37	0.113	0.24
Theoretical	0.029	0.35	0.098	0.22
Attachment No. 3 ($P = 1.296 \times 10^5$)				
Generator:				
Experiment	0.028	0.14	0.017	0.074
Calculated	0.025	0.12	0.053	0.104
Theoretical	0.023	0.14	0.066	0.094

TABLE L - Circumferential Couple, Bending and Membrane Stress per Unit Couple (psi./in-lbs.) at Edge of Attachment

Method	N_x/Pt	$6M_x/Pt^2$	N_ϕ/Pt	$6M_\phi/Pt^2$
Attachment No. 1 (P = 59,520 in-lbs.)				
Meridian:				
Experiment	0.015	0.24	0.079	0.53
Calculated	0.060	0.32	0.038	0.60
Theoretical	0.045	0.27	0.031	0.50
Attachment No. 3 (P = 87,340 in-lbs.)				
Meridian:				
Experiment	0.024	0.16	0.091	0.41
Calculated	0.042	0.14	0.027	0.25
Theoretical	0.044	0.15	0.023	0.29

6.3.2 DEFLECTIONS

Comparisons between the calculated and experimentally measured deflections per unit radial load for Attachments 1 and 3 are shown in Figures 6.5 and 6.6. In examining Figure 6.5, one see that the calculated deflections per unit radial load are much larger than those experimentally determined. This discrepancy is caused by the stiffening of the shell near the attachment due to the welds. Since the weld thickness is larger than the attachment thickness, this stiffening effect is very pronounced. A better calculated solution may be obtained by using the colocation technique.

In Figure 6.6, the calculated deflections per unit radial load are lower than the experimentally determined values. However, the slope of the calculated solution is in good agreement with the experimental results. Here the welds are not a major factor, since the thickness of the reinforcing pad is much larger than the thickness of the welds. The constant discrepancy between calculated and experimental results, leads one to infer that the dial gauges in the experiment are measuring the total deflection of the shell, including that due to the bending of the vessel as a whole. Since the method in the thesis calculates local deflection of the shell, it would not reflect the displacement resulting from the bending of the vessel, whose bending as a whole is analogous to that of a beam. As expected, the calculated solution approaches zero at the ends of the vessel. On the other hand, the experimental results do not approach zero. This leads one to believe that the actual supports of the pressure vessel do not reflect a simply-supported boundary condition.

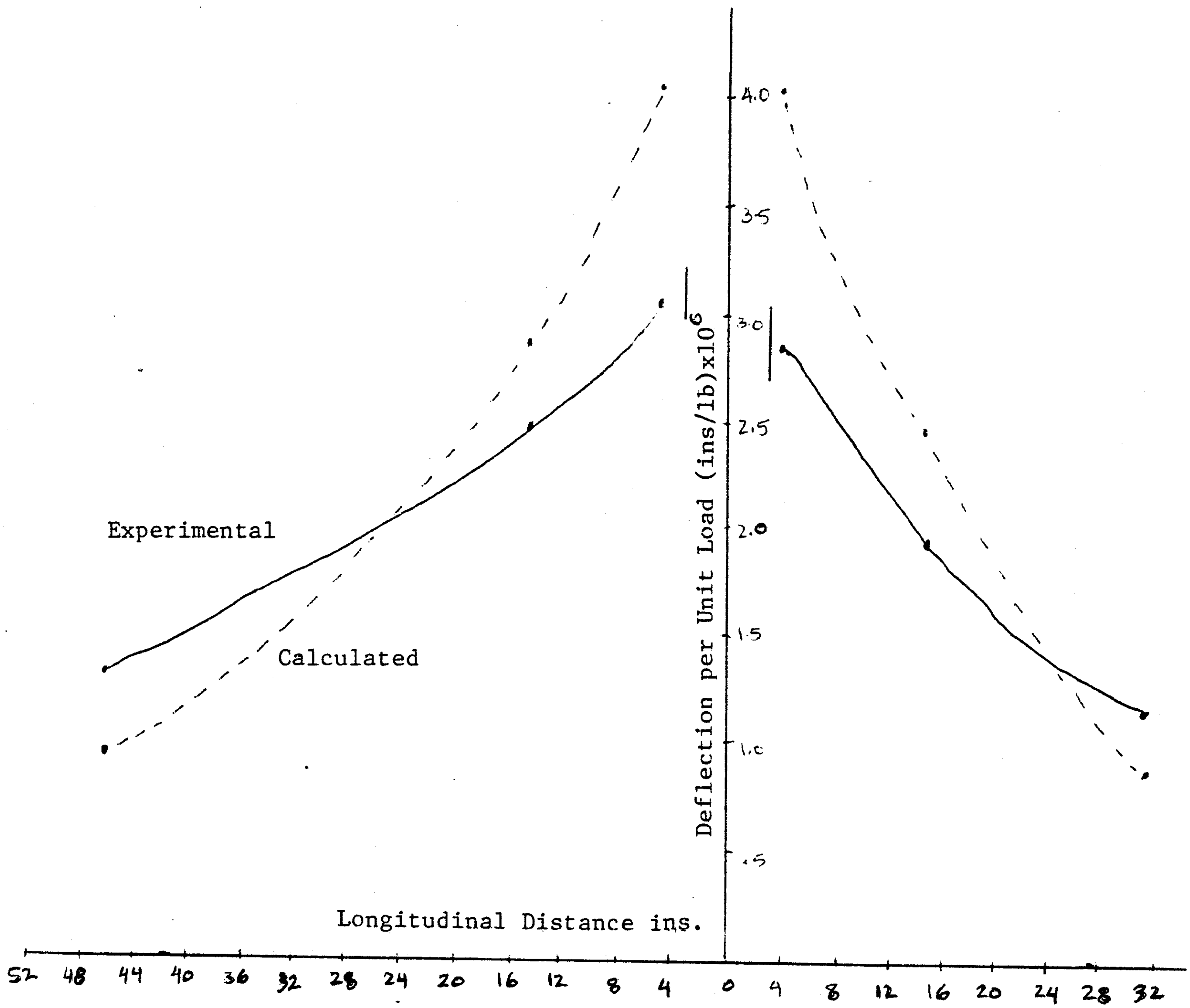


Figure 6.5 Deflection per Unit Radial Load, Attachment 1

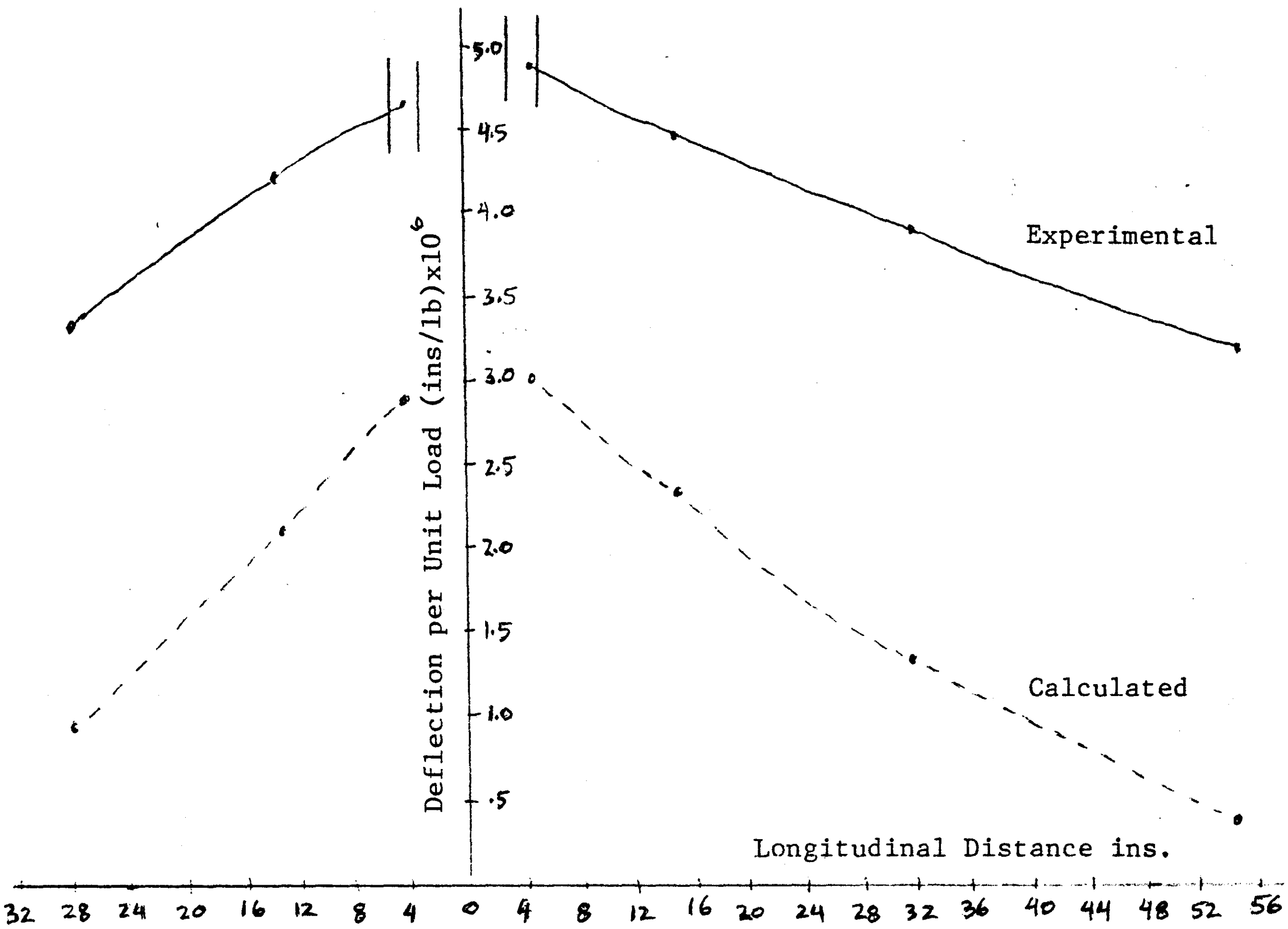


Figure 6.6 Deflection per Unit Radial Load, Attachment 3

7. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE IMPROVEMENTS

7.1 SUMMARY AND CONCLUSIONS

The objective of this thesis was to extend the Fourier method of modelling pressure vessels as continuous simply-connected surfaces with central patch loads. This work has resulted in a closed-form solution for determining stresses and flexibities of pressure vessel attachments. The method's capabilities include placing loads or load combinations anywhere on the cylindrical surface, designating rectangular or circular attachments, examining displacements and stresses resulting from multiple attachments, and simulating rigid or soft attachments.

In comparing the calculations of the thesis with experimental tests in existing literature, there was generally good agreement between my calculated values and those determined from experiments. In comparing the calculations with Bijlaard's results, the calculated values are in good agreement with Bijlaard's values, as they should be.

In simulating experimental tests on an API-650 nozzle located on a storage tank, the discrepancy in the boundary condition primarily accounted for inaccuracies between experimental and calculated data. The calculations assumed a simply-supported base, when in reality, it is fixed. However, the effect of the fixed boundary condition may be lessened by locating the attachment farther from the base or by including a hydrostatic load. For the hydrostatic load, I found the calculated deflections were in good agreement with experimental values, especially when the tank was full. When the tank is full, the hydrostatic load is predominant over the base rigidity.

Despite ambiguities in interpreting the experimental procedure of tests performed on pressure vessel attachments in Welding Research Council Bulletin 60, there was generally good agreement between my calculated stresses and the Bulletin's experimental and theoretical results. The agreement between my calculated values and the Bulletin's theoretical stresses was expected, since both were based on Bijlaard's work in cylindrical shell theory. Because the method in the thesis does not truncate the partial differential equations which govern equilibrium, the calculated stresses were at times closer to the experimental values than were the previous theoretical predictions. Variations between my calculated values and the Bulletin's experimental results were largely due to the stiffening effect of the welds on the attachments and the ambiguity in the method of support for the vessel.

In attempting to compare the method of the thesis with tests in existing literature, one problem that was often encountered was the fact that not many comprehensive tests have been performed. As seen in the Welding Research Council Bulletin 60, many ambiguities arose while trying to accurately simulate the experiments. As a result, one should use these comparisons to evaluate the general trend of the solution, rather than use them for numerical exactness.

7.2 RECOMMENDATIONS FOR FUTURE IMPROVEMENTS

A few recommendations can be made to enhance the method presented in this thesis. An effective convergence algorithm could be incorporated to ensure that the Fourier series has converged. The convergence of the Fourier series is an important factor for thin vessels and attachments with small annular cross sections.

A torsional moment could easily be incorporated into the method. To simulate a torsional moment, one would load each of the elements in the attachment-shell intersection with forces that represent a torsional vector. The displacements and stresses would then be calculated in the usual manner. Torsion could also be included in colocation. As a result, a torsional moment could be applied to a rigid attachment.

The simulation of an external moment could be more accurately represented. The method in the thesis does not account for the linearly varying stress field across the face of an element. It distributes a uniform stress field over the face of each element, whose magnitude is determined by the distance that the center of the element is from the neutral axis. The difference between the linearly varying and uniform stress field for a particular element is a couple. Adding this couple would simulate the moment more accurately. However, using the present approach in the thesis, one would use more elements above and below the neutral axis to obtain a reasonable represetation of the moment. This is an improvement over Bijlaard's method which only used one element above and below the neutral axis.

Finally, a method to simulate a fixed boundary condition would be useful in analyzing tanks or vessels with fixed ends.

APPENDIX A

NOMENCLATURE

Symbol	Description	Units	Defining Equation or Page
D	Flexural rigidity [$D = Et^3/12(1 - \nu^2)$]		p. 7
E	Modulus of elasticity	psi.	p. 7
I_x, I_ϕ	Moments of inertia	in.^{4}	p. 14
L	Length of cylinder	in.	p. 6
M_x, M_ϕ	Radial and tangential moments per unit distance	lb.	(4.4d,e)
$M_{x\phi}$	Twisting moment per unit distance on radial plane	lb.	(4.4f)
N_x, N_ϕ	Radial and tangential forces per unit distance	lb/ft	(4.4a,b)
$N_{x\phi}$	Shearing force per unit distance on axial plane and parallel to ϕ-axis of cylinder	lb/ft	(4.4c)
P_x, P_ϕ, P_r	Surface loadings in axial, tangential, and radial directions	psi	p. 5
a	Radius of cylinder	in.	p. 6
b, β	Dimensions of rectangular element	in.	p. 7
g	Acceleration of gravity	ft/s^2	p. 30
m, n	integers, Fourier indices	–	(4.1)
t	Thickness of cylindrical shell	in.	p. 5
u, v, w	Displacements in axial, tangential, and radial directions	in.	p. 5
x, ϕ, r	Cylindrical coordinates	–	p. 5

ν	Poisson's ratio	–	(2.1)
ρ	Density	lbs/in^3	p. 30
σ_x, σ_ϕ, σ_r	Radial, tangential, and axial normal stress	psi	p. 28
τ	Shear stress	psi	p. 28

APPENDIX B : MATHEMATICAL DETAILS

SUMMARY OF ORTHOGONAL FUNCTIONS

$$\int_0^{2\pi} \sin(m'\phi)\sin(m\phi)\, d\phi = \begin{cases} 0 & m' = m \\ \pi & m' = m \end{cases}$$

$$\int_0^{2\pi} \cos(m'\phi)\cos(m\phi)\, d\phi = \begin{cases} 0 & m' = m \\ \pi(1 + \delta_{m,0}) & m' = m \end{cases}$$

$$\int_0^{L} \sin(n'\pi x/L)\sin(n\pi x/L)\, dx = \begin{cases} 0 & n' = n \\ L/2 & n' = n \end{cases}$$

$$\int_0^{L} \cos(n'\pi x/L)\cos(n\pi x/L)\, dx = \begin{cases} 0 & n' = n \\ L(1 + \delta_{n,0})/2 & n' = n \end{cases}$$

$\delta_{i,0}$ = Kronecker Delta

INTEGRALS USED TO DETERMINE FOURIER COEFFICIENTS FOR RECTANGULAR AREAS

$$\int_{x_o - b}^{x_o + b} \cos(n\pi x/L)\, dx = \frac{2L}{n\pi} \cos(n\pi x/L)\sin(n\pi b/L)$$

$$\int_{x_o - b}^{x_o + b} \sin(n\pi x/L)\, dx = \frac{2L}{n\pi} \sin(n\pi x/L)\sin(n\pi b/L)$$

$$\int_{\phi_o - \beta}^{\phi_o + \beta} \cos(m\phi)\, d\phi = \frac{2}{m} \cos(m\phi)\sin(m\beta)$$

$$\int_{\phi_o - \beta}^{\phi_o + \beta} \sin(m\phi)\, d\phi = \frac{2}{m} \sin(m\phi)\sin(m\beta)$$

USEFUL TRIGONOMETRIC IDENTITIES

$$\sin(A + B) = \sin A \cos B + \cos A \sin B$$
$$\sin(A - B) = \sin A \cos B - \cos A \sin B$$

$$\cos(A + B) = \cos A \cos B - \sin A \sin B$$
$$\cos(A - B) = \cos A \cos B + \sin A \sin B$$

$$\cos(Ax)\cos(Bx) = 1/2\, [\, \cos(A + B)x + \cos(A - B)x \,]$$
$$\sin(Ax)\sin(Bx) = 1/2\, [\, \cos(A - B)x - \cos(A + B)x \,]$$
$$\sin(Ax)\cos(Bx) = 1/2\, [\, \sin(A - B)x + \sin(A + B)x \,]$$

INTEGRALS USED TO DETERMINE FOURIER COEFFICIENTS FOR TRIANGULAR AREAS

$$\int_{\phi_a}^{\phi_b} \cos(m\phi)\cos(k\phi)\, d\phi = 1/2 \int_{\phi_a}^{\phi_b} [\cos(m + k)\phi + \cos(m - k)\phi]\, d\phi$$

$$= \frac{1}{2}\left[\frac{\sin(m + k)\phi}{m + k} + \frac{\sin(m - k)\phi}{m - k}\right]_{\phi_a}^{\phi_b}$$

$$\int_{\phi_a}^{\phi_b} \cos(m\phi)\sin(k\phi)\, d\phi = 1/2 \int_{\phi_a}^{\phi_b} [\sin(m + k)\phi - \sin(m - k)\phi]\, d\phi$$

$$= \frac{1}{2}\left[-\frac{\cos(m + k)\phi}{m + k} + \frac{\cos(m - k)\phi}{m - k}\right]_{\phi_a}^{\phi_b}$$

$$\int_{\phi_a}^{\phi_b} \sin(m\phi)\cos(k\phi)\, d\phi = 1/2 \int_{\phi_a}^{\phi_b} [\sin(m + k)\phi + \sin(m - k)\phi]\, d\phi$$

$$= \frac{1}{2}\left[-\frac{\cos(m + k)\phi}{m + k} - \frac{\cos(m - k)\phi}{m - k}\right]_{\phi_a}^{\phi_b}$$

$$\int_{\phi_a}^{\phi_b} \sin(m\phi)\sin(k\phi)\, d\phi = 1/2 \int_{\phi_a}^{\phi_b} [-\cos(m + k)\phi + \cos(m - k)\phi]\, d\phi$$

$$= \frac{1}{2}\left[-\frac{\sin(m + k)\phi}{m + k} + \frac{\sin(m - k)\phi}{m - k}\right]_{\phi_a}^{\phi_b}$$

APPENDIX C

ATTACHMENT NO. 5 OF WELDING RESEARCH COUNCIL BULLETIN 60

Attachment No. 5 of Welding Research Council Bulletin 60 consisted of a solid circular steel rod inserted through a hole in the vessel and welded to the surface. The radius of the rod was 1.75 inches and the average weld size was 17/32 inches. This attachment was located 65 inches from the left hand side of the vessel or at the same longitundinal location as Attachment 1 (See Figure 6.1). The major difficulty in simulating Attachment 5 is that it is a true rigid body. Stresses on Attachments 1 and 3 were calculated in the usual manner, as soft attachments. This procedure assumes that the attachment-shell junction is flexible. However, to accurately model Attachment 5, this junction is must be rigid. As a result, the method of colocation would be well suited to this application. The data presented here is the result of simulating Attachment 5 as a soft attachment and is included for completeness.

Tables M shows the bending and membrane stresses at the edge of the attachment due to a radial load. While at times the calculated values are within the same order of magnitude as the experimental results, they are usually lower than the experimentally determined stresses. This discrepancy is primarily caused by rigidity of the attachment. If the colocation method was employed, then one would expect the stresses to become higher near edge of the attachment, since the attachment-shell junction would be rigid. Table N shows the bending and membrane stresses at the edge of the attachment due to longitudinal and circumferential couples. As before, due to the discrepancy in simulating the rigidity of the attachment, the calculated stresses are usually lower than experimental values.

The deflection per unit radial load of Attachment 5 is shown in Figure C.1. The calculated deflections are lower than the experimentally determined values. However, the slope of the both curves is consistent. Since the cross section of the attachment-shell intersection is large compared to the size of the welds, the welds are not a significant consideration in the discrepancy between calculated and experimental results. Rather, as seen in the deflection of Attachment 3, the difference between the two curves is due to the fact that the experimental data reflects the total deflection of the vessel as a whole. The method in this thesis only calculates the local deflection in the shell.

TABLE M - Attachment No. 5, Radial Load, Bending and Membrane Stress per Unit Force (psi./lbs.), P = 10,300 lbs.

Method	N_x/Pt	$6M_x/Pt^2$	N_ϕ/Pt	$6M_\phi/Pt^2$
Meridian:				
Experiment	-0.26	-1.35	-0.55	-2.96
Calculated	-0.29	-0.94	-0.30	-1.11
Bijlaard	-0.44	-1.80	-0.46	-2.32
Generator:				
Experiment	-0.44	-1.49	-0.17	-1.01
Calculated	-0.30	-0.83	-0.24	-1.46
Bijlaard	-0.44	-1.80	-0.46	-2.32

TABLE N - Attachment 5, Longitudinal and Circumferential Couples, Bending and Membrane Stress per Unit Couple (psi./in-lbs)

Method	N_x/Pt	$6M_x/Pt^2$	N_ϕ/Pt	$6M_\phi/Pt^2$
Longitudinal Moment ($P = 5.412 \times 10^4$)				
Generator:				
Experiment	0.098	0.82	0.10	0.44
Calculated	0.003	0.10	0.01	0.05
Bijlaard	0.031	0.93	0.12	0.55
Circumferential Moment (P = 35,520 in-lbs.)				
Meridian:				
Experiment	0.039	0.57	0.032	1.05
Calculated	0.000	0.39	0.065	0.12
Bijlaard	0.040	0.60	0.063	1.05

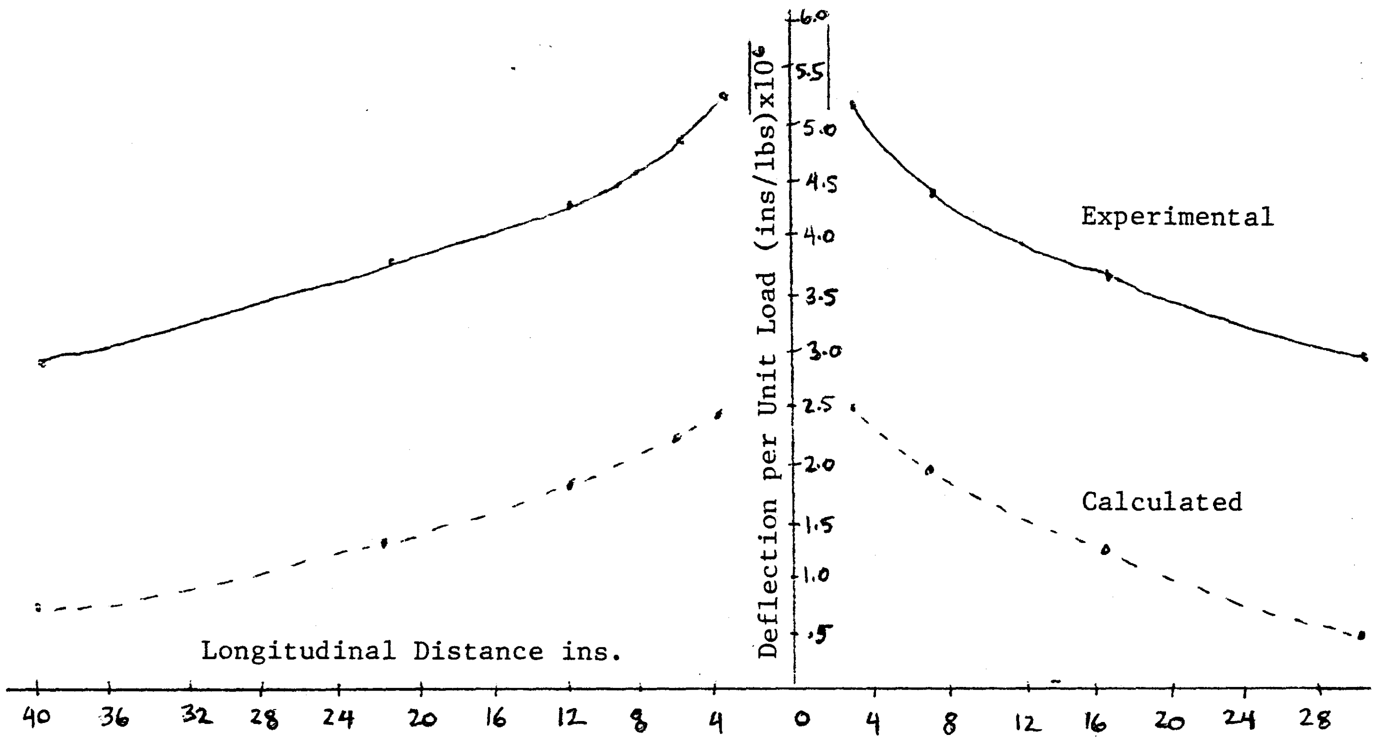


Figure C.1 Deflection per Unit Load,
Attachment 5

APPENDIX D

COMPUTER PROGRAM LISTINGS

```
C     *****************************************************************
      PROGRAM BIJLAARD
C
C     DRIVER PROGRAM TO CALCULATE LOADS, DEFLECTIONS, AND STRESSES
C     OF CYLINDERS DUE TO RADIAL LOADS
C
C     CREATED                        :   15-JUL-83   BY:  W. CRAFT
C     DIVIDED INTO SUBROUTINES       :   18-JUL-83   BY:  F.M.G. WONG
C     INTERNAL PRESSURE INCLUDED     :   10-AUG-83   BY:  F.M.G. WONG
C     EXTERNAL MOMENTS INCLUDED      :   02-SEP-83   BY:  F.M.G. WONG
C     COLOCATION INCLUDED            :   07-OCT-83   BY:  F.M.G. WONG
C     MULTIPLE NOZZLES INCLUDED      :   12-NOV-83   BY:  F.M.G. WONG
C     TRIANGULAR PATCHES INCLUDED    :   22-NOV-83   BY:  F.M.G. WONG
C     HYDROSTATIC PRESS. INCLUDED    :   14-DEC-83   BY:  F.M.G. WONG
C
C
C      LATEST REVISION               :   19-JAN-84
C
C     *****************************************************************
C
      REAL L, NU,LPHI,LX
      CHARACTER*9 CDATE
      CHARACTER*8 CTIME
      DIMENSION PLAT(4,4), FOR(3), NDIR(3)
      COMMON / CONSTANTS / A, E, L, NU, PI, T
      COMMON / INDEXES / M, N
       COMMON / LOADS / PRE(100,100), PRO(100,100), PXE(100,100),
     1                  PXO(100,100), PPHIE(100,100), PPHIO(100,100)
      COMMON / DISPL / UE(100,100), UO(100,100), VE(100,100), VO(100,100),
     1                 WE(100,100), WO(100,100)
      COMMON / INTPRS / PRESS(100), QPRES
      COMMON / CENTELM / CNTXEL(4,4,4), CNTPHIEL(4,4,4)
      COMMON / CIRCTB / CIRCX(7,4), CIRCPHI(7,4)
      COMMON / CIRCSD / CIRCSX(3,4), CIRCSP(3,4)
      COMMON / STIFMAT / STIFFA(8,8), STIFFB(32,32), STIFFC(72,72),
     1                    STIFFD(128,128)
C
      DATA PRE/10000*0/,PRO/10000*0/,PXE/10000*0/,PXO/10000*0/
      DATA PPHIE/10000*0/,PPHIO/10000*0/
C
      PI=3.14159265359
C
      OPEN(UNIT=1,FILE='BIJLRD.DAT',CARRIAGECONTROL='LIST',
     1 STATUS='NEW')
      OPEN (UNIT=4, FILE='AUTOZ.DAT', STATUS = 'OLD')
C
      CALL DATE(CDATE)
      CALL TIME(CTIME)
C
C     *****************************************************************
C
C     THIS ROUTINE ROLLS CYLINDER INTO A FLAT SURFACE OF SIZE L*(2PI*A) OR
C     SMALLER AND CHECKS LOAD INTENSITY DUE TO A COMBINATION OF AXIAL,
C     SHEAR, RADIAL LOADS OF ANY INTENSITY AND LOCATION.
C
C     A = CYLINDER RADIUS (INCHES)
C     L = CYLINDER LENGTH (INCHES)
```

```
C       M = SUMMATION LIMIT PHI HARMONICS
C       N = SUMMATION LIMIT X HARMONICS
C       T = CYLINDER WALL THICKNESS (INCHES)
C       E = YOUNGS MODULUS FOR CYLINDER (PSI)
C       NU = POISSONS RATIO FOR CYLINDER (DIM'LESS)
C
C       X AND PHI ARE CYLINDER COORDINATES (X INCHES, PHI INPUT IN DEGREES)
C                 (XC,PHIC) IS CENTER OF TUBE CONNECTION TO CYLINDER
C
C       TAUX = TUBE WALL THICKNESS IN THE X DIRECTION (INCHES)
C       TAUPHI = TUBE WALL THICKNESS IN THE HOOP DIRECTION (INCHES)
C
C       (CORX,CORPHI) ARE OUTER DIMENSIONS OF TUBE IN THE
C                 AXIAL AND HOOP DIRECTIONS (EACH IN INCHES)
C
C       MTUB = THE NUMBER OF LOADING PATCH ELEMENTS IN ONE QUARTER OF
C              THE TUBE WALL IN THE HOOP DIRECTION
C
C       NTUB = THE NUMBER OF LOADING PATCH ELEMENTS IN ONE QUARTER OF
C               THE TUBE WALL IN THE AXIAL DIRECTION
C
C       F = THE LATERAL FORCE IMPOSED ON THE CYLINDER BY THE ATTACHMENT
C           TUBE (LBS.)
C
C       QPRES = INTERNAL PRESSURE OF CYLINDER (PSI)
C
C       RMOM = EXTERNAL APPLIED MOMENT
C
C       *****************************************************************
C
C
        WRITE(1,40) CDATE,CTIME
40      FORMAT(15X,'OUTPUT FILE FOR PROGRAM BIJLAARD',2X,A9,2X,A8,/)
C
C       :::::::
C       ::::::: INITIAL CYLINDER PARAMETERS
C       :::::::
C
        WRITE(6,13)
        WRITE(1,13)
13      FORMAT(1X,'INPUT CYLINDER PARAMETERS: A,L,M,N,T,E,NU WHERE',/,
     1  1X,'A = RADIUS, L = LENGTH, M,N<=100,100: AS SERIES SUMS,',/,
     2  1X,'T = THICKNESS, E = MODULUS OF ELASTICITY,',/,
     3  1X,'NU = POISSON''S RATIO :',/)
C
        READ(4,10) A, L, M, N, T, E, NU
10      FORMAT (1X,2E11.4,2I4,3E11.4)
        WRITE(1,42) A, L, M, N, T, E, NU
42      FORMAT(1X,'A, L, M, N, T, E, NU=',2E11.4,2I4,3E11.4)
C
C       :::::::
C       :::::::  INITIAL ATTACHMENT DIMENSIONS
C       :::::::
C
        WRITE (6,61)
        WRITE (1,61)
```

```
61      FORMAT (/,1X,'MULTIPLE NOZZLES: YES = 1, NO = 0')
        READ (4,201) MULNZ
        WRITE (6,63) MULNZ
        WRITE (1,63) MULNZ
63      FORMAT (1X,'MULTIPLE NOZZLES = ',I1)
        IF ( MULNZ .EQ. 1 ) THEN
                CALL MULTNOZ
                GO TO 49
        ENDIF
C
        WRITE (6,65)
        WRITE (1,65)
65      FORMAT (/,1X,'DESIGNATION OF CIRCULAR ATTACHMENTS: YES=1, NO=0')
        READ (4,201) ICIRC
        WRITE (6,66) ICIRC
        WRITE (1,66) ICIRC
66      FORMAT (1X,'ICIRC = ',I1)
        IF ( ICIRC .NE. 1 ) GO TO 50
        WRITE (6,67)
        WRITE (1,67)
67      FORMAT (/,1X,'ENTER CIRCULAR PARAMETERS: XC, PHIC, R1, R2, NFOR')
        READ (4,68) XC, PHIC, R1, R2, NFOR
68      FORMAT (1X,4E11.4,1X,I1)
        WRITE (6,68) XC, PHIC, R1, R2 ,NFOR
        WRITE (1,68) XC, PHIC, R1, R2 ,NFOR
        DO 69 K = 1, NFOR
                WRITE (6,70) K
                WRITE (1,70) K
70              FORMAT (/,'ENTER FORCE: ',I1,' MAGNITUDE & DIRECTION ')
                READ (4,71) FOR(K), NDIR(K)
71              FORMAT(1X,E11.4,1X,I1)
                WRITE (6,71) FOR(K), NDIR(K)
                WRITE (1,71) FOR(K), NDIR(K)
69      CONTINUE
        GO TO 200
C
50      WRITE(6,18)
        WRITE(1,18)
18      FORMAT(/,1X,'INPUT TUBE CENTER (XC,PHIC), WALL THICKNESSES,',/,
     1   1X,'TAUX, TAUPHI, THEN TUBE SECTIONS, LX, LPHI',/,
     2   1X,'THEN NO. OF QUARTER-TUBE ELEMENTS, MTUB, NTUB, AND LATERAL',
     3   1X,'FORCE, F, IN POUNDS',/)
C
        WRITE(6,15)
        WRITE(1,15)
15      FORMAT(1X,'INPUT XC, PHIC, TAUX, TAUPHI, LX, LPHI, MTUB, NTUB, F',/,
     1   1X,'PHIC IN DEGREES.',/)
        READ(4,11) XC, PHIC, TAUX, TAUPHI, LX, LPHI, MTUB, NTUB, F
11      FORMAT (1X,6E11.4,2I3,E11.4)
        WRITE(6,44) XC, PHIC, TAUX, TAUPHI, LX, LPHI, MTUB, NTUB, F
        WRITE(1,44) XC, PHIC, TAUX, TAUPHI, LX, LPHI, MTUB, NTUB, F
44      FORMAT(1X,'XC, PHIC, TAUX, TAUPHI, LX, LPHI, MTUB, NTUB, F =',/,
     1   1X,6E11.4,2I3,E11.4,/)
C
        PHIC=PI*PHIC/180.
C
```

```
C       ::::::: EXTERNALLY APPLIED MOMENTS
C
200     WRITE (6,20)
        WRITE (1,20)
20      FORMAT (1X,'EXTERNALLY APPLIED MOMENTS: YES = 1, NO = 0')
        READ (4,201) MOMRPY
201     FORMAT (1X,I1)
        WRITE (6,21) MOMRPY
        WRITE (1,21) MOMRPY
21      FORMAT (1X,'RESPONSE TO MOMENT PROMPT = ',I,/)
        IF ( MOMRPY .EQ. 0 ) GO TO 49
        WRITE (6,22)
        WRITE (1,22)
22      FORMAT (1X,'TYPE OF MOMENT: LONG. = 1, CIRCUM. = 2, TORSION = 3')
        READ (4,201) MOMTYP
        WRITE (6,23) MOMTYP
        WRITE (1,23) MOMTYP
23      FORMAT (1X,'TYPE OF MOMENT = ',I,/)
        WRITE (6,24)
        WRITE (1,24)
24      FORMAT (1X,'ENTER EXTERNALLY APPLIED MOMENT (IN.-LBS.):')
        READ (4,25) RMOM
25      FORMAT (1X,E11.4)
        IF (MOMTYP .EQ. 3 ) TMOM = RMOM
        WRITE (6,26) RMOM
        WRITE (1,26) RMOM
26      FORMAT (1X,'EXTERNALLY APPLIED MOMENT (IN.-LBS.) = ',E11.4,/)
C
C       ::::::: INTERNAL PRESSURE PARAMETER
C
49      WRITE (6,491)
        WRITE (1,491)
491     FORMAT (1X,'ENTER TYPE OF INTERNAL PRESS. 1=RAD; 2=HYDRO.; 0=NONE')
        READ (4,201) INPRTYP
        WRITE (6,490) INPRTYP
        WRITE (1,490) INPRTYP
490     FORMAT (1X,'INTERNAL PRESSURE TYPE = ',I1)
        IF ( INPRTYP .EQ. 1 ) THEN
                WRITE (6,14)
                WRITE (1,14)
14              FORMAT (1X,'ENTER VALUE OF INTERNAL PRESSURE (PSI) :')
                READ (4,12) QPRES
12              FORMAT (1X,F7.2)
                WRITE (6,48) QPRES
                WRITE (1,48) QPRES
48              FORMAT (1X,'INTERNAL PRESSURE =',F7.2,/)
        ENDIF
        IF ( INPRTYP .EQ. 2 ) THEN
                WRITE (6,481)
                WRITE (1,481)
481             FORMAT (1X,'ENTER VALUES FOR RHOG, XO :')
                READ (4,482) RHOG, XO
482             FORMAT (1X,E11.4,1X,E11.4)
                WRITE (6,483) RHOG, XO
                WRITE (1,483) RHOG, XO
483             FORMAT (1X,'RHOG = ',E11.4,'  XO = ',E11.4,/)
```

```
      ENDIF
C
C     :::::::  COLOCATION METHOD FOR RIGID PLUG
C
      WRITE (6,31)
      WRITE (1,31)
31    FORMAT (1X,'WISH TO USE COLOCATION METHODS?  YES = 1, NO = 0')
      READ (4,201) ICOLC
      WRITE (6,32) ICOLC
      WRITE (1,32) ICOLC
32    FORMAT (1X,'ICOLC = ',I1,/)
C
      IF ( ICOLC .EQ. 1 ) THEN
              IF ( ICIRC .EQ. 1 ) THEN
                      CALL COLOCATE (XC, PHIC, R2, 0.0, 0.0, 0.0, 0, 0, 2)
                      GO TO 99
              ENDIF
              CALL COLOCATE (XC, PHIC, TAUX, TAUPHI, LX, LPHI, MTUB, NTUB, 1)
              GO TO 99
      ENDIF
C
C     ::::::: DISPLACEMENT SOLUTION TYPE
C
      WRITE (6,80)
      WRITE (1,80)
80    FORMAT(/,1X,'ENTER DISPLACEMENT SOLUTION TYPE:')
      READ (4,201) IDIS
      WRITE (6,201) IDIS
      WRITE (1,201) IDIS

C
C     ::::::: CALCULATE LATERAL PRESSURES
C
      IF ( INPRTYP .NE. 0 )   CALL INTPRES (INPRTYP, RHOG, XO)
C
      IF ( (ICIRC .EQ. 1) .AND. (NFOR .NE. 0) ) THEN
              DO 85 K = 1, NFOR
              PRS = FOR(K)/((R2*R2 - R1*R1)*PI)
              PHIC = PHIC*PI/180.
              CALL CIRCSPLIT (XC, PHIC, R1, R2, PRS, NDIR(K))
85            CONTINUE
              GO TO 89
      ENDIF
C
      IF ( F .NE. 0. ) THEN
      PL = F/( (LX*LPHI) -(LX - 2.*TAUX)*(LPHI - 2.*TAUPHI) )
      PRINT *, ' PL =',PL
      CALL RECTUBE (PL, XC, PHIC, TAUX, TAUPHI, LX, LPHI, MTUB, NTUB, IKP)
      IF ( IKP .EQ. 1 ) GO TO 50
      ENDIF
C
89    IF ( MOMRPY .EQ. 1 ) THEN
              IF ( ICIRC .EQ. 1 ) THEN
                      PHIC = PHIC*PI/180.
                      IF ( MOMTYP .EQ. 3 )  THEN
                              CALL TORSION (XC, PHIC, R1, R2, TMOM)
```

```
                        GO TO 90
                    ENDIF
                    CALL CIRCSET (XC, PHIC, R1, R2, RMOM, MOMTYP)
                    GO TO 90
                ENDIF
                CALL MOMENTSET (RMOM, XC, PHIC, TAUX, TAUPHI, LX, LPHI, MTUB,
     1                          NTUB, IKP, MOMTYP)
                IF ( IKP .EQ. 1 )  GO TO 50
            ENDIF
C
C
C       ::::::: CALCULATE LOAD MAP FOR THIS PATCH
C
90      CALL LOADMAP (MAP, N1)
        IF (MAP.NE.3) GO TO 50
        IF ( INPRTYP .NE. 0.0 ) CALL MERGPRES
C
C       ::::::: CALCULATE DISPLACEMENTS AND DISPLACEMENT MAP FOR THIS PATCH
C
        CALL DISPLACE (IDIS)
        CALL DISPMAP
C
C       ::::::: CALCULATE STRESSES AND STRESS MAP FOR THIS PATCH
C
        CALL STRESS
C
99      STOP
        END
C
C
```

```
C       ********************************************************************
C
        SUBROUTINE PATCH (XO, PHIO, B, BETA, F, N1)
C
C       CALCULATES THE LOADS OF A RECTANGULAR PATCH
C
C       CREATED:  26-JUL-83        BY: W. CRAFT
C       ********************************************************************
C
        REAL L, NU, LXEL, LPHIEL
        COMMON / CONSTANTS / A, E, L, NU, PI, T
        COMMON / INDEXES / M, N
        COMMON / LOADS / PRE(100,100), PRO(100,100), PXE(100,100),
     1                   PXO(100,100), PPHIE(100,100), PPHIO(100,100)
C
        PHIOD=180.*PHIO/PI
        BETAD=180.*BETA/PI
        LXEL=2.*B
        LPHIEL=2.*BETAD
        IF (F.EQ.0) GOTO 99
        PI2=PI*PI
        DO 3 MT = 0, M-1
                IF (MT.EQ.0) GO TO 5
                XMT=FLOAT(MT)
                SBETA=SIN(XMT*BETA)
                CPHIO=COS(XMT*PHIO)
                SPHIO=SIN(XMT*PHIO)
5               CONTINUE
                DO 2 NT = 1, N
                        XNT=FLOAT(NT)
                        SB=SIN(XNT*PI*B/L)
                        IF ((N1.EQ.1).OR.(N1.EQ.2)) SXO=SIN(XNT*PI*XO/L)
                        IF (N1.EQ.3) CXO=COS(XNT*PI*XO/L)
                        IF (MT.GT.0) GO TO 4
                        D=8./(XNT*PI2)
                        PHI=D*F*BETA
                        IF (N1.EQ.1) PRE(MT+1,NT)=PRE(MT+1,NT)+PHI*SXO*SB
                        IF (N1.EQ.2) PPHIO(MT+1,NT)=PPHIO(MT+1,NT)+PHI*SXO*SB
                        IF (N1.EQ.3) PXE(MT+1,NT)=PXE(MT+1,NT)+PHI*CXO*SB
                        GO TO 2
4                       C=16./(XMT*XNT*PI2)
                        IF (N1.GT.1) GO TO 20
                        PRE(MT+1,NT)=PRE(MT+1,NT)+C*F*CPHIO*SBETA*SXO*SB
                        PRO(MT+1,NT)=PRO(MT+1,NT)+C*F*SPHIO*SBETA*SXO*SB
                        GO TO 2
20                      IF (N1.GT.2) GO TO 30
                        PPHIE(MT+1,NT)=PPHIE(MT+1,NT)+C*F*SPHIO*SBETA*SXO*SB
                        PPHIO(MT+1,NT)=PPHIO(MT+1,NT)+C*F*CPHIO*SBETA*SXO*SB
                        GO TO 2
30                      PXE(MT+1,NT)=PXE(MT+1,NT)+C*F*CPHIO*SBETA*CXO*SB
                        PXO(MT+1,NT)=PXO(MT+1,NT)+C*F*SPHIO*SBETA*CXO*SB
2               CONTINUE
3       CONTINUE
C
99      RETURN
        END
C
```

```
C
C     **********************************************************************
C
      SUBROUTINE LOADMAP (MAP, N1)
C
C     WRITES THE LOAD MAP
C
C     CREATED:  18-JUL-83         BY: W. CRAFT
C
C     **********************************************************************
C
      REAL L, NU
      DIMENSION XMAT(11),PHIMAT(11),INT(11,11), RINT(11,11)
      COMMON / CONSTANTS / A, E, L, NU, PI, T
      COMMON / INDEXES / M, N
       COMMON / LOADS / PRE(100,100), PRO(100,100), PXE(100,100),
     1                  PXO(100,100), PPHIE(100,100), PPHIO(100,100)
      COMMON / INTPRS / PRESS(100), QPRES
C
C
11    WRITE(6,25)
      WRITE(1,25)
25    FORMAT(/,1X,'DO YOU WANT A NEW LOAD MAP? 1=YES, 2=NO, 3=END')
      READ(4,26) MAP
26    FORMAT (1X,I1)
      WRITE(1,46) MAP
46    FORMAT(1X,'MAP=',I1,/)
      IF (MAP.NE.1) GO TO 100
      WRITE(6,14)
      WRITE(1,14)
14    FORMAT(1X,'INPUT XST, DXST, PHIST, DPHIST, N1 (ANGLES/DEG)',/)
      READ(4,15) XST, DXST, PHIST, DPHIST, N1
15    FORMAT (1X,4E11.4,I2)
      WRITE(1,47) XST, DXST, PHIST, DPHIST, N1
47    FORMAT(1X,'XST, DXST, PHIST, DPHIST, N1=',4E11.4,I2,/)
C
      PHIST=PI*PHIST/180.
      DPHIST=PI*DPHIST/180.
C
      CALL TIMESTAT (1)
C
      DO 10 I=1,11
              XCO=XST+DXST*FLOAT(I-6)/10.
              XMAT(I)=XCO
              DO 12 J=1,11
                      PHICO=PHIST+DPHIST*FLOAT(J-6)/10.
                      IF (I.EQ.1) PHIMAT(J)=PHICO*180./PI
C
C      ::::::: NOW COMPUTE THE LOADS ON THE SURFACE MAP AND PRESENT THE
C      ::::::: RESULTS AS 0 TO 100% OF F AT EACH GRID POINT AND PLOT.
C
                      XINT=0.
                      PRSINT = 0.
                      DO 6 MT=0,M-1
                      SPHI=SIN(FLOAT(MT)*PHICO)
                      CPHI=COS(FLOAT(MT)*PHICO)
```

```
      DO 6 NT=1,N
      IF ( (QPRES .NE. 0.0) .AND. (MT .EQ. 0) )
     1     PRSINT = 2.*PRESS(NT)*SIN(FLOAT(NT)*PI*XCO/L)
      IF (N1.EQ.1) XINC=(PRE(MT+1,NT)*CPHI+PRO(MT+1,NT)
     1 *SPHI)*SIN(FLOAT(NT)*PI*XCO/L)
      IF (N1.EQ.2) XINC=(PPHIE(MT+1,NT)*SPHI+PPHIO(MT+1,NT)
     1 *CPHI)*SIN(FLOAT(NT)*PI*XCO/L)
      IF (N1.EQ.3) XINC=(PXE(MT+1,NT)*CPHI+PXO(MT+1,NT)
     1 *SPHI)*COS(FLOAT(NT)*PI*XCO/L)
      XINT=XINT+XINC
      IF ( (QPRES .NE. 0.0) .AND. (MT .EQ. 0) )
     1   XINT = XINT - PRSINT
6     CONTINUE
      XINT=XINT*.5
      INT(I,J) = NINT(XINT)
      RINT(I,J) = XINT
12    CONTINUE
C
      CALL TIMESTAT (2)
10    CONTINUE
      CALL TIMESTAT (3)
C
      CALL FINDMIN (RINT, RMIN, RMAX)
C
      CALL DIVTEN (RINT, RMIN, RMAX, EXPT)
      IF ( EXPT .EQ. 0.0 ) EXPT = 1.0
C
C      ::::::: NOW THE INTEGER ARRAY IS SETUP FOR LOADS
C
      IF ( LOG10(EXPT) .GT. 4.0 .OR. LOG10(EXPT) .LT. 0.0 ) THEN
          WRITE (6,18) EXPT
          WRITE (1,18) EXPT
18        FORMAT (15X,'---> VALUES IN MAP SCALED AS ',E9.2,/)
      ENDIF
      WRITE(6,17) (PHIMAT(K),K=1,11)
      WRITE(1,17) (PHIMAT(K),K=1,11)
17    FORMAT(1X,'OUTPUT OF LOADING ON GRID IS:',/,7X,11F6.1,/)
C
      DO 9 NN=1,11
          IF ( LOG10(EXPT) .GT. 4.0 .OR. LOG10(EXPT) .LT. 0.0 ) THEN
              WRITE(6,19) XMAT(12-NN),(RINT(12-NN,K),K=1,11)
              WRITE(1,19) XMAT(12-NN),(RINT(12-NN,K),K=1,11)
19            FORMAT(1X,F6.2,11F8.4)
              ELSE
              WRITE(6,191) XMAT(12-NN),(INT(12-NN,K),K=1,11)
              WRITE(1,191) XMAT(12-NN),(INT(12-NN,K),K=1,11)
191           FORMAT(1X,F6.2,11I6)
          ENDIF
9     CONTINUE
      GO TO 11
C
100   RETURN
      END
C
C
```

```
C
C     ********************************************************************
C
      SUBROUTINE RECTUBE (PL, XC, PHIC, TAUX, TAUPHI, LX, LPHI, M, N, IKP)
C
C     RECTUBE IS A SUBROUTINE THAT PREPARES A RECTANGULAR LOAD
C                             THROUGH 4 CALLS TO PATCHLAT
C
C     PLAT   = LATERAL PRESSURE AT EACH ELEMENT (PSI)
C     XC     = X-COORDINATE OF TUBE CENTER (INCHES)
C     PHIC   = PHI-COORDINATE OF TUBE CENTER (RADIANS)
C     TAUX   = THICKNESS OF ATTACHMENT TUBE ALONG X (INCHES)
C     TAUPHI = THICKNESS OF ATTACHMENT TUBE ALONG PHI (INCHES)
C     LPHI   = OUTER HORIZONTAL DIMENSION (INCHES) OF ATTACHMENT TUBE
C     LX     = OUTER VERTICAL DIMENSION (INCHES) OF ATTACHMENT TUBE
C
C     M,N ARE ELEMENT MEMBERS OF PATCH IN PHI AND X DIRECTIONS, RESP.
C
C     CREATED              :  26-JUL-83      BY: W. CRAFT
C     LATEST REVISION      :  19-AUG-83
C
C     ********************************************************************
C
      REAL L,NU,LPHI,LX
      DIMENSION PLAT(4,4)
      COMMON / CONSTANTS / A, E, L, NU, PI, T
C
C
      ALPO = ASIN(LPHI/(2.*A))
      ALPI = ASIN((LPHI - 2.*TAUPHI)/(2.*A))
      V1 = .5*(ALPO + ALPI)
      V2 = LX-2.*TAUX
      V3 = .5*(LX - TAUX)
C
      DO 1 I = 1,M

              DO 1 J = 1,N
1             PLAT(I,J) = PL
C
      ANG = ALPO - ALPI
      V4 = 2.*ALPO
C
      CALL PATCHLAT (M, N, XC, PHIC, 0, V1, V2, ANG, PLAT, IKP)
      CALL PATCHLAT (M, N, XC, PHIC, V3, 0, TAUX, V4, PLAT, IKP)
      CALL PATCHLAT (M, N, XC, PHIC, 0, -V1, V2, ANG, PLAT, IKP)
      CALL PATCHLAT (M, N, XC, PHIC, -V3, 0, TAUX, V4, PLAT, IKP)
C
      RETURN
      END
C
```

```
C
C     ******************************************************************
C
      SUBROUTINE PATCHLAT (M, N, XC, PHIC, XL, PHIL, THX, THPHI, PLAT, IKP)
C
C     PATCHLAT ALLOWS PARTITIONING OF LOADS IN LOCAL COORDINATES TO MATCH
C     LATERAL COMPONENTS
C
C     M          = NO. OF PHI DIRECTION ELEMENTS IN PATCH
C     N          = NO. OF X DIRECTION ELEMENTS IN PATCH
C     XC         = LOCAL PATCH REFERENCE X COORDINATE
C     PHIC       = LOCAL PATCH REFERENCE PHI COORDINATE
C     (XC,PHIC) DEFINE DIRECTION OF LATERAL LOAD AXIS ON CYLINDER
C     XL         = X COOR DISTANCE FROM XC TO PATCH CENTROID
C     PHIL       =  PHI(ANGLE) DISTANCE FROM PHIC TO PATCH PHI CENTROID
C     THX        = PATCH DIMENSION ALONG X - AXIS
C     THPHI      = PATCH ARCLENGTH (RADIANS) ALONG PHI - AXIS
C     PLAT(M,N) = THE APPLIED PRESSURE ACROSS (THX)X(A*THPHI) OF PATCH (M,N)
C                  ALONG NORMAL TO CYLINDER AT XC,PHIC  + INWARD
C
C     THIS SUBROUTINE CREATES PR AND PPHI LOAD HARMONICS FOR SUCH LATERAL
C      LOADS IKP IS AN ERROR CHECK IF IKP .NE. 0 GO TO END OF PROGRAM
C
C     CREATED               :  27-JUL-83         BY: W. CRAFT
C     LATEST REVISION       :  30-SEP-83         BY: F.M.G. WONG
C
C     ******************************************************************
C
      REAL L,NU
      DIMENSION PLAT(4,4)
      COMMON / CONSTANTS / A, E, L, NU, PI, T
C
C
      IKP = 0
      IF ( (M .GT. 4) .OR. (N .GT. 4) ) IKP = 1
      IF ( (M .LT. 1) .OR. (N .LT. 1) ) IKP = 1
      IF ( IKP .EQ. 1 ) GO TO 98
      XM = FLOAT(M)
      XN = FLOAT(N)
C
      DO 1 I = 1, M
      XI = FLOAT(I)
      PHIEL = PHIC + PHIL + THPHI*(2.*XI - 1. - XM)/(2.*XM)
      PSI = PHIEL - PHIC
C
      DO 1 J = 1, N
      XJ = FLOAT(J)
      XEL = XC + XL + THX*(2.*XJ - 1. - XN)/(2.*XN)
C
C      ::::::: NOW PHI AND X COORDINATES OF ELEMENT M,N ARE KNOWN
C
      PR = PLAT(I,J)*(COS(PSI))**2
      PPHI = .5*PLAT(I,J)*SIN(2.*PSI)
      THXT = THX/(2.*XN)
      THPHIT = THPHI/(2.*XM)
C
      CALL PATCH (XEL, PHIEL, THXT, THPHIT, PR, 1)
```

```
      CALL PATCH (XEL, PHIEL, THXT, THPHIT, PPHI, 2)
C
1     CONTINUE
C
      GO TO 99
C
98    WRITE(6,2) M, N
2     FORMAT(1X,'ERROR DETECTED IN ELEMENTS - NO LOAD CREATES,M,N = ',2I3)
      WRITE(1,2)
C
C
99    RETURN
      END
C
```

```
C
C      ************************************************************************
C
       SUBROUTINE ELMCNT(M, N, XC, PHIC, XL, PHIL, THX, THPHI, IND, IKP)
C
C      COMPUTES THE ELEMENT CENTERS FOR ALL PATCHES AND STORES THEM
C      IN THE COMMON BLOCK "CENTELM"
C
C      IND = CODE FOR DESIGNATING THE PATCH AREA (1, 2, 3, OR 4)
C
C      CREATED                :  06-OCT-83          BY: F.M.G. WONG
C      LATEST REVISION        :  24-OCT-83          BY: F.M.G. WONG
C
C      ************************************************************************
C
       REAL L,NU
       COMMON / CONSTANTS / A, E, L, NU, PI, T
       COMMON / CENTELM / CNTXEL(4,4,4), CNTPHIEL(4,4,4)
C
C
       IKP = 0
       IF ( (M .GT. 4) .OR. (N .GT. 4) ) IKP = 1
       IF ( (M .LT. 1) .OR. (N .LT. 1) ) IKP = 1
       IF ( IKP .EQ. 1 ) GO TO 98
       XM = FLOAT(M)
       XN = FLOAT(N)
C
       DO 20 I = 1, N
C
               DO 10 J = 1, M
C
               XI = FLOAT(I)
               CNTPHIEL(I,J,IND) =PHIC+PHIL + THPHI*(2.*XI - 1. - XM)/(2.*XM)
C
               XJ = FLOAT(J)
               CNTXEL(I,J,IND) = XC + XL + THX*(2.*XJ - 1. - XN)/(2.*XN)
C
               PRINT *,'PHIC = ',PHIC,' PHIL = ',PHIL,' THPHI = ',THPHI
               PRINT *, 'CENTER: ',IND,CNTXEL(I,J,IND), CNTPHIEL(I,J,IND)
               PRINT *
C
10             CONTINUE
20      CONTINUE
C
       GO TO 99
C
98     WRITE(6,2) M, N
2      FORMAT(1X,'ERROR DETECTED IN ELEMENTS - NO LOAD CREATES,M,N = ',2I3)
       WRITE(1,2)
C
C
99     RETURN
       END
```

```
C     **********************************************************************
C
      SUBROUTINE DISPLACE (INDEX)
C
C     CALCULATES THE U, V, AND W DISPLACEMENTS
C
C     INDEX : 1 = W, 2 = V, 3 = U, 4 = W & V, 5 = W & U, 6 = W, V, & U
C
C     CREATED            :  18-JUL-83        BY:  F.M.G. WONG
C     LATEST REVISION    :  19-JAN-84
C
C     **********************************************************************
C
C
      REAL*4  L, NU
      COMMON / CONSTANTS / A, E, L, NU, PI, T
      COMMON / INDEXES / M, N
      COMMON / LOADS / PRE(100,100), PRO(100,100), PXE(100,100),
     1                  PXO(100,100), PPHIE(100,100), PPHIO(100,100)
      COMMON / DISPL / UE(100,100), UO(100,100), VE(100,100), VO(100,100),
     1                  WE(100,100), WO(100,100)
      DIMENSION EVEN(3), ODD(3), COEFF(3,3)
C
      DATA EVEN / 3*0 /, ODD / 3*0 /
C
C     :::::: SET UP VALUES OF THE COEFFICIENT MATRIX
C
      A2 = A*A
      A4 = A2*A2
      RL2 = L*L
      PI2 = PI*PI
      T2 = T*T
C
      CALL TIMESTAT (1)
C
      DO 20 I = 0, M-1
C
      RM = FLOAT(I)
      RM2 = RM*RM
C
      DO 10 J = 1, N
C
      RN = FLOAT(J)
      RN2 = RN*RN
C
      COEFF(1,1) = -( (RN2*PI2)/RL2 + (RM2*(1. - NU))/(2.*A2) )
      COEFF(1,2) = ((1. + NU)*RM*RN*PI)/(2.*A*L)
      COEFF(1,3) = -(NU*RN*PI)/(A*L)
      COEFF(2,1) = COEFF(1,2)
      COEFF(2,2) = -( ((1. - NU)*RN2*PI2)/(2.*RL2) + RM2/A2 +
     1              (T2*(1. - NU)*RN2*PI2)/(12.*RL2*A2) +
     1              (RM2*T2)/(12.*A4) )
      COEFF(2,3) = (T2*RM**3)/(12.*A4) + (RM*RN2*PI2*T2)/(12.*A2*RL2)+RM/A2
      COEFF(3,1) = COEFF(1,3)
      COEFF(3,2) = COEFF(2,3)
      COEFF(3,3) = -(1./A2 + (T2*RN2*RN2*PI2*PI2)/(12.*RL2*RL2) +
     1              (2.*T2*RM2*RN2*PI2)/(12.*A2*RL2) +
```

```
     1                (RM2*RM2*T2)/(12.*A4))
C
C      ::::::: SET UP VALUES FOR LOADING COLUMN MATRIX
C
       CONST = (1. - NU*NU)/(E*T)
C
       IF ((INDEX.EQ.1).OR.(INDEX.EQ.4).OR.(INDEX.EQ.5).OR.(INDEX.EQ.6)) THEN
               EVEN(3) = -PRE(I+1,J)*CONST
               ODD(3) = -PRO(I+1,J)*CONST
       ENDIF
       IF ( (INDEX .EQ. 2) .OR. (INDEX .EQ. 4) .OR. (INDEX .EQ. 6) ) THEN
               EVEN(2) = -PPHIE(I+1,J)*CONST
               ODD(2) = -PPHIO(I+1,J)*CONST
       ENDIF
       IF ( (INDEX .EQ. 3) .OR. (INDEX .EQ. 5) .OR. (INDEX .EQ. 6) ) THEN
               EVEN(1) = -PXE(I+1,J)*CONST
               ODD(1) = -PXO(I+1,J)*CONST
       ENDIF
C
C      ::::::: SOLVE FOR EVEN U, V, AND W DISPLACEMENTS
C
        CALL MATRIX (U, V, W, EVEN, COEFF)
C
       UE(I+1,J) = U
       VE(I+1,J) = V
       WE(I+1,J) = W
C
C      ::::::: SOLVE FOR ODD U, V, AND W DISPLACEMENTS
C
       COEFF(1,2) = -COEFF(1,2)
       COEFF(2,1) = -COEFF(2,1)
       COEFF(2,3) = -COEFF(2,3)
       COEFF(3,2) = -COEFF(3,2)
C
       CALL MATRIX (U, V, W, ODD, COEFF)
C
       UO(I+1,J) = U
       VO(I+1,J) = V
       WO(I+1,J) = W
C
C
10     CONTINUE
C
       IF ( I/2 .EQ. INT(I/2) ) CALL TIMESTAT (2)
20     CONTINUE
       CALL TIMESTAT (3)
C
C
       RETURN
       END
C
C
C      **********************************************************************
C
       SUBROUTINE MATRIX (U, V, W, RLOAD, COEFF)
C
```

```
C       SOLVES FOR THE U, V, AND W DISPLACMENTS FROM A 3 X 3 MATRIX
C       EQUATION
C
C       U       = U DISPLACEMENTS
C       V       = V DISPLACEMENTS
C       W       = W DISPLACEMENTS
C       RLOAD   = 1 X 3 COLUMN MATRIX OF EVEN OR ODD LOADING COEFFICIENTS
C       COEFF   = 3 X 3 COEFFICIENT MATRIX
C
C       CREATED:  18-JUL-83        BY:  F.M.G. WONG
C
C       ***********************************************************************
C
C
        DIMENSION RLOAD(3), SLOAD(3), DOEFF(3,3), COEFF(3,3)
C
C
C
C       ::::::: MAKE FIRST COLUMN IN MATRIX ALL 1'S
C
         DOEFF(1,2) = COEFF(1,2)/COEFF(1,1)
         DOEFF(1,3) = COEFF(1,3)/COEFF(1,1)
         SLOAD(1) = RLOAD(1)/COEFF(1,1)
        DOEFF(1,1) = 1.0
C
        IF ( COEFF(2,1) .EQ. 0.00 ) GOTO 15
        DOEFF(2,2) = COEFF(2,2)/COEFF(2,1)
        DOEFF(2,3) = COEFF(2,3)/COEFF(2,1)
        SLOAD(2) = RLOAD(2)/COEFF(2,1)
        DOEFF(2,1) = 1.0
C
15      DOEFF(3,2) = COEFF(3,2)/COEFF(3,1)
        DOEFF(3,3) = COEFF(3,3)/COEFF(3,1)
        SLOAD(3) = RLOAD(3)/COEFF(3,1)
        DOEFF(3,1) = 1.0
C
C       ::::::: MAKE FIRST ELEMENT IN 2ND AND 3RD ROWS ZERO
C
        IF ( COEFF(2,1) .EQ. 0.00 ) GOTO 25
        DOEFF(2,1) = DOEFF(1,1) - DOEFF(2,1)
        DOEFF(2,2) = DOEFF(1,2) - DOEFF(2,2)
        DOEFF(2,3) = DOEFF(1,3) - DOEFF(2,3)
        SLOAD(2) = SLOAD(1) - SLOAD(2)
C
25      DOEFF(3,1) = DOEFF(1,1) - DOEFF(3,1)
        DOEFF(3,2) = DOEFF(1,2) - DOEFF(3,2)
        DOEFF(3,3) = DOEFF(1,3) - DOEFF(3,3)
        SLOAD(3) = SLOAD(1) - SLOAD(3)
C
C       ::::::: MAKE DOEFF(3,2) ZERO TO COMPLETE UPPER TRIANGULAR MATRIX
C
        IF ( DOEFF(3,2) .EQ. 0.00 ) GOTO 30
        DOEFF(2,3) = DOEFF(2,3)/DOEFF(2,2)
        SLOAD(2) = SLOAD(2)/DOEFF(2,2)
        DOEFF(2,2) = 1.0
C
```

```
      DOEFF(3,3) = DOEFF(3,3)/DOEFF(3,2)
      SLOAD(3) = SLOAD(3)/DOEFF(3,2)
      DOEFF(3,2) = 1.0
C
      DOEFF(3,2) = DOEFF(3,2) - DOEFF(2,2)
      DOEFF(3,3) = DOEFF(3,3) - DOEFF(2,3)
      SLOAD(3) = SLOAD(3) - SLOAD(2)
C
C
C     ::::::: SOLVE FOR U, V, AND W: DISPLACEMENTS
C
30    W = SLOAD(3)/DOEFF(3,3)
      V = SLOAD(2) - DOEFF(2,3)*W
      U = SLOAD(1) - DOEFF(1,2)*V - DOEFF(1,3)*W
C
C     PRINT *, ' U =',U,' V =',V,' W =',W
C
      RETURN
      END
C
C
C
C     **********************************************************************
C
      SUBROUTINE DISPMAP
C
C     WRITES THE DISPLACEMENT MAP
C
C     CREATED:  20-JUL-83       BY: F.M.G. WONG
C
C     **********************************************************************
C
      REAL L, NU
      DIMENSION XMAT(11),PHIMAT(11),RINT(11,11)
      COMMON / CONSTANTS / A, E, L, NU, PI, T
      COMMON / INDEXES / M, N
      COMMON / DISPL / UE(100,100), UO(100,100), VE(100,100), VO(100,100),
     1                 WE(100,100), WO(100,100)
C
C
11    WRITE(6,25)
      WRITE(1,25)
25    FORMAT(/,1X,'DO YOU WANT A NEW DISPLACEMENT MAP? 1=YES, 2=NO, 3=END')
      READ(4,26) MAP
26    FORMAT (1X,I1)
      WRITE(1,46) MAP
46    FORMAT(1X,'MAP=',I1,/)
      IF (MAP.NE.1) GOTO 100
      WRITE (6,13)
      WRITE (1,13)
13    FORMAT (1X,'INPUT DESIRED DISPLACEMENT MAP: W = 1, V = 2, U = 3')
      READ (4,26) INDEX
      WRITE (1,45) INDEX
45    FORMAT (1X,'INDEX =',I1,/)
      WRITE(6,14)
      WRITE(1,14)
```

```
14      FORMAT(1X,'INPUT XST, DXST, PHIST, DPHIST (ANGLES/DEG)',/)
        READ(4,27) XST, DXST, PHIST, DPHIST
27      FORMAT (1X,4F11.4)
        WRITE(1,47) XST, DXST, PHIST, DPHIST
47      FORMAT(1X,'XST, DXST, PHIST, DPHIST =',4F11.4,/)
C
        PHIST=PI*PHIST/180.
        DPHIST=PI*DPHIST/180.
C
        CALL TIMESTAT (1)
C
        DO 10 I=1,11
                XCO=XST+DXST*FLOAT(I-6)/10.
                XMAT(I)=XCO
                DO 12 J=1,11
                        PHICO=PHIST+DPHIST*FLOAT(J-6)/10.
                        IF (I.EQ.1) PHIMAT(J)=PHICO*180./PI
C
C       ::::::: NOW COMPUTE THE DISPLACEMENTS ON THE SURFACE MAP AND PRESENT
C        ::::::: THE RESULTS AS 0 TO 100% OF F AT EACH GRID POINT AND PLOT.
C
                        XINT=0
                        DO 6 MT=0,M-1
                        SPHI=SIN(FLOAT(MT)*PHICO)
                        CPHI=COS(FLOAT(MT)*PHICO)
                        DO 6 NT=1,N
                        IF ( INDEX .EQ. 1 ) XINC=(WE(MT+1,NT)*CPHI+WO(MT+1,NT)
     1                  *SPHI)*SIN(FLOAT(NT)*PI*XCO/L)
                        IF ( INDEX .EQ. 2 ) XINC=(VE(MT+1,NT)*SPHI+VO(MT+1,NT)
     1                  *CPHI)*SIN(FLOAT(NT)*PI*XCO/L)
                        IF ( INDEX .EQ. 3 ) XINC=(UE(MT+1,NT)*CPHI+UO(MT+1,NT)
     1                  *SPHI)*COS(FLOAT(NT)*PI*XCO/L)
                        XINT=XINT+XINC
6                       CONTINUE
                XINT=XINT*.5
                RINT(I,J)=XINT
12              CONTINUE
        CALL TIMESTAT (2)
10      CONTINUE
        CALL TIMESTAT (3)
C
        CALL FINDMIN (RINT, RMIN, RMAX)
C
        CALL DIVTEN (RINT, RMIN, RMAX, EXPT)
C
C        ::::::: NOW THE ARRAY IS SETUP THAT HAS PERCENTAGES OF DISPLACEMENTS
C
        IF ( EXPT .NE. 0.0 ) THEN
                WRITE (6,18) EXPT
                WRITE (1,18) EXPT
18              FORMAT (15X,'---> VALUES IN MAP SCALED AS ',E9.2,/)
        ENDIF
        IF ( INDEX .EQ. 1 ) THEN
                WRITE (6,1801)
                WRITE (1,1801)
1801            FORMAT (26X,'DISPLACEMENT MAP OF W COMPONENT',/)
```

```
      ENDIF
      IF ( INDEX .EQ. 2 ) THEN
              WRITE (6,1802)
              WRITE (1,1802)
1802          FORMAT (26X,'DISPLACEMENT MAP OF V COMPONENT',/)
      ENDIF
      IF ( INDEX .EQ. 3 ) THEN
              WRITE (6,1803)
              WRITE (1,1803)
1803          FORMAT (26X,'DISPLACEMENT MAP OF U COMPONENT',/)
      ENDIF
C
      WRITE(6,17) (PHIMAT(K),K=1,11)
      WRITE(1,17) (PHIMAT(K),K=1,11)
17    FORMAT(1X,'OUTPUT OF DISPLACEMENTS ON GRID IS:',/,8X,11(F6.1,2X),/)
C
      DO 9 NN=1,11
              WRITE(6,19) XMAT(12-NN),(RINT(12-NN,K),K=1,11)
              WRITE(1,19) XMAT(12-NN),(RINT(12-NN,K),K=1,11)
19            FORMAT(1X,F6.2,11F8.4)
9     CONTINUE
      GO TO 11
C
100   RETURN
      END
C
C
C     ***********************************************************************
C
      SUBROUTINE FINDMIN (DISP, RMIN, RMAX)
C
C     FINDS THE MININUM  AND MAXIMUM VALUE IN THE ARRAY 'DISP' AND
C      RETURNS MINIMUM VALUE IN 'RMIN' AND MAXIMUM VALUE IN 'RMAX.'
C
C     CREATED:  21-JUL-83        BY: F.M.G. WONG
C
C     ***********************************************************************
C
      DIMENSION DISP(11,11)
C
      RMIN = DISP(1,1)
      RMAX = DISP(1,1)
      DO 20 I = 1, 11
              DO 10 J = 1,11
                      RMIN = MIN(ABS(RMIN), ABS(DISP(I,J)))
                      RMAX = MAX(ABS(RMAX), ABS(DISP(I,J)))
10            CONTINUE
20    CONTINUE
C
      RETURN
      END
C
C
C     ***********************************************************************
C
      SUBROUTINE DIVTEN (DISP, RMIN, RMAX, EXPT)
```

```
C
C       SCALES THE VALUES OF THE 'DISP' ARRAY ACCORDING TO THE LOWEST
C       POWER OF TEN
C
C       CREATED       :  21-JUL-83        BY: F.M.G. WONG
C       LAST REVISED  :  06-SEP-83
C
C       ************************************************************************
C
        DIMENSION DISP(11,11)
C
        EXPT = 0.0
        IF ( (ABS(RMIN) .GT. 0.001) .AND. (ABS(RMAX) .LT. 10.0) ) RETURN
        IF ( (ABS(RMIN) .LT. 0.001) .AND. (ABS(RMAX) .GT. 10.0) ) RMIN = 1.0
        IF ( (ABS(RMIN) .LT. 0.001) .AND. (ABS(RMAX) .GT. 1.0) ) RMIN = 1.0
C
        IF ( RMIN .LE. 0.0 ) RMIN = 1.0
        IF ( RMAX .LE. 0.0 ) RMAX = 1.0
        EXMIN = LOG10(RMIN)
        EXMAX = LOG10(RMAX)
        EXBIG = MAX( ABS(EXMIN),ABS(EXMAX) )
        IF ( EXMIN .LT. 0.0 .AND. EXMAX .LT. 0.0 )  EXBIG = ABS(EXMAX)
        IF ( (EXMAX - EXMIN) .GT. 4.0 ) EXBIG = ABS(EXMAX)
C
        TEN = 10.00
        IF ( EXBIG .EQ. ABS(EXMIN) ) EXPT = TEN**NINT(EXMIN)
        IF ( EXBIG .EQ. ABS(EXMAX) ) EXPT = TEN**NINT(EXMAX)
C
C       PRINT *, 'MAX =',RMAX,' MIN =',RMIN,' EXPT =',EXPT
        DO 40 I = 1, 11
                DO 30 J = 1, 11
                        DISP(I,J) = DISP(I,J)/EXPT
                        IF ( ABS(DISP(I,J)) .LT. 0.0001 )  DISP(I,J) = 0.0
C                       PRINT *, DISP(I,J)
30              CONTINUE
40      CONTINUE
C
        RETURN
        END
C
C
C
        SUBROUTINE MATPRT (COEFF, RLOAD)
C
        DIMENSION COEFF(3,3), RLOAD(3)
C
        PRINT *
        PRINT *, COEFF(1,1), COEFF(1,2), COEFF(1,3), ':', RLOAD(1)
        PRINT *, COEFF(2,1), COEFF(2,2), COEFF(2,3), ':', RLOAD(2)
        PRINT *, COEFF(3,1), COEFF(3,2), COEFF(3,3), ':', RLOAD(3)
        PRINT *
C
        RETURN
        END
```

```
C
C     ********************************************************************
C
      SUBROUTINE MOMENTSET(RMOM, XC, PHIC, TAUX, TAUPHI, LX, LPHI, M, N, IKP,
     1                     MOMTYP)
C
C     SETS UP THE CALLS TO "MOMENTLAT" WHICH DECOMPOSES THE SPECIFIED
C     MOMENT INTO DISCRETE RADIAL FORCES
C
C     XC     = X-COORDINATE OF TUBE CENTER (INCHES)
C     PHIC   = PHI-COORDINATE OF TUBE CENTER (RADIANS)
C     TAUX   = THICKNESS OF ATTACHMENT TUBE ALONG X (INCHES)
C     TAUPHI = THICKNESS OF ATTACHMENT TUBE ALONG PHI (INCHES)
C     LPHI   = OUTER HORIZONTAL DIMENSION (INCHES) OF ATTACHMENT TUBE
C     LX     = OUTER VERTICAL DIMENSION (INCHES) OF ATTACHMENT TUBE
C      RMOM   = EXTERNALLY APPLIED MOMENT
C     MOMTYP = DIRECTION OF MOMENT
C
C     M,N ARE ELEMENT MEMBERS OF PATCH IN PHI AND X DIRECTIONS, RESP.
C
C     CREATED              :  02-SEP-83      BY: F.M.G. WONG
C     LATEST REVISION      :  29-SEP-83
C
C     ********************************************************************
C
      REAL L,NU,LPHI,LX
      COMMON / CONSTANTS / A, E, L, NU, PI, T
C
C
      ALPO = ASIN(LPHI/(2.*A))
      ALPI = ASIN((LPHI - 2.*TAUPHI)/(2.*A))
      V1 = .5*(ALPO + ALPI)
      V2 = LX-2.*TAUX
      V3 = .5*(LX - TAUX)
C
      ANG = ALPO - ALPI
      V4 = 2.*ALPO
      IF ( MOMTYP .EQ.  1 )  CALL MOMINERT (E, LX, LPHI, TAUX, TAUPHI)
      IF ( MOMTYP .EQ.  2 )  CALL MOMINERT (E, LPHI, LX, TAUPHI, TAUX)
      WRITE (1,10) E
10    FORMAT (1X,'MOMENT OF INERTIA: ',F10.5)
C
      CALL MOMENTLAT (M, N, XC, PHIC, 0., V1, V2, ANG, IKP, RMOM, MOMTYP, E)
      CALL MOMENTLAT (M, N, XC, PHIC, V3, 0., TAUX, V4, IKP, RMOM, MOMTYP, E)
      CALL MOMENTLAT (M, N, XC, PHIC, 0., -V1, V2, ANG, IKP, RMOM, MOMTYP, E)
      CALL MOMENTLAT (M, N, XC, PHIC, -V3, 0., TAUX, V4, IKP, RMOM, MOMTYP,E)
C
      RETURN
      END
C
C
C
C     ********************************************************************
C
      SUBROUTINE MOMINERT (ERTIA, LX, LPHI, TAUX, TAUPHI)
C
C     CALCULATES THE MOMENT OF INERTIA OF A RECTANGULAR ATTACHMENT.
```

```
C
C       ERTIA    =  MOMENT OF INERTIA RETURNED
C       LX       =  PATCH AXIAL LENGTH
C       LPHI     =  PATCH CIRCUMFERENTIAL LENGTH
C       TAUX     =  PATCH AXIAL THICKNESS
C       TAUPHI   =  PATCH CIRCUMFERENTIAL THICKNESS
C
C       CREATED          :  09-SEP-83                BY  :  F.M.G. WONG
C       LATEST REVISION  :  09-SEP-83
C
C       **********************************************************************
C
C
        REAL LX, LPHI
C
C
        ENTIRE = (LPHI*LX**3)/12.
        HOLE = ((LPHI - 2.*TAUPHI)*(LX - 2.*TAUX)**3)/12.
C
        ERTIA = ENTIRE - HOLE
C
C
        RETURN
        END
```

```
C
C     ********************************************************************
C
      SUBROUTINE MOMENTLAT (M, N, XC, PHIC, XL, PHIL, THX, THPHI, IKP,
     1                        RMOM, MOMTYP, ERTIA)
C
C     MOMENTLAT ALLOWS PARTITIONING OF LOADS IN LOCAL COORDINATES TO MATCH
C     LATERAL COMPONENTS
C
C     M         =   NO. OF PHI DIRECTION ELEMENTS IN PATCH
C     N         =   NO. OF X DIRECTION ELEMENTS IN PATCH
C     XC        =   LOCAL PATCH REFERENCE X COORDINATE
C     PHIC      =   LOCAL PATCH REFERENCE PHI COORDINATE
C     (XC,PHIC) DEFINE DIRECTION OF LATERAL LOAD AXIS ON CYLINDER
C     XL        =   X COOR DISTANCE FROM XC TO PATCH CENTROID
C     PHIL      =   PHI(ANGLE) DISTANCE FROM PHIC TO PATCH PHI CENTROID
C     THX       =   PATCH DIMENSION ALONG X - AXIS
C     THPHI     =   PATCH ARCLENGTH (RADIANS) ALONG PHI - AXIS
C     RMOM      =   EXTERNALLY APPLIED MOMENT (IN.-LBS.)
C     MOMTYP    =   DIRECTION OF MOMENT
C     ERTIA     =   MOMENT OF INERTIA
C
C     THIS SUBROUTINE CREATES PR AND PPHI LOAD HARMONICS FOR SUCH LATERAL
C      LOADS.
C      IKP IS AN ERROR CHECK IF IKP .NE. 0  GO TO END OF PROGRAM
C
C     CREATED              :  01-SEP-83          BY: F.M.G. WONG
C     LATEST REVISION      :  06-DEC-83
C
C     ********************************************************************
C
      REAL L,NU
      DIMENSION PLAT(4,4), PHINTL(4,4), XNTL(4,4), XEL(4,4), PHIEL(4,4),
     1          PSI(4,4)
      COMMON / CONSTANTS / A, E, L, NU, PI, T
C
C
      IKP = 0
      IF ( (M .GT. 4) .OR. (N .GT. 4) ) IKP = 1
      IF ( (M .LT. 1) .OR. (N .LT. 1) ) IKP = 1
      IF ( IKP .EQ. 1 ) GO TO 98
      XM = FLOAT(M)
      XN = FLOAT(N)
C
      DO 20 I = 1, N
C
              DO 10 J = 1, M
                      XJ = FLOAT(I)
                      XI = FLOAT(J)
C
                      PHIEL(I,J) = PHIC+PHIL+ THPHI*(2.*XI - 1. - XM)/(2.*XM)
                      PSI(I,J) = PHIEL(I,J) - PHIC
C
                      XEL(I,J) = XC + XL - THX*(2.*XJ - 1. - XN)/(2.*XN)
C
                      XNTL(I,J) = XEL(I,J) - XC
                      PHINTL(I,J) = A*SIN( PHIEL(I,J) - PHIC )
```

```
C
10              CONTINUE
20        CONTINUE
C
          DO 50 I = 1, N
                  DO 40 J = 1, M
C
                          IF ( MOMTYP .EQ. 1 ) ELD = XNTL(I,J)
                          IF ( MOMTYP .EQ. 2 ) ELD = PHINTL(I,J)
C
                          PLAT(I,J) = RMOM*ELD/ERTIA
C
C          :::::: NOW PHI AND X COORDINATES OF ELEMENT M,N ARE KNOWN
C
                          PR = PLAT(I,J)*(COS(PSI(I,J)))**2
                          PPHI = .5*PLAT(I,J)*SIN(2.*PSI(I,J))
                          THXT = THX/(2.*XN)
                          THPHIT = THPHI/(2.*XM)
C
                          CALL PATCH (XEL(I,J), PHIEL(I,J), THXT, THPHIT, PR, 1)
                          CALL PATCH (XEL(I,J), PHIEL(I,J), THXT, THPHIT, PPHI,2)
C
40              CONTINUE
50        CONTINUE
C
          GO TO 99
C
98        WRITE(6,2) M, N
2         FORMAT(1X,'ERROR DETECTED IN ELEMENTS - NO LOAD CREATES,M,N = ',2I3)
          WRITE(1,2)
C
C
99        RETURN
          END
```

```
C
C      ******************************************************************
C
       SUBROUTINE STRESS
C
C      CALCULATES THE BENDING AND MEMBRANE STRESS
C
C      CREATED          :  27-JUL-83          BY: F.M.G. WONG
C      LATEST REVISION  :  25-AUG-83
C
C       ******************************************************************
C
       CHARACTER*5 TYPE(12)
       REAL L, NU
       DIMENSION RMAP(11,11), XMAT(11), PHIMAT(11)
       COMMON / CONSTANTS / A, E, L, NU, PI, T
       COMMON / INDEXES / M, N
       COMMON / DISPL / UE(100,100), UO(100,100), VE(100,100), VO(100,100),
     1                  WE(100,100), WO(100,100)
C
C
       DATA TYPE/ 'NX   ','NPHI ','NXP  ','MX   ','MPHI ','MXP  ',
     1            'SIGX-','SIGX+','SIGP-','SIGP+','TAU -','TAU +' /
11     WRITE(6,25)
       WRITE(1,25)
25     FORMAT(/,1X,'DO YOU WANT A NEW STRESS MAP? 1=YES, 2=NO, 3=END')
       READ(4,26) MAP
26     FORMAT (1X,I2)
       WRITE(1,46) MAP
46     FORMAT(1X,'MAP=',I1,/)
       IF (MAP.NE.1) GOTO 100
       WRITE (6,13)
       WRITE (1,13)
13     FORMAT(1X,'INPUT DESIRED STRESS MAP: NX = 1,    NP = 2,   NXP = 3,',
     1 /,1X,'                          MX = 4,    MP = 5,   MXP = 6,',/,
     1 1X,'                       SIGX- = 7, SIGX+ = 8, SIGP- = 9,',/,
     1 1X,'                       SIGP+ = 10, TAU- = 11, TAU+ = 12')
       READ (4,26) INDEX
       WRITE (1,45) INDEX
45     FORMAT (1X,'INDEX =',I2,/)
       WRITE(6,14)
       WRITE(1,14)
14     FORMAT(1X,'INPUT XST, DXST, PHIST, DPHIST (ANGLES/DEG)',/)
       READ(4,48) XST, DXST, PHIST, DPHIST
48     FORMAT (1X,4F11.4)
       WRITE(1,47) XST, DXST, PHIST, DPHIST
47     FORMAT(1X,'XST, DXST, PHIST, DPHIST =',4F11.4,/)
C
       PHIST=PI*PHIST/180.
       DPHIST=PI*DPHIST/180.
C
C
       A2 = A*A
       T2 = T*T
C
       CALL TIMESTAT (1)
       DO 10 I = 1, 11
```

```
          XCO = XST + DXST*FLOAT(I-6)/10.
          XMAT(I) = XCO
          DO 12 J = 1, 11
                  PHICO = PHIST + DPHIST*FLOAT(J-6)/10.
                  IF ( I .EQ. 1 ) PHIMAT(J) = PHICO*180./PI
C
C       ::::::: COMPUTE DERIVATIVES OF DISPLACEMENTS
C
C
                  CALL DERIVDSP (XCO, PHICO, W, DWXXS, DWPPS, DWXPS,
     1                           DVXS, DVPS, DUXS, DUPS)
C
                  RNX  = E*T/(1. - NU*NU)*(DUXS + NU*(DVPS - W)/A)
                  RNP  = E*T/(1. - NU*NU)*(DVPS/A - W/A +NU*DUXS)
                  RNXP = E*T/(2.*(1. + NU))*(DVXS +DUPS/A)
C
                  D = E*T**3/(12*(1. - NU*NU))
C
                  RMX  = -D*(DWXXS + NU*(DVPS + DWPPS)/A2)
                  RMP  = -D*((DVPS + DWPPS)/A2 + NU*DWXXS)
                  RMXP = -D*(1. - NU)/A*(DVXS + DWXPS)
C
C       ::::::: COMPUTE X AND PHI STRESS, AND SHEAR
C
C
                  SIGXN = RNX/T - 6.*RMX/T2
                  SIGXP = RNX/T + 6.*RMX/T2
C
                  SIGPN = RNP/T - 6.*RMP/T2
                  SIGPP = RNP/T + 6.*RMP/T2
C
                  TAXPN = RNXP/T - 6.*RMXP/T2
                  TAXPP = RNXP/T + 6.*RMXP/T2
C
                  IF ( INDEX .EQ. 1 )  RMAP(I,J) = RNX
                  IF ( INDEX .EQ. 2 )  RMAP(I,J) = RNP
                  IF ( INDEX .EQ. 3 )  RMAP(I,J) = RNXP
                  IF ( INDEX .EQ. 4 )  RMAP(I,J) = RMX
                  IF ( INDEX .EQ. 5 )  RMAP(I,J) = RMP
                  IF ( INDEX .EQ. 6 )  RMAP(I,J) = RMXP
                  IF ( INDEX .EQ. 7 )  RMAP(I,J) = SIGXN
                  IF ( INDEX .EQ. 8 )  RMAP(I,J) = SIGXP
                  IF ( INDEX .EQ. 9 )  RMAP(I,J) = SIGPN
                  IF ( INDEX .EQ. 10 ) RMAP(I,J) = SIGPP
                  IF ( INDEX .EQ. 11 ) RMAP(I,J) = TAXPN
                  IF ( INDEX .EQ. 12 ) RMAP(I,J) = TAXPP
C
12        CONTINUE
10    CONTINUE
      CALL TIMESTAT (3)
C
      CALL FINDMIN (RMAP, RMIN, RMAX)
C
      CALL DIVTEN (RMAP, RMIN, RMAX, EXPT)
C
      IF ( EXPT .NE. 0.0 ) THEN
```

```
                WRITE (6,18) EXPT
                WRITE (1,18) EXPT
18              FORMAT (1X,'VALUES IN MAP SCALED AS ',E9.2,/)
        ENDIF
C
        WRITE (6,15) TYPE(INDEX)
        WRITE (1,15) TYPE(INDEX)
15      FORMAT (26X,'M, N, OR STRESS MAP OF ',A5,/)
C
        WRITE(6,17) (PHIMAT(K),K=1,11)
        WRITE(1,17) (PHIMAT(K),K=1,11)
17      FORMAT(1X,'OUTPUT OF M, N, OR STRESS ON GRID IS:',/,8X,11(F6.1,2X),/)
C
        DO 9 NN=1,11
                WRITE(6,19) XMAT(12-NN),(RMAP(12-NN,K),K=1,11)
                WRITE(1,19) XMAT(12-NN),(RMAP(12-NN,K),K=1,11)
19              FORMAT(1X,F6.2,11F8.4)
9       CONTINUE
        GO TO 11
C
100     RETURN
        END
C
C
C
C
C       ********************************************************************
C
        SUBROUTINE DERIVDSP (XCO, PHICO, WS, DWXXS, DWPPS, DWXPS,
     1                          DVXS, DVPS, DUXS, DUPS)
C
C       CALCULATES THE DERIVATIVES OF THE DISPLACEMENTS U, V, AND W
C
C       W     = DISPLACEMENT W
C       DWXXS = SECOND DERIV. OF W WITH RESP. TO X
C       DWPPS = SECOND DERIV. OF W WITH RESP. TO PHI
C       DWXPS = MIXED PARTIAL OF W WITH RESP. TO X THEN PHI
C       DVXS  = FIRST  DERIV. OF V WITH RESP. TO X
C       DVPS  = FIRST  DERIV. OF V WITH RESP. TO PHI
C       DUXS  = FIRST  DERIV. OF U WITH RESP. TO X
C       DUPS  = FIRST  DERIV. OF U WITH RESP. TO PHI
C
C       CREATED:  27-JUL-83       BY: F.M.G. WONG
C
C       ********************************************************************
C
        REAL L, NU
        COMMON / CONSTANTS / A, E, L, NU, PI, T
        COMMON / INDEXES / M, N
        COMMON / DISPL / UE(100,100), UO(100,100), VE(100,100), VO(100,100),
     1                  WE(100,100), WO(100,100)
C
C       :::::: NOW COMPUTE THE  DERIVATIVES OF THE DISPLACEMENTS
C
        WS    = 0.
        DWXXS = 0.
```

```
      DWPPS = 0.
      DWXPS = 0.
      DVXS  = 0.
      DVPS  = 0.
      DUXS  = 0.
      DUPS  = 0.
C
      DO 6 MT = 0, M-1
C
              RMT = FLOAT(MT)
              SPHI = SIN(RMT*PHICO)
              CPHI = COS(RMT*PHICO)
C
                      DO 6 NT = 1, N
C
                      RNT = FLOAT(NT)
C
                      W    = .5*(WE(MT+1,NT)*CPHI + WO(MT+1,NT)*SPHI)
     1                          *SIN(RNT*PI*XCO/L)
                      DWXX = -.5*(RNT*PI/L)*(RNT*PI/L)*(WE(MT+1,NT)*CPHI +
     1                          WO(MT+1,NT)*SPHI)*SIN(RNT*PI*XCO/L)
                      DWPP = -.5*RMT*RMT*(WE(MT+1,NT)*CPHI + WO(MT+1,NT)
     1                          *SPHI)*SIN(RNT*PI*XCO/L)
                      DWXP = .5*RMT*(RNT*PI/L)*(-WE(MT+1,NT)*SPHI +
     1                          WO(MT+1,NT)*CPHI)*COS(RNT*PI*XCO/L)
C
                      DVX = .5*(RNT*PI/L)*(VE(MT+1,NT)*SPHI + VO(MT+1,NT)
     1                         *CPHI)*COS(RNT*PI*XCO/L)
                      DVP = .5*RMT*(VE(MT+1,NT)*CPHI - VO(MT+1,NT)
     1                         *SPHI)*SIN(RNT*PI*XCO/L)
C
                       DUX = -.5*(RNT*PI/L)*(UE(MT+1,NT)*CPHI +
     1                         UO(MT+1,NT)*SPHI)*SIN(RNT*PI*XCO/L)
                       DUP = .5*RMT*(-UE(MT+1,NT)*SPHI + UO(MT+1,NT)
     1                         *CPHI)*SIN(RNT*PI*XCO/L)
C
                         WS = WS    + W
                      DWXXS = DWXXS + DWXX
                      DWPPS = DWPPS + DWPP
                      DWXPS = DWXPS + DWXP
                      DVXS  = DVXS  + DVX
                      DVPS  = DVPS  + DVP
                      DUXS  = DUXS  + DUX
                      DUPS  = DUPS  + DUP
C
6             CONTINUE
      CALL TIMESTAT (2)
C
C
      RETURN
      END
C
C
```

```
C       ********************************************************************
C
        SUBROUTINE INTPRES (INDEX, RHOG, XO)
C
C       ADDS INTERNAL PRESSURE TO THE RADIAL COMPONENT OF THE LOAD
C
C       INDEX = 1: UNIFORM INTERNAL RADIAL PRESSURE; 2: HYDROSTATIC PRESSURE
C       RHOG = DENSITY OF FLUID (LBS./IN**3)
C       XO = FLUID LEVEL IN TANK (IN.)
C       QPRES = INTERNAL PRESSURE (PSI)
C
C       CREATED          :   10-AUG-83           BY:  F.M.G. WONG
C       LATEST REVISION  :   14-DEC-83
C
C       ********************************************************************
C
        REAL L, NU
        COMMON / CONSTANTS / A, E, L, NU, PI, T
        COMMON / INDEXES / M, N
        COMMON / INTPRS / PRESS(100), QPRES
C
C
        IF ( INDEX .EQ. 1 ) THEN
                DO 40 NT = 1, N, 2
                        XNT = FLOAT(NT)
                        PRESS(NT) = 4.*QPRES/(PI*XNT)
40              CONTINUE
        ENDIF
C
        IF ( INDEX .EQ. 2 ) THEN
                DO 50 NT = 1, N
                        XNT = FLOAT(NT)
                        C = (XNT*PI)/L
                        ONE = (XO/C)*(1. - COS(C*XO))
                        F = 1./(C**2)
                        G = XO/C
                        TWO = F*SIN(C*XO) - G*COS(C*XO)
                        VALPRS = 2.*RHOG*(ONE - TWO)/L
                        PRESS(NT) = VALPRS
50              CONTINUE
                QPRES = 1.0
        ENDIF
C
C
C
        RETURN
        END
C
C
C
C       ********************************************************************
C
        SUBROUTINE MERGPRES
C
C       INTEGRATES THE INTERNAL PRESSURE COMPONENT INTO THE GENERAL
C       RADIAL LOAD.
C
```

```
C      CREATED:  22-AUG-83         BY:  F.M.G. WONG
C
C      ********************************************************************
C
C
       COMMON / INDEXES / M, N
       COMMON / LOADS / PRE(100,100), PRO(100,100), PXE(100,100),
     1                  PXO(100,100), PPHIE(100,100), PPHIO(100,100)
       COMMON / INTPRS  / PRESS(100), QPRES
C
       DO 10 NT = 1, N
               PRE(1,NT) = PRE(1,NT) - 2.*PRESS(NT)
10     CONTINUE
C
C      RETURN
       END
```

```
C      ********************************************************************
C
       SUBROUTINE MULTNOZ
C
C      DRIVER MODULE FOR MULTIPLE NOZZLES/ATTACHMENTS CAPABILITY
C
C      CREATED      :  12-NOV-83          BY :  F.M.G. WONG
C      LAST REVISED :  12-NOV-83
C
C       ********************************************************************
C
       REAL LPHI, LX
C
       WRITE (6,5)
        WRITE (1,5)
5      FORMAT (/,1X,'ENTER TOTAL NUMBER OF NOZZLES: ')
       READ (4,7) NOZ
7      FORMAT (1X,I2)
       WRITE (6,8) NOZ
       WRITE (1,8) NOZ
8      FORMAT (1X,'TOTAL NUMBER OF NOZZLES = ',I2)
C
       DO 90 INOZ = 1, NOZ
C
       WRITE (6,10) INOZ
        WRITE (1,10) INOZ
10     FORMAT(/,1X,'    ---> ENTER DATA FOR NOZZLE',I2)
C
C      :::::::
C      :::::::  INITIAL ATTACHMENT DIMENSIONS WITH RADIAL FORCE
C      :::::::
C
       WRITE(6,11)
       WRITE(1,11)
11     FORMAT(/,1X,'INPUT TUBE CENTER (XC,PHIC), WALL THICKNESSES,',/,
     1  1X,'TAUX, TAUPHI, THEN TUBE SECTIONS, LX, LPHI',/,
     2  1X,'THEN NO. OF QUARTER-TUBE ELEMENTS, MTUB, NTUB, AND LATERAL',
     3  1X,'FORCE, F, IN POUNDS',/)
C
       WRITE(6,15)
       WRITE(1,15)
15     FORMAT(1X,'INPUT XC, PHIC, TAUX, TAUPHI, LX, LPHI, MTUB, NTUB, F',/,
     1  1X,'PHIC IN DEGREES.',/)
       READ(4,17) XC, PHIC, TAUX, TAUPHI, LX, LPHI, MTUB, NTUB, F
17     FORMAT (1X,6E11.4,2I3,E11.4)
       WRITE(1,19) XC, PHIC, TAUX, TAUPHI, LX, LPHI, MTUB, NTUB, F
19     FORMAT(1X,'XC, PHIC, TAUX, TAUPHI, LX, LPHI, MTUB, NTUB, F =',/,
     1  1X,6E11.4,2I3,E11.4,/)
C
       PHIC=PI*PHIC/180.
C
C      :::::::
C      :::::::   EXTERNAL MOMENT DATA
C      :::::::
C
       WRITE (6,20)
       WRITE (1,20)
```

```
20      FORMAT (1X,'EXTERNALLY APPLIED MOMENTS: YES = 1, NO = 0')
        READ (4,201) MOMRPY
201     FORMAT (1X,I1)
        WRITE (6,21) MOMRPY
        WRITE (1,21) MOMRPY
21      FORMAT (1X,'RESPONSE TO MOMENT PROMPT = ',I,/)
        IF ( MOMRPY .EQ. 0 ) GO TO 30
        WRITE (6,22)
        WRITE (1,22)
22      FORMAT (1X,'TYPE OF MOMENT: LONGITUDINAL = 1, CIRCUMFERENTIAL = 2')
        READ (4,201) MOMTYP
        WRITE (1,23) MOMTYP
23      FORMAT (1X,'TYPE OF MOMENT = ',I,/)
        WRITE (6,24)
        WRITE (1,24)
24      FORMAT (1X,'ENTER EXTERNALLY APPLIED MOMENT (IN.-LBS.):')
        READ (4,25) RMOM
25      FORMAT (1X,F9.1)
        WRITE (1,26) RMOM
26      FORMAT (1X,'EXTERNALLY APPLIED MOMENT (IN.-LBS.) = ',F9.1,/)
C
C       :::::::
C       :::::::  ASSEMBLE LOADING COEFFICIENTS --> FORCES & MOMENTS
C       :::::::
C
        IF ( MOMRPY .EQ. 1 ) THEN
                CALL MOMENTSET (RMOM, XC, PHIC, TAUX, TAUPHI, LX, LPHI, MTUB,
     1                          NTUB, IKP, MOMTYP)
                IF ( IKP .EQ. 1 )  GO TO 900
        ENDIF
C
30      IF ( F .NE. 0. ) THEN
        PL = F/( (LX*LPHI) -(LX - 2.*TAUX)*(LPHI - 2.*TAUPHI) )
        CALL RECTUBE (PL, XC, PHIC, TAUX, TAUPHI, LX, LPHI, MTUB, NTUB, IKP)
        IF ( IKP .EQ. 1 ) GO TO 900
        ENDIF
C
90      CONTINUE
        WRITE (6,95)
        WRITE (1,95)
95      FORMAT (/,1X,'    ---> END MULTIPLE NOZZLE DATA',/)
C
        GO TO 1000
C
900     WRITE (6,901)
        WRITE (1,901)
901     FORMAT(1X,'ERROR DETECTED IN -- IKP --')
C
1000    RETURN
        END
```

```
C     ************************************************************************
C
      SUBROUTINE CIRCSPLIT (XC, PHIC, R1, R2, PRS, NDIR)
C
C     SPLITS CIRCULAR ATTACHMENT INTO FOUR QUADRANTS THROUGH FOUR CALLS
C     TO "CIRCPART."
C
C     XC   = X-COORDINATE OF ATTACHMENT CENTER
C     PHIC = PHI-COORDINATE OF ATTACHMENT CENTER
C     R1   = INNER RADIUS OF ATTACHMENT
C     R2   = OUTER RADIUS OF ATTACHMENT
C     PRS  = LOADING
C     NDIR = LOADING DIRECTION
C
C     CREATED      : 18-NOV-83                        BY: F.M.G. WONG
C     LAST REVISED : 06-DEC-83
C
C     ************************************************************************
C
      REAL L, NU
      COMMON / CONSTANTS / A, E, L, NU, PI, T
C
      RADIAN = PI/180.
      ANG1 = 45.*RADIAN
      ANG2 = 135.*RADIAN
      ANG3 = 225.*RADIAN
      ANG4 = 315.*RADIAN
C
      CALL CIRCPART (XC, PHIC, R1, R2, ANG1, 1.0, 1.0, 1)
      CALL CIRCPART (XC, PHIC, R1, R2, ANG2, 1.0, -1.0, 2)
      CALL CIRCPART (XC, PHIC, R1, R2, ANG3, -1.0, -1.0, 3)
      CALL CIRCPART (XC, PHIC, R1, R2, ANG4, -1.0, 1.0, 4)
C
      CALL CIRCPRES (XC, PHIC, R1, R2, ANG1, 1.0, 1.0, 1, PRS, NDIR)
      CALL CIRCPRES (XC, PHIC, R1, R2, ANG2, 1.0, -1.0, 2, PRS, NDIR)
      CALL CIRCPRES (XC, PHIC, R1, R2, ANG3, -1.0, -1.0, 3, PRS, NDIR)
      CALL CIRCPRES (XC, PHIC, R1, R2, ANG4, -1.0, 1.0, 4, PRS, NDIR)
C
C     CALL TRIPATCH (XA, XB, PHIA, PHIB, PRS, IORIENT, 1)
      RETURN
      END
```

```
C       ********************************************************************
C
        SUBROUTINE CIRCPART (XC, PHIC, R1, R2, ANG, XFACT, PFACT, INDEX)
C
C       PARTITIONS A CIRCULAR ATTACHMENT INTO SEPARATE ELEMENTS
C
C       XC    =  X-COORDINATE OF ATTACHMENT CENTER
C       PHIC  =  PHI-COORDINATE OF ATTACHMENT CENTER
C       R1    =  INNER RADIUS OF ATTACHMENT
C       R2    =  OUTER RADIUS OF ATTACHMENT
C       ANG   =  ANGLE=45, WHICH DEFINES THE QUADRANT
C       XFACT =  DETERMINES SIGN OF X-DIRECTION
C       PFACT =  DETERMINES SIGN OF PHI-DIRECTION
C       INDEX =  INTEGER DESIGNATION OF QUADRANT: 1, 2, 3, OR 4
C
C       CREATED      : 18-NOV-83                         BY: F.M.G. WONG
C       LAST REVISED : 28-NOV-83
C
C       ********************************************************************
C
        REAL L, NU
        COMMON / CONSTANTS / A, E, L, NU, PI, T
        COMMON / CIRCTB / CIRCX(7,4), CIRCPHI(7,4)
        COMMON / CIRCSD / CIRCSX(3,4), CIRCSP(3,4)
C
C       ::::::: CALCULATE PROJECTED RADII IN PHI-DIRECTION
C
        PR1 = ASIN(R1/A)
        PR2 = ASIN(R2/A)
C
C
        ABSIN = ABS(SIN(ANG))
        ABCOS = ABS(COS(ANG))
C
        THD1 = 1./3.
        THD2 = 2./3.
C
C       :::::::
C       ::::::: ELEMENT PARTITION FOR A CIRCULAR TUBE
C       :::::::
C
        IF ( R1 .NE. 0.0 ) THEN
C
C       ::::::: CALCULATION OF CENTERS FOR RECTANGULAR ELEMENTS
C
        CIRCX(1,INDEX) = XC + 0.5*R1*SIN(ANG)
        CIRCPHI(1,INDEX) = PHIC + PFACT*(PR1 + (PR2 - PR1)/2.)
C
        CIRCX(2,INDEX) = XC +  R1*SIN(ANG) + 0.5*(R2 - R1)*SIN(ANG)
        CIRCPHI(2,INDEX) = PHIC + PR1*COS(ANG)+0.5*(PR2 - PR1)*COS(ANG)
C
        CIRCX(3,INDEX) = XC + XFACT*(R1 + (R2 - R1)/2.)
        CIRCPHI(3,INDEX) = PHIC + 0.5*PR1*COS(ANG)
C
C       ::::::: CALCULATION OF CENTERS FOR TRIANGULAR ELEMENTS
C
        CIRCX(4,INDEX) = XC + THD2*R1*SIN(ANG)
```

```
      CIRCPHI(4,INDEX) = PHIC + PR1*COS(ANG) + PFACT*THD2*(PR1 - PR1*ABCOS)
C
      CIRCX(5,INDEX) = XC + R1*SIN(ANG) + THD1*(R2 - R1)*SIN(ANG)
      CIRCPHI(5,INDEX) = PHIC+PR2*COS(ANG)+PFACT*THD1*(PR2-PR2*ABCOS)
C
      CIRCX(6,INDEX) = XC + R2*SIN(ANG) + XFACT*THD1*(R2 - R2*ABSIN)
      CIRCPHI(6,INDEX) = PHIC + PR1*COS(ANG) + THD1*(PR2 - PR1)*COS(ANG)
C
      CIRCX(7,INDEX) = XC + R1*SIN(ANG) + XFACT*THD2*(R1 - R1*ABSIN)
      CIRCPHI(7,INDEX) = PHIC + THD2*PR1*COS(ANG)
C
      ENDIF
C
C     :::::::
C     ::::::: ELEMENT PARTITION FOR A CIRCULAR SOLID
C     :::::::
C
      IF ( R1 .EQ. 0.0 ) THEN
C
      CIRCSX(1,INDEX) = XC + 0.5*R2*SIN(ANG)
      CIRCSP(1,INDEX) = PHIC + 0.5*PR2*COS(ANG)
C
      CIRCSX(2,INDEX) = XC + R2*SIN(ANG) + XFACT*THD1*(R2 - R2*ABSIN)
      CIRCSP(2,INDEX) = PHIC + THD1*R2*COS(ANG)
C
      CIRCSX(3,INDEX) = XC + THD1*R2*SIN(ANG)
      CIRCSP(3,INDEX) = PHIC + R2*COS(ANG) + PFACT*THD1*(R2 - R2*ABCOS)
C
      ENDIF
C
      RETURN
      END
```

```
C       ***********************************************************************
C
        SUBROUTINE CIRCPRES (XC, PHIC, R1, R2, ANG, XFACT, PFACT, IND, PRS,
     1                       NDIR)
C
C       APPLIES PRESSURE LOADINGS TO CIRCULAR ATTACHMENT
C
C       R1    = INNER RADIUS OF ATTACHMENT
C       R2    = OUTER RADIUS OF ATTACHMENT
C       ANG   = ANGLE DESIGNATING THE QUADRANT
C       XFACT = DETERMINES SIGN OF X-DIRECTION
C       PFACT = DETERMINES SIGN OF PHI-DIRECTION
C       IND   = INTEGER DESIGNATING QUADRANT
C       PRS   = PRESSURE LOADING TO BE UNIFORMLY DISTRIBUTED
C       NDIR  = DIRECTION OF LOADING
C
C       CREATED      : 18-NOV-83                          BY: F.M.G. WONG
C       LAST REVISED : 08-DEC-83
C
C       ***********************************************************************
C
        REAL L, NU
        DIMENSION THX(3,4), THP(3,4), TRIXA(4,4), TRIXB(4,4), TRIPA(4,4),
     1             TRIPB(4,4), IORIENT(16), ITYPE(4)
        DIMENSION  KORIENT(4)
        COMMON / CONSTANTS / A, E, L, NU, PI, T
        COMMON / CIRCTB / CIRCX(7,4), CIRCPHI(7,4)
        COMMON / CIRCSD / CIRCSX(3,4), CIRCSP(3,4)
C
        DATA IORIENT / 2, 3, 3, 2, 4, 1, 1, 4, 3, 2, 2, 3, 1, 4, 4, 1 /
        DATA ITYPE / 1, 5, 9, 13 /
        DATA KORIENT / 3, 1, 2, 4 /
C
        ABTRIG = COS(45.*PI/180.)
        PR1 = ASIN(R1/A)
        PR2 = ASIN(R2/A)
C
C       ::::::: LOADINGS FOR CIRCULAR TUBES
C
        IF ( R1 .NE. 0.0 )   THEN
C
        THX(1,IND) = 0.5*R1*ABTRIG
        THP(1,IND) = 0.5*(PR2 - PR1)
C
        THX(2,IND) = 0.5*(R2 - R1)*ABTRIG
        THP(2,IND) = 0.5*(PR2 - PR1)*ABTRIG
C
        THX(3,IND) = 0.5*(R2 - R1)
        THP(3,IND) = 0.5*PR1*ABTRIG
C
        PRINT *, ' RECTANGLES '
        DO 40 I = 1, 3
                CIRX = CIRCX(I,IND)
                CIRPHI = CIRCPHI(I,IND)
                PSI = CIRPHI - PHIC
                PR = PRS*(COS(PSI))**2
                PPHI = 0.5*PRS*SIN(2.*PSI)
```

```
                CALL PATCH (CIRX, CIRPHI, THX(I,IND), THP(I,IND), PR, 1)
                CALL PATCH (CIRX, CIRPHI, THX(I,IND), THP(I,IND), PPHI, 2)
C               PRINT *, CIRX, CIRPHI, PR, I
C               PRINT *, CIRX, CIRPHI, PPHI, I
40      CONTINUE
C
        TRIXA(1,IND) = XC + R1*SIN(ANG)
        TRIXB(1,IND) = XC
        TRIPA(1,IND) = PHIC + PR1*COS(ANG)
        TRIPB(1,IND) = PHIC + PR1*PFACT
C
        TRIXA(2,IND) = XC + R2*SIN(ANG)
        TRIXB(2,IND) = XC + R1*SIN(ANG)
        TRIPA(2,IND) = PHIC + PR2*COS(ANG)
        TRIPB(2,IND) = PHIC + PR2*PFACT
C
        TRIXA(3,IND) = XC + R2*XFACT
        TRIXB(3,IND) = XC + R2*SIN(ANG)
        TRIPA(3,IND) = PHIC + PR1*COS(ANG)
        TRIPB(3,IND) = PHIC + PR2*COS(ANG)
C
        TRIXA(4,IND) = XC + R1*XFACT
        TRIXB(4,IND) = XC + R1*SIN(ANG)
        TRIPA(4,IND) = PHIC
        TRIPB(4,IND) = PHIC + PR1*COS(ANG)
C
        PRINT *, 'IND =',IND
        IF ( (IND .EQ. 2) .OR. (IND .EQ. 3) ) THEN
                DO 45 J = 1, 4
                        TEMPA = TRIXA(J,IND)
                        TEMPB = TRIXB(J,IND)
                        TRIXB(J,IND) = TEMPA
                        TRIXA(J,IND) = TEMPB
                        TEMPC = TRIPA(J,IND)
                        TEMPD = TRIPB(J,IND)
                        TRIPA(J,IND) = TEMPD
                        TRIPB(J,IND) = TEMPC
45              CONTINUE
        ENDIF
C
        KTYP = ITYPE(IND)
        DO 50 K = 1, 4
C
        PSI = CIRCPHI(K+3,IND) - PHIC
        PR = PRS*(COS(PSI))**2
        PPHI = 0.5*PRS*SIN(2.*PSI)
C
        CALL TRIPATCH (TRIXA(K,IND),TRIXB(K,IND),TRIPA(K,IND),
     1  TRIPB(K,IND),PR,IORIENT(KTYP),1)
        CALL TRIPATCH (TRIXA(K,IND),TRIXB(K,IND),TRIPA(K,IND),
     1  TRIPB(K,IND),PPHI,IORIENT(KTYP),2)
C
        KTYP = KTYP + 1
C
50      CONTINUE
C
```

```
      ENDIF
C
C     :::::: LOADINGS FOR SOLID CIRCULAR ATTACHMENTS
C
      IF ( R1 .EQ. 0.0 )   THEN
C
      THX(1,IND) = 0.5*R2*ABTRIG
      THP(1,IND) = 0.5*PR2*ABTRIG
C
      CIRX = CIRCSX(1,IND)
      CIRPHI = CIRCSP(1,IND)
      PSI = CIRPHI - PHIC
      PR = PRS*(COS(PSI))**2
      PPHI = 0.5*PRS*SIN(2.*PSI)
C
      CALL PATCH (CIRX, CIRPHI, THX(1,IND), THP(1,IND), PR, 1)
      CALL PATCH (CIRX, CIRPHI, THX(1,IND), THP(1,IND), PPHI, 2)
C
      TRIXA(1,IND) = XC + XFACT*R2
      TRIXB(1,IND) = XC + R2*SIN(ANG)
      TRIPA(1,IND) = PHIC
      TRIPB(1,IND) = PHIC + PR2*COS(ANG)
C
      TRIXA(2,IND) = XC + R2*SIN(ANG)
      TRIXB(2,IND) = XC
      TRIPA(2,IND) = PHIC + PR2*COS(ANG)
      TRIPB(2,IND) = PHIC + PFACT*PR2
C
      PRINT *,'IND =',IND
      IF ( (IND .EQ. 2) .OR. (IND .EQ. 3) ) THEN
              DO 55 J = 1, 4
                      TEMPA = TRIXA(J,IND)
                      TEMPB = TRIXB(J,IND)
                      TRIXB(J,IND) = TEMPA
                      TRIXA(J,IND) = TEMPB
                      TEMPC = TRIPA(J,IND)
                      TEMPD = TRIPB(J,IND)
                      TRIPA(J,IND) = TEMPD
                      TRIPB(J,IND) = TEMPC
55            CONTINUE
      ENDIF
C
      DO 60 K = 1, 2
C
      PSI = CIRCSP(K+1,IND) - PHIC
      PR = PRS*(COS(PSI))**2
      PPHI = 0.5*PRS*SIN(2.*PSI)
C
      CALL TRIPATCH (TRIXA(K,IND),TRIXB(K,IND),TRIPA(K,IND),TRIPB(K,IND),
     1 PR,KORIENT(IND),1)
      CALL TRIPATCH (TRIXA(K,IND),TRIXB(K,IND),TRIPA(K,IND),TRIPB(K,IND),
     1 PPHI,KORIENT(IND),2)
C
60    CONTINUE
C
      ENDIF
```

```
C
C
      RETURN
      END
```

```
C      ************************************************************************
C
       SUBROUTINE TRIPATCH (XA, XB, PHIA, PHIB, PRS, IORIENT, N1)
C
C      PERFORMS LOADING DISTIBUTION OF TRIANGULAR PATCH AREA WITH
C      DIMENSIONS: XA, XB, PHIA, PHIB.
C
C      PRS     = LOADING TO BE DISTRIBUTED OVER TRIANGLE
C      IORIENT = ORIENT OF VERTEX OF TRIANGLE:
C                  1:        2:      3:       4:
C                    *      *****     *        *****
C                   * *      *  *     * *      *  *
C                  *  *       * *     *  *     * *
C                 *****        *      *****    *
C
C      N1      = 1 : W, 2 : V, 3 : X
C
C      CREATED      : 21-NOV-83                  BY : F.M.G. WONG
C      LAST REVISED : 30-JAN-84
C
C      ************************************************************************
C
       REAL L, NU
       COMMON / CONSTANTS / A, E, L, NU, PI, T
       COMMON / INDEXES / M, N
       COMMON / LOADS / PRE(100,100), PRO(100,100), PXE(100,100),
     1                 PXO(100,100), PPHIE(100,100), PPHIO(100,100)
C
C      IORIENT = 1
       PRINT *,XA,XB,PHIA,PHIB,PRS,IORIENT,N1
       ONE = 1.0
       IF ( (IORIENT.EQ.2) .OR. (IORIENT.EQ.4) ) ONE = -1.0
       THP = (PHIB - PHIA)/2.
       CNTP = PHIA + THP
       IF ( IORIENT .EQ. 3 ) THEN
               THX = (XA - XB)/2.
               CNTX = XB + THX
               CALL PATCH (CNTX,CNTP,THX,THP,PRS,N1)
       ENDIF
       IF ( IORIENT .EQ. 4 ) THEN
               THX = (XB - XA)/2.
               CNTX = XA + THX
               CALL PATCH (CNTX,CNTP,THX,THP,PRS,N1)
       ENDIF
C
C      XA = 0.
C      XB = 6.
C      XB = 0.
C      XA = 6.
C      PHIA = 0.
C      PHIB = 0.75038286
C
C      XA = 68.
C      XB = 62.
C      XA = 62.
C      XB = 68.
C      PHIA = -0.137881
```

```
C       PHIB = 0.137881
C       PRS = 10.
C
C       PRINT *,XA,XB,PHIA,PHIB,PRS
        AA = (XB - XA)/(PHIB - PHIA)
        B = (XA*PHIB - XB*PHIA)/(PHIB - PHIA)
        PI2 = PI*PI
        PRINT *, ' AA = ',AA,' B =',B
C
        DO 200 MT = 0, M-1
                IF ( MT .EQ. 0 ) GO TO 5
                XMT = FLOAT(MT)
                CPHIO = COS(XMT*PHIO)
                SPHIO = SIN(XMT*PHIO)
5               CONTINUE
                DO 100 NT = 1, N
                        XNT = FLOAT(NT)
                        XK = XNT*PI*AA/L
C                       SXO = SIN(XNT*PI*XO/L)
C                       CXO = COS(XNT*PI*XO/L)
                        XMKPV = 1./(XMT + XK)
                        DUM = XMT - XK
                        IF ( DUM .EQ. 0.0 ) PRINT*,DUM,XMT,XK,MT,NT
                        XMKNV = 1./(XMT - XK)
C
                        CSXA = COS(XNT*PI*XA/L)
                        SSXA = SIN(XNT*PI*XA/L)
                        CSB = 0.5*COS(XNT*PI*B/L)
                        SSB = 0.5*SIN(XNT*PI*B/L)
                        IF (MT .NE. 0) THEN
                        SMPA = (1./XMT)*SIN(XMT*PHIA)
                        CMPA = (1./XMT)*COS(XMT*PHIA)
                        SMKPA = XMKPV*SIN((XMT+XK)*PHIA)
                        SMKNA = XMKNV*SIN((XMT-XK)*PHIA)
                        CMKPA = XMKPV*COS((XMT+XK)*PHIA)
                        CMKNA = XMKNV*COS((XMT-XK)*PHIA)
                        SMPB = (1./XMT)*SIN(XMT*PHIB)
                        CMPB = (1./XMT)*COS(XMT*PHIB)
                        SMKPB = XMKPV*SIN((XMT+XK)*PHIB)
                        SMKNB = XMKNV*SIN((XMT-XK)*PHIB)
                        CMKPB = XMKPV*COS((XMT+XK)*PHIB)
                        CMKNB = XMKNV*COS((XMT-XK)*PHIB)
                        ENDIF
C
                        IF ( MT .EQ. 0 ) THEN
C
                        CKPA = COS(XK*PHIA)
                        SKPA = SIN(XK*PHIA)
                        CKPB = COS(XK*PHIB)
                        SKPB = SIN(XK*PHIB)
                        XKV = 1./XK
C
                        FUNA = PHIA*CSXA-XKV*2.*CSB*SKPA-XKV*2.*SSB*CKPA
                        FUNB = PHIB*CSXA-XKV*2.*CSB*SKPB-XKV*2.*SSB*CKPB
                        FUNC = FUNB - FUNA
C
```

```
      FUND = -PHIA*SSXA-XKV*2.*CSB*CKPA+XKV*2.*SSB*SKPA
      FUNE = -PHIB*SSXA-XKV*2.*CSB*CKPB+XKV*2.*SSB*SKPB
      FUNF = FUNE - FUND
C
      CON = 2.*PRS*ONE/(XNT*PI2)
C
      IF (N1 .EQ. 1) PRE(MT+1,NT) = PRE(MT+1,NT)+CON*FUNC
      IF (N1.EQ.2) PPHIO(MT+1,NT)=PPHIO(MT+1,NT)+CON*FUNC
      IF (N1 .EQ. 3) PXE(MT+1,NT) = PXE(MT+1,NT)+CON*FUNF
C     PRINT *, ' M=0:  ',IORIENT,CON*FUNC*ONE,NT
C
      GO TO 100
      ENDIF
C
      IF ((N1 .EQ. 1).OR.(N1 .EQ. 2)) THEN
C
      FUNA = CSXA*SMPA-CSB*(SMKPA+SMKNA)+SSB*(-CMKPA+CMKNA)
      FUNB = CSXA*SMPB-CSB*(SMKPB+SMKNB)+SSB*(-CMKPB+CMKNB)
      FUNC = FUNB - FUNA
C
      FUND = -CSXA*CMPA-CSB*(-CMKPA-CMKNA)+SSB*(-SMKPA+SMKNA)
      FUNE = -CSXA*CMPB-CSB*(-CMKPB-CMKNB)+SSB*(-SMKPB+SMKNB)
      FUNF = FUNE - FUND
C
      CON = 4.*PRS*ONE/(XNT*PI2)
C
      IF (N1 .EQ. 1) THEN
              PRE(MT+1,NT) = PRE(MT+1,NT) + CON*FUNC
              PRO(MT+1,NT) = PRO(MT+1,NT) + CON*FUNF
      ENDIF
C     PRINT *, IORIENT, CON*FUNC*ONE, CON*FUNF*ONE
      IF (N1 .EQ. 2) THEN
              PPHIE(MT+1,NT)=PPHIE(MT+1,NT)+CON*FUNF
              PPHIO(MT+1,NT)=PPHIO(MT+1,NT)+CON*FUNC
      ENDIF
C
      ENDIF
C
      IF (N1 .EQ. 3) THEN
C
      CXO = COS(XNT*PI*XO/L)
C
      FUNA = -SSXA*SMPA+CSB*(-CMKPA+CMKNA)+SSB*(SMKPA+SMKNA)
      FUNB = -SSXA*SMPB+CSB*(-CMKPB+CMKNB)+SSB*(SMKPB+SMKNB)
      FUNC = FUNB - FUNA
C
      FUND = SSXA*CMPA+CSB*(-SMKPA+SMKNA)+SSB*(-CMKPA-CMKNA)
      FUNE = SSXA*CMPB+CSB*(-SMKPB+SMKNB)+SSB*(-CMKPB-CMKNB)
      FUNF = FUNE - FUND
C
      CON = 4.*PRS/(XNT*PI2)
C
      PXE(MT+1,NT) = PXE(MT+1,NT) + CON*FUNC*ONE
      PXO(MT+1,NT) = PXO(MT+1,NT) + CON*FUNF*ONE
C
      ENDIF
```

```
C
C
100             CONTINUE
200     CONTINUE
C
C
        RETURN
        END
```

```
C      ***********************************************************************
C
       SUBROUTINE CIRCSET (XC, PHIC, R1, R2, RMOM, MOMTYP)
C
C      SPLITS CIRCULAR ATTACHMENT INTO FOUR QUADRANTS THROUGH FOUR CALLS
C      TO "CIRCPART."
C
C      XC    = X-COORDINATE OF ATTACHMENT CENTER
C      PHIC  = PHI-COORDINATE OF ATTACHMENT CENTER
C      R1    = INNER RADIUS OF ATTACHMENT
C      R2    = OUTER RADIUS OF ATTACHMENT
C      RMOM  = EXTERNAL MOMENT
C      MOMTYP = TYPE OF MOMENT: LONG. = 1, CIRCUM = 2, NONE = 0
C
C      CREATED      : 01-DEC-83                    BY: F.M.G. WONG
C      LAST REVISED : 01-DEC-83
C
C      ***********************************************************************
C
       REAL L, NU
       COMMON / CONSTANTS / A, E, L, NU, PI, T
C
       RADIAN = PI/180.
       ANG1 = 45.*RADIAN
       ANG2 = 135.*RADIAN
       ANG3 = 225.*RADIAN
       ANG4 = 315.*RADIAN
C
       CALL CIRCPART (XC, PHIC, R1, R2, ANG1, 1.0, 1.0, 1)
       CALL CIRCPART (XC, PHIC, R1, R2, ANG2, 1.0, -1.0, 2)
       CALL CIRCPART (XC, PHIC, R1, R2, ANG3, -1.0, -1.0, 3)
       CALL CIRCPART (XC, PHIC, R1, R2, ANG4, -1.0, 1.0, 4)
C
       CALL CIRCINERT (R1, R2, ERT)
C
       CALL CIRCMOM (XC, PHIC, R1, R2, ANG1, 1.0, 1.0, 1, RMOM, ERT, MOMTYP)
       CALL CIRCMOM (XC, PHIC, R1, R2, ANG2, 1.0, -1.0, 2, RMOM, ERT, MOMTYP)
       CALL CIRCMOM (XC, PHIC, R1, R2, ANG3, -1.0, -1.0, 3, RMOM, ERT, MOMTYP)
       CALL CIRCMOM (XC, PHIC, R1, R2, ANG4, -1.0, 1.0, 4, RMOM, ERT, MOMTYP)
C
       RETURN
       END
C
C
C      ***********************************************************************
C
       SUBROUTINE CIRCINERT (R1, R2, ERT)
C
C      CALCULATES THE MOMENT OF INERTIA FOR A CIRCULAR ATTACHMENT
C
C      CREATED : 01-DEC-83                   BY: F.M.G. WONG
C
C      ***********************************************************************
C
       COMMON / CONSTANTS / A, E, L, NU, PI, T
C
       ERT = (R2**4 - R1**4)*PI/4.
```

```
C
C
      RETURN
      END
```

```
C       ********************************************************************
C
        SUBROUTINE CIRCMOM (XC, PHIC, R1, R2, ANG, XFACT, PFACT, IND, RMOM,
     1                       ERTIA, MOMTYP)
C
C       APPLIES PRESSURE LOADINGS TO CIRCULAR ATTACHMENT
C
C       R1     = INNER RADIUS OF ATTACHMENT
C       R2     = OUTER RADIUS OF ATTACHMENT
C       ANG    = ANGLE DESIGNATING THE QUADRANT
C       XFACT  = DETERMINES SIGN OF X-DIRECTION
C       PFACT  = DETERMINES SIGN OF PHI-DIRECTION
C       IND    = INTEGER DESIGNATING QUADRANT
C       RMOM   = EXTERNAL MOMENT TO BE APPLIED
C       ERTIA  = MOMENT OF INERTIA
C       MOMTYP = MOMENT TYPE : LONGITUDINAL = 1, CIRCUMFERENTIAL = 2
C
C       CREATED      : 01-DEC-83                        BY: F.M.G. WONG
C       LAST REVISED : 08-DEC-83
C
C       ********************************************************************
C
        REAL L, NU
        DIMENSION THX(3,4), THP(3,4), TRIXA(4,4), TRIXB(4,4), TRIPA(4,4),
     1            TRIPB(4,4), IORIENT(16), ITYPE(4)
        DIMENSION  KORIENT(4)
        COMMON / CONSTANTS / A, E, L, NU, PI, T
        COMMON / CIRCTB / CIRCX(7,4), CIRCPHI(7,4)
        COMMON / CIRCSD / CIRCSX(3,4), CIRCSP(3,4)
C
        DATA IORIENT / 2, 3, 3, 2, 4, 1, 1, 4, 3, 2, 2, 3, 1, 4, 4, 1 /
        DATA ITYPE / 1, 5, 9, 13 /
        DATA KORIENT / 3, 1, 2, 4 /
C
        ABTRIG = COS(45.*PI/180.)
        PR1 = ASIN(R1/A)
        PR2 = ASIN(R2/A)
C
C       ::::::: LOADINGS FOR CIRCULAR TUBES
C
        IF ( R1 .NE. 0.0 )   THEN
C
        THX(1,IND) = 0.5*R1*ABTRIG
        THP(1,IND) = 0.5*(PR2 - PR1)
C
        THX(2,IND) = 0.5*(R2 - R1)*ABTRIG
        THP(2,IND) = 0.5*(PR2 - PR1)*ABTRIG
C
        THX(3,IND) = 0.5*(R2 - R1)
        THP(3,IND) = 0.5*PR1*ABTRIG
C
        DO 40 I = 1, 3
                CIRX = CIRCX(I,IND)
                CIRPHI = CIRCPHI(I,IND)
                PSI = CIRPHI - PHIC
                IF ( MOMTYP .EQ. 1 ) ELD = CIRX - XC
                IF ( MOMTYP .EQ. 2 ) ELD = A*SIN( CIRPHI - PHIC )
```

```
               PRS = RMOM*ELD/ERTIA
               PR = PRS*(COS(PSI))**2
               PPHI = 0.5*PRS*SIN(2.*PSI)
               CALL PATCH (CIRX, CIRPHI, THX(I,IND), THP(I,IND), PR, 1)
               CALL PATCH (CIRX, CIRPHI, THX(I,IND), THP(I,IND), PPHI, 2)
40       CONTINUE
C
         TRIXA(1,IND) = XC + R1*SIN(ANG)
         TRIXB(1,IND) = XC
         TRIPA(1,IND) = PHIC + PR1*COS(ANG)
         TRIPB(1,IND) = PHIC + PR1*PFACT
C
         TRIXA(2,IND) = XC + R2*SIN(ANG)
         TRIXB(2,IND) = XC + R1*SIN(ANG)
         TRIPA(2,IND) = PHIC + PR2*COS(ANG)
         TRIPB(2,IND) = PHIC + PR2*PFACT
C
         TRIXA(3,IND) = XC + R2*XFACT
         TRIXB(3,IND) = XC + R2*SIN(ANG)
         TRIPA(3,IND) = PHIC + PR1*COS(ANG)
         TRIPB(3,IND) = PHIC + PR2*COS(ANG)
C
         TRIXA(4,IND) = XC + R1*XFACT
         TRIXB(4,IND) = XC + R1*SIN(ANG)
         TRIPA(4,IND) = PHIC
         TRIPB(4,IND) = PHIC + PR1*COS(ANG)
C
         PRINT *, 'IND =',IND
         IF ( (IND .EQ. 2) .OR. (IND .EQ. 3) ) THEN
               DO 45 J = 1, 4
                     TEMPA = TRIXA(J,IND)
                     TEMPB = TRIXB(J,IND)
                     TRIXB(J,IND) = TEMPA
                     TRIXA(J,IND) = TEMPB
                     TEMPC = TRIPA(J,IND)
                     TEMPD = TRIPB(J,IND)
                     TRIPA(J,IND) = TEMPD
                     TRIPB(J,IND) = TEMPC
45             CONTINUE
         ENDIF
C
         KTYP = ITYPE(IND)
         DO 50 K = 1, 4
C
         PSI = CIRCPHI(K+3,IND) - PHIC
         IF ( MOMTYP .EQ. 1 ) ELD = CIRCX(K+3,IND) - XC
         IF ( MOMTYP .EQ. 2 ) ELD = A*SIN( CIRCPHI(K+3,IND) - PHIC )
         PRS = RMOM*ELD/ERTIA
         PR = PRS*(COS(PSI))**2
         PPHI = 0.5*PRS*SIN(2.*PSI)
C
         CALL TRIPATCH (TRIXA(K,IND),TRIXB(K,IND),TRIPA(K,IND),
     1    TRIPB(K,IND),PR,IORIENT(KTYP),1)
         CALL TRIPATCH (TRIXA(K,IND),TRIXB(K,IND),TRIPA(K,IND),
     1    TRIPB(K,IND),PPHI,IORIENT(KTYP),2)
C
```

```
      KTYP = KTYP + 1
C
50    CONTINUE
C
      ENDIF
C
C     ::::::: LOADINGS FOR SOLID CIRCULAR ATTACHMENTS
C
      IF ( R1 .EQ. 0.0 )   THEN
C
      THX(1,IND) = 0.5*R2*ABTRIG
      THP(1,IND) = 0.5*PR2*ABTRIG
C
      CIRX = CIRCSX(1,IND)
      CIRPHI = CIRCSP(1,IND)
C
      PSI = CIRPHI - PHIC
      IF ( MOMTYP .EQ. 1 ) ELD = CIRX - XC
      IF ( MOMTYP .EQ. 2 ) ELD = A*SIN( CIRPHI - PHIC )
      PRS = -RMOM*ELD/ERTIA
      PR = PRS*(COS(PSI))**2
      PPHI = 0.5*PRS*SIN(2.*PSI)
C
      CALL PATCH (CIRX, CIRPHI, THX(1,IND), THP(1,IND), PR, 1)
      CALL PATCH (CIRX, CIRPHI, THX(1,IND), THP(1,IND), PPHI, 2)
C
      TRIXA(1,IND) = XC + XFACT*R2
      TRIXB(1,IND) = XC + R2*SIN(ANG)
      TRIPA(1,IND) = PHIC
      TRIPB(1,IND) = PHIC + PR2*COS(ANG)
C
      TRIXA(2,IND) = XC + R2*SIN(ANG)
      TRIXB(2,IND) = XC
      TRIPA(2,IND) = PHIC + PR2*COS(ANG)
      TRIPB(2,IND) = PHIC + PFACT*PR2
C
      PRINT *,'IND =',IND
      IF ( (IND .EQ. 2) .OR. (IND .EQ. 3) ) THEN
              DO 55 J = 1, 4
                      TEMPA = TRIXA(J,IND)
                      TEMPB = TRIXB(J,IND)
                      TRIXB(J,IND) = TEMPA
                      TRIXA(J,IND) = TEMPB
                      TEMPC = TRIPA(J,IND)
                      TEMPD = TRIPB(J,IND)
                      TRIPA(J,IND) = TEMPD
                      TRIPB(J,IND) = TEMPC
55            CONTINUE
      ENDIF
C
      DO 60 K = 1, 2
C
      PSI = CIRCSP(K+1,IND) - PHIC
      IF ( MOMTYP .EQ. 1 ) ELD = CIRCSX(K+1,IND) - XC
      IF ( MOMTYP .EQ. 2 ) ELD = A*SIN( CIRCSP(K+1,IND) - PHIC )
      PRS = RMOM*ELD/ERTIA
```

```
      PR = PRS*(COS(PSI))**2
      PPHI = 0.5*PRS*SIN(2.*PSI)
C
      CALL TRIPATCH (TRIXA(K,IND),TRIXB(K,IND),TRIPA(K,IND),TRIPB(K,IND),
     1 PR,KORIENT(IND),1)
      CALL TRIPATCH (TRIXA(K,IND),TRIXB(K,IND),TRIPA(K,IND),TRIPB(K,IND),
     1 PPHI,KORIENT(IND),1)
C
60    CONTINUE
C
      ENDIF
C
C
      RETURN
      END
```

```
C      ******************************************************************
C
       SUBROUTINE COLOCATE (XC, PHIC, TAUX, TAUPHI, LX, LPHI, M, N, IGEOM)
C
C      DRIVER MODULE FOR THE COLOCATION METHOD FOR RIGID PLUG PROBLEMS.
C
C      IGEOM = TYPE OF ATTACHMENT GEOMETRY: RECTANGULAR = 1, CIRCULAR = 2
C
C      CREATED          :  05-OCT-83            BY :  F.M.G. WONG
C      LATEST REVISION  :  02-DEC-83            BY :  F.M.G. WONG
C
C      ******************************************************************
C
C
       REAL LX, LPHI
       DIMENSION TEMPA(8,8), TEMPB(32,32), TEMPC(72,72), TEMPD(128,128),
     1           TEMPE(24,24), DISPL(64), FOR(64)
       COMMON / CONSTANTS / A, E, L, NU, PI, T
       COMMON / CENTELM / CNTXEL(4,4,4), CNTPHIEL(4,4,4)
       COMMON / STIFMAT / STIFFA(8,8), STIFFB(32,32), STIFFC(72,72),
     1                    STIFFD(128,128), STIFFE(24,24)
       COMMON / CIRCSD / CIRCSX(3,4), CIRCSP(3,4)
C
C      :::::::  COMPUTE STIFFNESS MATRIX BY APPLYING UNIT PRESSURES
C      :::::::  ON ATTACHMENT AREA AND SOLVING FOR THE DEFLECTIONS.
C
       IF (IGEOM.EQ.1) CALL PATSPLIT (XC, PHIC, TAUX, TAUPHI, LX, LPHI, M, N)
       IF (IGEOM.EQ.2) CALL CIRCOLSPL(XC, PHIC, 0.0, TAUX)
C
C      :::::::  STIFFNESS MATRIX IS NOW DEFINED.  PRESSURES MAY NOW
C      :::::::  BE CALCULATED, GIVEN A KNOWN DISPLACEMENT.
C
       IF ( IGEOM .EQ. 1 )  ISIZE = M*N*4
       IF ( IGEOM .EQ. 2 )  ISIZE = 12
C
       DO 10 I = 1, ISIZE*2
               DO 5  J = 1, ISIZE*2
                       IF ( ISIZE .EQ. 4 )  TEMPA(I,J) = STIFFA(I,J)
                       IF ( ISIZE .EQ. 16 ) TEMPB(I,J) = STIFFB(I,J)
                       IF ( ISIZE .EQ. 36 ) TEMPC(I,J) = STIFFC(I,J)
                       IF ( ISIZE .EQ. 64 ) TEMPD(I,J) = STIFFD(I,J)
                       IF ( ISIZE .EQ. 12 ) TEMPE(I,J) = STIFFE(I,J)
5              CONTINUE
10     CONTINUE
C
C      :::::::  ASSEMBLE KNOWN DISPLACEMENT MATRIX
C
       INC = 1
       PRINT *
C
       DO 40 J = 1, 4
C
               IF ( IGEOM .EQ. 1 ) THEN
               DO 25 K = 1, N
                       DO 20 L = 1, M
                       DISPL(INC) = COS( CNTPHIEL(K,L,J) - PHIC )
                       DISPL(INC+ISIZE) = SIN( CNTPHIEL(K,L,J) - PHIC )
```

```
                        INC = INC + 1
20                  CONTINUE
25              CONTINUE
                ENDIF
C
                IF ( IGEOM .EQ. 2 ) THEN
                DO 30 K = 1, 3
                        DISPL(INC) = COS( CIRCSP(K,J) - PHIC )
                        DISPL(INC+ISIZE) = SIN( CIRCSP(K,J) - PHIC )
                        PRINT *,'DISPR =',DISPL(INC),' DISPP=',DISPL(INC+ISIZE)
                        INC = INC + 1
30              CONTINUE
                ENDIF
C
40      CONTINUE
C
C       :::::::  SOLVE MATRIX EQUATION --
C       :::::::
C       :::::::       [ STIFF ] X [ PRESSURES ] = [ DEFLECTION ]
C       :::::::
C       :::::::  FOR PRESSURES.
C
        WRITE (1,100)
        WRITE (6,100)
100     FORMAT (1X,'----> SOLVING MATRIX',/)
        CALL TIMESTAT (1)
C
        DO 300 K = 1, ISIZE*2
                WRITE (1,200) TEMPE(K,1), TEMPE(K,13)!( TEMPE(K,L), L=1,ISIZE*2 ), DISPL(K)
                WRITE (6,200) TEMPE(K,1), TEMPE(K,13)!( TEMPE(K,L), L=1,ISIZE*2 ), DISPL(K)
200             FORMAT (1X,'TEMPE = ',E12.4,3X,'TEMPE13 = ',E12.4)
C200            FORMAT (<ISIZE*2>(E12.4,1X),2X,'DSP= ',E12.4)
300     CONTINUE
C
        IF ( ISIZE .EQ. 4 )    CALL SOLVE ( TEMPA, FOR, DISPL, 8, DET)
        IF ( ISIZE .EQ. 16 )   CALL SOLVE ( TEMPB, FOR, DISPL, 32, DET)
        IF ( ISIZE .EQ. 36 )   CALL SOLVE ( TEMPC, FOR, DISPL, 72, DET)
        IF ( ISIZE .EQ. 64 )   CALL SOLVE ( TEMPD, FOR, DISPL, 128, DET)
        IF ( ISIZE .EQ. 12 )   CALL SOLVE ( TEMPE, FOR, DISPL, 24, DET)
C
        CALL TIMESTAT (2)
        WRITE (6,800)
        WRITE (1,800)
800     FORMAT (/,1X,'THE COLOCATION FORCES ARE:',/)
C
        DO 900 I = 1, ISIZE*2
                WRITE (1,910) I, FOR(I)
                WRITE (6,910) I, FOR(I)
910             FORMAT (1X,'FOR(',I2,') = ',E12.4,/)
900     CONTINUE
C
        RETURN
        END
```

```
C
C     ******************************************************************
C
      SUBROUTINE PATSPLIT (XC, PHIC, TAUX, TAUPHI, LX, LPHI, M, N)
C
C     SPLITS AREA OF ATTACHMENT INTO 4 REGIONS.
C     SETS UP "ZEROED" PRESSURE MATRICES IN EACH REGION THROUGH 4
C     CALLS TO PATPRES.
C
C     XC     = X-COORDINATE OF TUBE CENTER (INCHES)
C     PHIC   = PHI-COORDINATE OF TUBE CENTER (RADIANS)
C     TAUX   = THICKNESS OF ATTACHMENT TUBE ALONG X (INCHES)
C     TAUPHI = THICKNESS OF ATTACHMENT TUBE ALONG PHI (INCHES)
C     LPHI   = OUTER HORIZONTAL DIMENSION (INCHES) OF ATTACHMENT TUBE
C     LX     = OUTER VERTICAL DIMENSION (INCHES) OF ATTACHMENT TUBE
C
C     M,N ARE ELEMENT MEMBERS OF PATCH IN PHI AND X DIRECTIONS, RESP.
C
C     CREATED            :  05-OCT-83     BY :  F.M.G. WONG
C     LATEST REVISION    :  03-NOV-83     BY :  F.M.G. WONG
C
C     ******************************************************************
C
      REAL L, NU, LPHI, LX
      COMMON / CONSTANTS / A, E, L, NU, PI, T
C
C
      ALPO = ASIN(LPHI/(2.*A))
      ALPI = ASIN((LPHI - 2.*TAUPHI)/(2.*A))
      V1 = .5*(ALPO + ALPI)
      V2 = LX-2.*TAUX
      V3 = .5*(LX - TAUX)
C
C
      ANG = ALPO - ALPI
      V4 = 2.*ALPO
C
      ISIZE = 4*M*N
C
      WRITE (1,100)
      WRITE (6,100)
100   FORMAT (1X,'----->  CALCULATING ELEMENT CENTERS',/)
      CALL TIMESTAT (1)
C
      CALL ELMCNT (M, N, XC, PHIC, 0, V1, V2, ANG, 1, IKP)
      CALL ELMCNT (M, N, XC, PHIC, V3, 0, TAUX, V4, 2, IKP)
      CALL ELMCNT (M, N, XC, PHIC, 0, -V1, V2, ANG, 3, IKP)
      CALL ELMCNT (M, N, XC, PHIC, -V3, 0, TAUX, V4, 4, IKP)
C
      CALL TIMESTAT (3)
C
C     ::::::: OBTAIN COEFFICIENTS FOR RADIAL UNIT LOAD
C
      IGLOB = 1
      IFOR = 1
C
      CALL PATPRES (M, N, XC, PHIC, 0, V1, V2, ANG, 1, IGLOB, IFOR, IKP)
```

```
C       CALL PATPRES (M, N, XC, PHIC, V3, 0, TAUX, V4, 2, IGLOB, IFOR, IKP)
C       CALL PATPRES (M, N, XC, PHIC, 0, -V1, V2, ANG, 3, IGLOB, IFOR, IKP)
C       CALL PATPRES (M, N, XC, PHIC, -V3, 0, TAUX, V4, 4, IGLOB, IFOR, IKP)
C
C       ::::::: OBTAIN COFFICIENTS FOR SHEAR UNIT LOAD
C
        IGLOB = ISIZE + 1
        IFOR = 2
        CALL INITIALIZE
C
C       CALL PATPRES (M, N, XC, PHIC, 0, V1, V2, ANG, 1, IGLOB, IFOR, IKP)
C       CALL PATPRES (M, N, XC, PHIC, V3, 0, TAUX, V4, 2, IGLOB, IFOR, IKP)
C       CALL PATPRES (M, N, XC, PHIC, 0, -V1, V2, ANG, 3, IGLOB, IFOR, IKP)
C       CALL PATPRES (M, N, XC, PHIC, -V3, 0, TAUX, V4, 4, IGLOB, IFOR, IKP)
C
        RETURN
        END
```

```
C
C     ******************************************************************
C
      SUBROUTINE PATPRES (M, N, XC, PHIC, XL, PHIL, THX, THPHI, IND, IGLOB,
     1                    IFOR, IKP)
C
C     SETS UP "ZEROED" PRESSURE MATRICES, AND THEN SUPERPOSES THE
C     UNIT PRESSURE AS REQUIRED BY COLOCATION.
C
C     M          = NO. OF PHI DIRECTION ELEMENTS IN PATCH
C     N          = NO. OF X DIRECTION ELEMENTS IN PATCH
C     XC         = LOCAL PATCH REFERENCE X COORDINATE
C     PHIC       = LOCAL PATCH REFERENCE PHI COORDINATE
C     (XC,PHIC) DEFINE DIRECTION OF LATERAL LOAD AXIS ON CYLINDER
C     XL         = X COOR DISTANCE FROM XC TO PATCH CENTROID
C     PHIL       = PHI(ANGLE) DISTANCE FROM PHIC TO PATCH PHI CENTROID
C     THX        = PATCH DIMENSION ALONG X - AXIS
C     THPHI      = PATCH ARCLENGTH (RADIANS) ALONG PHI - AXIS
C     IND        = INTEGER DESIGNATION OF PATCH AREA ( 1, 2, 3, OR 4 )
C     IGLOB      = GLOBAL DESIGNATION OF ELEMENT IN ATTACHMENT AREA
C     IFOR       = DETERMINES TYPE OF FORCE, 1 = RADIAL, 2 = SHEAR
C
C     THIS SUBROUTINE CREATES PR AND PPHI LOAD HARMONICS FOR SUCH LATERAL
C     LOADS IKP IS AN ERROR CHECK IF IKP .NE. 0 GO TO END OF PROGRAM
C
C     CREATED              :  05-OCT-83          BY: F.M.G. WONG
C     LATEST REVISION      :  30-NOV-83          BY: F.M.G. WONG
C
C     ******************************************************************
C
      REAL L,NU
      COMMON / CONSTANTS / A, E, L, NU, PI, T
      COMMON / CENTELM / CNTXEL(4,4,4), CNTPHIEL(4,4,4)
C
C
      IKP = 0
      IF ( (M .GT. 4) .OR. (N .GT. 4) ) IKP = 1
      IF ( (M .LT. 1) .OR. (N .LT. 1) ) IKP = 1
      IF ( IKP .EQ. 1 ) GO TO 98
      XM = FLOAT(M)
      XN = FLOAT(N)
C
      WRITE (1,100) IND, IFOR
      WRITE (6,100) IND, IFOR
100   FORMAT (1X,'-----> PERFORMING PRESS. DISTRIB.: ',I1,' FORCE: ',I1,/)
C
      CALL TIMESTAT (1)
C
      ISIZE = M*N*4
C
      DO 1 I = 1, N
C
      PSI = CNTPHIEL(I,J,IND) - PHIC
C
      DO 1 J = 1, M
C
C      :::::: NOW PHI AND X COORDINATES OF ELEMENT M,N ARE KNOWN
```

```
C
C
      THXT = THX/(2.*XN)
      THPHIT = THPHI/(2.*XM)
      AREA = THXT*A*THPHIT*4.
      FORCE = 1.0
C
      PR = FORCE/AREA
      PPHI = FORCE/AREA
C
      IF ( IND .NE. 1 )   CALL INITIALIZE
C
      XFOR = CNTXEL(I,J,IND)
      PHIFOR = CNTPHIEL(I,J,IND)
C
C     PRINT *
C     PRINT *,'XFOR = ',XFOR,' PHIFOR = ',PHIFOR,IND
C     PRINT *
C
      IF ( IFOR .EQ. 1 )  CALL PATCH (XFOR, PHIFOR, THXT, THPHIT, PR, 1)
      IF ( IFOR .EQ. 2 )  CALL PATCH (XFOR, PHIFOR, THXT, THPHIT, PPHI, 2)
C
      CALL TIMESTAT (2)
C
C     ::::::: CALCULATE STIFFNESS COEFFICIENTS FOR COLOCATION
C
      CALL CALCMAT (M, N, IND, ISIZE, IGLOB, IFOR)
C
      IGLOB = IGLOB + 1
C
1     CONTINUE
      CALL TIMESTAT (3)
C
C
      GO TO 99
C
98    WRITE(6,2) M, N
2     FORMAT(1X,'ERROR DETECTED IN ELEMENTS - NO LOAD CREATES,M,N = ',2I3)
      WRITE(1,2)
C
C
99    RETURN
      END
C
C
C     **********************************************************************
C
      SUBROUTINE INITIALIZE
C
C     INTIALIZES ALL PRESSURE ARRAYS TO ZERO
C
C     CREATED       :  19-OCT-83                BY : F.M.G. WONG
C      LAST REVISED :  24-OCT-83
C     **********************************************************************
C
      COMMON / LOADS / PRE(100,100), PRO(100,100), PXE(100,100),
```

```
     1                    PXO(100,100), PPHIE(100,100), PPHIO(100,100)
C
       DO 20 I = 1, 100
               DO 10 J = 1, 100
C
                       PRE(I,J)   = 0.
                       PRO(I,J)   = 0.
                       PXE(I,J)   = 0.
                       PXO(I,J)   = 0.
                       PPHIE(I,J) = 0.
                       PPHIO(I,J) = 0.
C
10             CONTINUE
       CALL TIMESTAT (2)
20     CONTINUE
C
C
       RETURN
       END
```

```
C      ***************************************************************************
C
       SUBROUTINE CALCMAT (M, N, IPATCH, ISIZE, IGLOB, IFOR)
C
C      CALCULATES THE STIFFNESS MATRIX FOR COLOCATION.
C
C      IPATCH  = INTEGER DESIGNATION OF PATCH AREA ON WHICH FORCE IS
C                  APPLIED ( 1, 2, 3, OR 4 )
C      ISIZE   = TOTAL NUMBER OF ELEMENTS IN ATTACHMENT AREA ( M*N*4 )
C      IGLOB   = CURRENT GLOBAL COUNTER TO WHICH THE UNIT PRESSURE IS
C                APPLIED
C
C      IF  M = N = 0, THEN COLOCATION IS PERFORMED FOR A CIRCLE
C
C      CREATED         :  05-OCT-83        BY : F.M.G. WONG
C      LATEST REVISION :  02-DEC-83
C
C      ***************************************************************************
C
       COMMON / CENTELM / CNTXEL(4,4,4), CNTPHIEL(4,4,4)
       COMMON / CIRCSD / CIRCSX(3,4), CIRCSP(3,4)
       COMMON / STIFMAT / STIFFA(8,8), STIFFB(32,32), STIFFC(72,72),
     1                    STIFFD(128,128), STIFFE(24,24)
C
       WRITE (1,800) IPATCH
       WRITE (6,800) IPATCH
800    FORMAT (1X,'-----> CALCULATING DISPLACEMENTS: ',I1,/)
C
       IF ( IFOR .EQ. 1 )   CALL DISPLACE (1)
       IF ( IFOR .EQ. 2 )   CALL DISPLACE (2)
C
       WRITE (1,900) IPATCH
       WRITE (6,900) IPATCH
900    FORMAT (1X,'-----> SUMMING FOURIER DISPLACEMENT: ',I1,/)
       CALL TIMESTAT (1)
C
       ILOC = 0
C
C      ::::::  COLOCATION FOR RECTANGULAR ATTACHEMENTS
C
       IF ( (M .NE. 0) .AND. (N .NE. 0) ) THEN
C
       DO 30 IND = 1, 4
               DO 20 I = 1, M
               DO 10 J = 1, N
C
               XPOS = CNTXEL(I,J,IND)
               PHIPOS = CNTPHIEL(I,J,IND)
C
               CALL SUMDISP ( XPOS, PHIPOS, DEFLR, 1 )
               CALL SUMDISP ( XPOS, PHIPOS, DEFLP, 2 )
C
               ILOC = ILOC + 1
C
               IF ( ISIZE .EQ. 4 )   THEN
                       STIFFA(ILOC,IGLOB) = DEFLR
                       STIFFA(ILOC+ISIZE,IGLOB) = DEFLP
```

```
                ENDIF
C
                IF ( ISIZE .EQ. 16 )  THEN
                        STIFFB(ILOC,IGLOB) = DEFLR
                        STIFFB(ILOC+ISIZE,IGLOB) = DEFLP
                ENDIF
C
                IF ( ISIZE .EQ. 36 )  THEN
                        STIFFC(ILOC,IGLOB) = DEFLR
                        STIFFC(ILOC+ISIZE,IGLOB) = DEFLP
                ENDIF
C
                IF ( ISIZE .EQ. 64 )  THEN
                        STIFFD(ILOC,IGLOB) = DEFLR
                        STIFFD(ILOC+ISIZE,IGLOB) = DEFLP
                ENDIF
C
10              CONTINUE
20              CONTINUE
30      CONTINUE
C
        ENDIF
C
C       ::::::  COLOCATION FOR CIRCULAR ATTACHMENTS
C
        IF ( (M ..EQ. 0) .AND. (N .EQ. 0) ) THEN
C
        DO 60 IND = 1, 4
                DO 50 K = 1, 3
C
                XPOS = CIRCSX(K,IND)
                PHIPOS = CIRCSP(K,IND)
C
                CALL SUMDISP ( XPOS, PHIPOS, DEFLR, 1 )
                CALL SUMDISP ( XPOS, PHIPOS, DEFLP, 2 )
C
                ILOC = ILOC + 1
C
                STIFFE(ILOC,IGLOB) = DEFLR
                STIFFE(ILOC+ISIZE,IGLOB) = DEFLP
C
C               PRINT *,XPOS, PHIPOS
                PRINT *,'DEFLR=',DEFLR,' DEFLP =',DEFLP
50              CONTINUE
60      CONTINUE
C
        ENDIF
C
C
        RETURN
        END
```

```
C     ************************************************************************
C
      SUBROUTINE SUMDISP (XST, PHIST, DEFL, INDEX)
C
C     SUMS UP THE FOURIER SERIES OF THE DISPLACEMENTS FOR A GIVEN
C     X AND PHI ON THE CYLINDRICAL SURFACE
C
C     XST   = X-COORDINATE OF DEFLECTION
C     PHIST = PHI-COORDINATE OF DEFLECTION
C     DEFL  = SUMMED UP DEFLECTION
C     INDEX = TYPE OF DEFLECTION ( RADIAL, SHEAR, OR AXIAL )
C
C     CREATED         :  05-OCT-83        BY : F.M.G. WONG
C     LATEST REVISION :  17-OCT-83        BY : F.M.G. WONG
C
C     ************************************************************************
C
C
      REAL L, NU
      COMMON / CONSTANTS / A, E, L, NU, PI, T
      COMMON / INDEXES / M, N
      COMMON / DISPL / UE(100,100), UO(100,100), VE(100,100), VO(100,100),
     1                 WE(100,100), WO(100,100)
C
C
C
C     CALL TIMESTAT (1)
C
      XCO=XST
      PHICO=PHIST
C
C
      XINT=0.
      DO 6 MT=0,M-1
              SPHI=SIN(FLOAT(MT)*PHICO)
              CPHI=COS(FLOAT(MT)*PHICO)
              DO 6 NT=1,N
                      IF ( INDEX .EQ. 1 ) XINC=(WE(MT+1,NT)*CPHI+WO(MT+1,NT)
     1                *SPHI)*SIN(FLOAT(NT)*PI*XCO/L)
                      IF ( INDEX .EQ. 2 ) XINC=(VE(MT+1,NT)*SPHI+VO(MT+1,NT)
     1                *CPHI)*SIN(FLOAT(NT)*PI*XCO/L)
                      IF ( INDEX .EQ. 3 ) XINC=(UE(MT+1,NT)*CPHI+UO(MT+1,NT)
     1                *SPHI)*COS(FLOAT(NT)*PI*XCO/L)
                      XINT=XINT+XINC
6             CONTINUE
      XINT=XINT*.5
C
      DEFL = XINT
C
C     CALL TIMESTAT (2)
C
C     CALL TIMESTAT (3)
C
      RETURN
      END
```

```
C       ***********************************************************************
C
        SUBROUTINE SOLVE (A, X, B, N, DET)
C
C       SOLVES AN N X N MATRIX EQUATION THROUGH GAUSS-JORDAN ELIMINATION
C
C       A   =  N X N COEFFICIENT MATRIX
C       X   =  1 X N SOLUTION COLUMN MATRIX
C       B   =  1 X N RIGHT HAND SIDE MATRIX
C       N   =  DIMENSION OF MATRICES
C       DET =  DETERMINANT OF 'A' MATRIX
C
C       BY:  W. CRAFT
C       CONVERTED TO VAX 11/780 BY:  F.M.G. WONG    DATE: 10-AUG-83
C
C       ***********************************************************************
C
C       DOUBLE PRECISION A, X, B, Y, DET, TEMP, D, D1, SIG
        DIMENSION B(N), X(N), A(N,N), K(100), Y(100)
C
C       PRINT *
C       PRINT *, ' N = ',N
C       PRINT *
C       DO 300 JA = 1, N
C               WRITE (1,200) ( A(JA,L), L=1,N )
C               WRITE (6,200) ( A(JA,L), L=1,N )
C200            FORMAT (<N*2>(E12.4,1X))
C300    CONTINUE
C
        DO 16 I = 1, N
C
16      K(I) = I
        N1 = N - 1
C       DET = 1.
        SIG = 1.
C
        DO 8 L = 1, N1
C
C       ::::::: SEARCH FOR LARGEST ELEMENT
C
        D = 0.
C
        DO 1 L1 = L, N
C
        DO 1 L2 = L, N
C
        IF ( ABS(A(L1,L2)) - ABS(D) ) 1, 1, 1550
1550    D = A(L1,L2)
        ID = L1
        JD = L2
1       CONTINUE
        IF (D) 2, 99, 2
C
C       ::::::: INTERCHANGE ROWS AND COLUMNS TO PUT LARGEST
C       ::::::: ELEMENT ON DIAGONAL
C
2       IF ( JD .EQ. L ) GO TO 14
```

```
      SIG = -SIG
      IEMP = K(L)
      K(L) = K(JD)
      K(JD) = IEMP
C
      DO 4 I = 1, N
C
      TEMP = A(I,L)
      A(I,L) = A(I,JD)
4     A(I,JD) = TEMP
14    IF ( ID .EQ. L ) GO TO 15
      SIG = -SIG
C
      DO 3 J = L, N
C
      TEMP = A(L,J)
      A(L,J) = A(ID,J)
3     A(ID,J) = TEMP
      TEMP = B(L)
      B(L) = B(ID)
      B(ID) = TEMP
15    B(L) = B(L)/D
C
C     ::::::: ELIMINATE IN COLUMN UNDER LARGEST ELEMENT
C
      L1 = L + 1
C
      DO 5 J = L1, N
C
5     A(L,J) = A(L,J)/D
      DET = DET*D
C
      DO 7 I = L1, N
C
      IF ( A(I,L) ) 1515, 7, 1515
1515  D1 = A(I,L)
C
      DO 6 J = L1, N
C
6     A(I,J) = A(I,J) - D1*A(L,J)
      B(I) = B(I) - D1*B(L)
7     CONTINUE
8     CONTINUE
      IF ( A(N,N) ) 9, 99, 9
C
C     ::::::: BACK SUBSTITUTE TO SOLVE
C
9     Y(N) = B(N)/A(N,N)
      DET = DET*A(N,N)*SIG
C
      DO 11 L = 1, N1
C
      LL = N - L + 1
      D1 = B(LL-1)
C
      DO 10 J = LL, N
```

```
C
10      D1 = D1 - A(LL-1,J)*Y(J)
11      Y(LL-1) = D1
C
C       ::::::: RE-ORDER ANSWER
C
        DO 12 I =1, N
C
        J = K(I)
12      X(J) = Y(I)
        GO TO 13
C
99      WRITE (6,100)
        WRITE (1,100)
100     FORMAT (2X,'MATRIX IS SINGULAR NO SOLUTION GIVEN')
        DET = 0.
C
13      RETURN
        END
C
```

```
C       ***********************************************************************
C
      SUBROUTINE CIRCOLSPL (XC, PHIC, R1, R2)
C
C       SPLITS CIRCULAR ATTACHMENT INTO FOUR QUADRANTS THROUGH FOUR CALLS
C       TO "CIRCPART." THEN FOUR CALLS TO "CIRCOLPRS" ARE USED TO INVOKE
C       THE METHOD OF COLOCATION FOR A CIRCULAR ATTACHMENT.
C
C       XC   = X-COORDINATE OF ATTACHMENT CENTER
C       PHIC = PHI-COORDINATE OF ATTACHMENT CENTER
C       R1   = INNER RADIUS OF ATTACHMENT = 0.0
C       R2   = OUTER RADIUS OF ATTACHMENT
C
C       CREATED      : 02-DEC-83                         BY: F.M.G. WONG
C       LAST REVISED : 03-DEC-83
C
C       ***********************************************************************
C
      REAL L, NU
      COMMON / CONSTANTS / A, E, L, NU, PI, T
C
      RADIAN = PI/180.
      ANG1 = 45.*RADIAN
      ANG2 = 135.*RADIAN
      ANG3 = 225.*RADIAN
      ANG4 = 315.*RADIAN
      ISIZE = 12
C
      CALL CIRCPART (XC, PHIC, R1, R2, ANG1, 1.0, 1.0, 1)
      CALL CIRCPART (XC, PHIC, R1, R2, ANG2, 1.0, -1.0, 2)
      CALL CIRCPART (XC, PHIC, R1, R2, ANG3, -1.0, -1.0, 3)
      CALL CIRCPART (XC, PHIC, R1, R2, ANG4, -1.0, 1.0, 4)
C
      IGLOB = 1
      IFOR = 1
C
      CALL CIRCOLPRS (XC, PHIC, R1, R2, ANG1, 1.0, 1.0, 1, IGLOB, IFOR)
      CALL CIRCOLPRS (XC, PHIC, R1, R2, ANG2, 1.0, -1.0, 2, IGLOB, IFOR)
      CALL CIRCOLPRS (XC, PHIC, R1, R2, ANG3, -1.0, -1.0, 3, IGLOB, IFOR)
      CALL CIRCOLPRS (XC, PHIC, R1, R2, ANG4, -1.0, 1.0, 4, IGLOB, IFOR)
C
      IGLOB = ISIZE + 1
      IFOR = 2
      CALL INITIALIZE
C
      CALL CIRCOLPRS (XC, PHIC, R1, R2, ANG1, 1.0, 1.0, 1, IGLOB, IFOR)
      CALL CIRCOLPRS (XC, PHIC, R1, R2, ANG2, 1.0, -1.0, 2, IGLOB, IFOR)
      CALL CIRCOLPRS (XC, PHIC, R1, R2, ANG3, -1.0, -1.0, 3, IGLOB, IFOR)
      CALL CIRCOLPRS (XC, PHIC, R1, R2, ANG4, -1.0, 1.0, 4, IGLOB, IFOR)
C
      RETURN
      END
```

```
C       ******************************************************************
C
        SUBROUTINE CIRCOLPRS (XC, PHIC, R1, R2, ANG, XFACT, PFACT, IND, IGLOB,
     1                        IFOR)
C
C       APPLIES PRESSURE LOADINGS TO CIRCULAR ATTACHMENT
C
C       R1    = INNER RADIUS OF ATTACHMENT
C       R2    = OUTER RADIUS OF ATTACHMENT
C       ANG   = ANGLE DESIGNATING THE QUADRANT
C       XFACT = DETERMINES SIGN OF X-DIRECTION
C       PFACT = DETERMINES SIGN OF PHI-DIRECTION
C       IND   = INTEGER DESIGNATING QUADRANT
C       IGLOB = GLOBAL DESIGNATION OF ELEMENT IN ATTACHMENT AREA
C       IFOR  = DETERMINES TYPE OF FORCE: 1 = RADIAL, 2 = SHEAR
C
C       CREATED      : 02-DEC-83                         BY: F.M.G. WONG
C       LAST REVISED : 03-DEC-83
C
C       ******************************************************************
C
        REAL L, NU
        DIMENSION THX(3,4), THP(3,4), TRIXA(4,4), TRIXB(4,4), TRIPA(4,4),
     1            TRIPB(4,4), KORIENT(4)
        COMMON / CONSTANTS / A, E, L, NU, PI, T
        COMMON / CIRCSD / CIRCSX(3,4), CIRCSP(3,4)
C
        DATA KORIENT / 3, 1, 2, 4 /
C
        ABTRIG = COS(45.*PI/180.)
        PR1 = ASIN(R1/A)
        PR2 = ASIN(R2/A)
        FORCE = 1.0
        PRINT *,' WORKING ON QUADRANT :',IND
C
C       ::::::: LOADINGS FOR SOLID CIRCULAR ATTACHMENTS
C
        THX(1,IND) = 0.5*R2*ABTRIG
        THP(1,IND) = 0.5*PR2*ABTRIG
        AREA = R2*PR2*ABTRIG*ABTRIG
        PRS = FORCE/AREA
C
        CIRX = CIRCSX(1,IND)
        CIRPHI = CIRCSP(1,IND)
C
        IF ( IND .NE. 1 )   CALL INITIALIZE
C
        PRINT *
        PRINT *, ' RECTANGLE -- ',IND,' PRS = ',PRS
        CALL PATCH (CIRX, CIRPHI, THX(1,IND), THP(1,IND), PRS, IFOR)
        CALL CALCMAT (0, 0, IND, 12, IGLOB, IFOR)
C
        IGLOB = IGLOB + 1
C
        TRIXA(1,IND) = XC + XFACT*R2
        TRIXB(1,IND) = XC + R2*SIN(ANG)
        TRIPA(1,IND) = PHIC
```

```
      TRIPB(1,IND) = PHIC + PR2*COS(ANG)
C
      TRIXA(2,IND) = XC + R2*SIN(ANG)
      TRIXB(2,IND) = XC
      TRIPA(2,IND) = PHIC + PR2*COS(ANG)
      TRIPB(2,IND) = PHIC + PFACT*PR2
C
C     PRINT *,'IND =',IND
      IF ( (IND .EQ. 2) .OR. (IND .EQ. 3) ) THEN
              DO 55 J = 1, 4
                      TEMPA = TRIXA(J,IND)
                      TEMPB = TRIXB(J,IND)
                      TRIXB(J,IND) = TEMPA
                      TRIXA(J,IND) = TEMPB
                      TEMPC = TRIPA(J,IND)
                      TEMPD = TRIPB(J,IND)
                      TRIPA(J,IND) = TEMPD
                      TRIPB(J,IND) = TEMPC
55            CONTINUE
      ENDIF
C
      DO 60 K = 1, 2
C
      AREA = 0.5*(ABS( TRIXA(K,IND) - TRIXB(K,IND) )*ABS( TRIPA(K,IND) -
     1 TRIPB(K,IND) ))
      PRS = FORCE/AREA
C
      CALL INITIALIZE
C
      PRINT *,'TRIANGLE ---',IND
      CALL TRIPATCH (TRIXA(K,IND),TRIXB(K,IND),TRIPA(K,IND),TRIPB(K,IND),
     1 PRS,KORIENT(IND),IFOR)
      CALL CALCMAT (0, 0, IND, 12, IGLOB, IFOR)
C
      IGLOB = IGLOB + 1
C
60    CONTINUE
C
C
      RETURN
      END
```

APPENDIX E

REFERENCES

1. Wichman, K.R., Hopper, A.G., Mershon, J.L., "Local Stresses In Spherical And Cylindrical Shells Due To External Loading," Welding Research Council Bulletin 107, August 1965.

2. Bijlaard, P.P., "Stresses From Local Loadings In Cylindrical Pressure Vessels," Transactions A.S.M.E., 77, No. 6, 1954.

3. Bijlaard, P.P., "Stresses From Radial Loads And External Moments In Cylindrical Pressure Vessels," A.S.M.E., December 1955.

4. Ugural, A.C., Stresses In Plates And Shells, New York, McGraw-Hill Book Co., 1981.

5. Billimoria, H.D. and Tam, K.K., "Experimental Investigation of Stiffness Coefficients and Allowable Loads for a Nozzle in a Flat Bottom Storage Tank," A.S.M.E., April 1980.

6. Cranch, E.T., "An Experimental Investigation of Stresses in the Neighborhood of Attachments to a Cylindrical Shell," Welding Research Council Bulletin 60, May 1960.