Université d'Ottawa · University of Ottawa

ANALYSIS OF TOROIDAL SHELLS USING THE SEMI-ANALYTICAL DQM

by

Wen Jiang

A thesis submitted to

the Faculty of Graduate Studies and Research

as partial fulfillment of the requirements

for the degree of

M. A. Sc. in Mechanical Engineering

UNIVERSITY OF OTTAWA

0-612-72771-8

Abstract

The present thesis consists of two main parts. The first part is concerned with the vibration and statics of transversely isotropic thick-walled toroidal shells. The second part is concerned with the vibration and statics of orthotropic thin-walled toroidal shells. In the first part a solution based on the linear three-dimensional theory of elasticity is developed for vibration and static problems of toroidal shells. The theory is developed for transversely isotropic toroids of arbitrary but uniform thickness. In the semi-analytical method that is adopted Fourier series are written in the circumferential direction, forming a set of two-dimensional problems. Finally results are determined for local surface loading problems. In the second part a solution based on the linear elastic Sanders-Budiansky shell equations is developed. The vibration and static characteristics of orthotropic toroidal shells of variable thickness are considered. A semi-analytical method in which Fourier series are written in the circumferential direction is adopted, forming a set of one-dimensional problems. A novelty in the solution concerns the use of power series as trial functions in a domain exhibiting cyclic periodicity. Results are determined in the second part for two separate applications. The problems in both parts of the work are solved using the differential quadrature method. A commercial finite element program is used to determine alternative solutions. The results from these two methods are compared, and conclusions are drawn.

Acknowledgement

I would like to express my gratitude to my supervisor, Dr. D. Redekop, for his patient guidance, continuous encouragement and valuable suggestions throughout the research.

I would like to thank the members of staff of the Department of Mechanical Engineering who made contributions towards my studies and research.

Special thanks go to my parents for their moral and financial support and also go to my grandparents, uncle, and aunt for their encouragement during my study.

The financial support provided in part by the Natural Sciences and Engineering Research Council of Canada is gratefully acknowledged.

Contents

List of Tables

List of Figures

Nomenclature

a, b	inside and outside radius for thick toroidal shell
A_1, A_2	Lame parameters
E, E'	Young's modulus in the in-plane and transverse (θ) directions for transversely isotropic thick-walled toroidal shell
E_1, E_2	Young's modulus in the meridional (ϕ) and circumferential (θ) directions for orthotropic thin-walled toroidal shell
f_c	volume faction of cord
G, G'	shear moduli in the meridional and circumferential directions for isotropic transversely thick-walled toroidal shell
G_{12}, G_{13}, G_{23}	shear moduli for orthotropic thin-walled toroidal shell
$h(\phi)$	thickness function for orthotropic thin-walled toroidal shell
$\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$	Cartesian unit vector
L_{ij}	differential operator in governing equations
$\hat{L}_{ij}$	differential operator in boundary equations
$M, N, N1$	truncation integers
$M_\phi, M_\eta, M_{\phi\eta}$	bending stress resultants
$N_\phi, N_\eta, N_{\phi\eta}$	membrane stress resultants
p_i	magnitude of normal pressure subjected on the thick-walled toroidal shell
p_r	magnitude of normal pressure subjected on the thin-walled toroidal shell
r	cross-section radius of orthotropic thin-walled toroidal shell

$\bar{r}$	cross-section mean radius of transversely isotropic thick-walled toroidal shell
$\boldsymbol{r}$	radius vector
R	toroidal bend radius
R_1, R_2	radii of curvature
t	mean thickness for orthotropic thin-walled toroidal shell
$\hat{t}$	time
u, v, w	displacement components
x, y, z	Cartesian coordinates
α	a half angular width of band load
β	frequency parameter, defined as $\beta^2 = (\hat{\rho}\omega^2\bar{r}^2)/E$
γ	radius ratio, defined as r/R
$\varepsilon_r, \varepsilon_\phi, \varepsilon_\theta$ $\varepsilon_{r\phi}, \varepsilon_{r\theta}, \varepsilon_{\phi\theta}$	strain components of transversely isotropic thick-walled toroidal shell
$\varepsilon_\phi, \varepsilon_\eta, \varepsilon_{\phi\eta}$	strain components of orthotropic thin-walled toroidal shell
η	toroidal circumferential coordinate, defined as ϕ / γ
θ	toroidal circumferential coordinate
$\kappa_\phi, \kappa_\eta, \kappa_{\phi\eta}$	curvature components
ν, ν'	Poisson's ratios of transversely isotropic thick-walled toroidal shell
ν_{12}, ν_{21}	Poisson's ratios of orthotropic thin-walled toroidal shell
$\hat{\rho}$	mass density

$\sigma_r, \sigma_\phi, \sigma_\theta$
$\sigma_{r\phi}, \sigma_{r\theta}, \sigma_{\phi\theta}$ stress components of transversely isotropic thick-walled toroidal shell

$\sigma_\phi, \sigma_\eta, \sigma_{\phi\eta}$ stress components of orthotropic thin-walled toroidal shell

ϕ toroidal meridional coordinate

Chapter 1
INTRODUCTION

Toroidal shells are frequently used in many areas of engineering, finding such applications as a basic structural element in Tokamak-type fusion reactor, vehicle and aircraft tires, rotating space station, and buoyancy units.

1.1 Thin-Walled Toroidal Shells

When the thickness of a shell lies in the range thickness/radius<1/10, the shell should be considered as a thin shell. The approximate two-dimensional theories based on Kirchhoff-Love hypothesis of non-deformable normals are most often adapted to thin-walled shells. For these shells, there is one mathematical variable for an axisymmetric model, and there are two variables for a non-axisymmetric model. Two approximate methods, the membrane approximation and the bending approximation, are used in thin shell analysis. The membrane approximation is the simplest of these approximations, in which it is assumed that the bending stiffness can be neglected. Thus all the bending stress resultants and shear resultants are zero. The membrane approximation is also called the extensional approximation. Correspondingly, the bending approximation is also called the inextensional approximation. In this assumption both membrane and bending stress resultants are considered. According to above-mentioned approximations, there is an assumption of small deflections and an elastic constitutive relationship.

1.2 Thick-Walled Toroidal Shells

The thickness of toroidal shells in some applications is not small enough to permit the adoption of thin shell theory. In such cases, the three-dimensional theory of elasticity is required. The number of variables in the general problems of thick-walled toroidal shell increases to two for an axisymmetric model, and three for a non-axisymmetic model. Work on the stress-based toroidal elasticity (TE) method was initiated by Göhner [1930] and continued by Lang [1989]. The load cases covered were in-plane loads, out-of-plane loads and seismic forces. The displacement-based toroidal elasticity method was developed by Redekop [1992]. The latter theory allows for a comprehensive study of thick-walled toroid including displacements and stresses.

1.3 Toroidal Shells with Variable Thickness and Non-Isotropic Material

Considerable work has been done on the analysis of isotropic toroidal shells with uniform thickness. Recently, a number of studies on variable thickness toroidal shells have appeared, dealing with applications such as tires, modern pressure vessels and thin-walled piping elbows which differ from the perfect toroidal shape as a result of manufacturing processes. In these studies the effects of geometric irregularities on the design analysis of toroidal shells were considered. A better understanding and qualification of the effects of geometric irregularities would enable the designer to select the most economical manufacturing process consistent with acceptable stress behavior.

With the advent of advanced fiber reinforced composite materials having many attractive properties like high strength to weight and stiffness to weight ratios, good corrosion and heat resistance and excellent fatigue strength, shells made of these materials have been extensively used in civil, chemical, mechanical and marine industries.

1.4 Approximate Solutions of Toroidal Shells

The majority of the studies on toroidal shells have relied on analytical or numerical solutions. Approximate solutions generally require presentation of functional values at certain discrete points of the domain. The corresponding approximate solution of numerical discretization is called the numerical solution. Currently, numerical methods, such as finite difference (FD), finite element (FE), and finite volume (FV) find frequent use. They are under the category of low order methods. A large number of mesh points are needed in these methods. The FD method is based on the polynomial approximation or the Taylor series expansion and the FE method is based on the variation principle or the principle of weighted residuals, while the FV method applies the physical conservation law directly to a finite cell.

To these methods has been added the method of differential quadrature (DQ), where a partial derivative of a function with respect to a coordinate direction is expressed as a linear weighted sum of all the functional values at all mesh points along that direction. DQ method (DQM) is a high order method which permits the use of coarser mesh spacing than the other methods offering savings in computation to the user.

1.5 Objective of This Thesis

The objective of this thesis is to consider the problems of vibration and statics of transversely isotropic thick-walled toroidal shells and orthotropic thin-walled toroidal shells. Three-dimensional elasticity theory and the Sanders-Budiansky thin shell theory, respectively, are used to solve the problems. Results in each case are found using the DQM and the finite element method (FEM). The main part of this thesis is divided into four parts.

In chapter two, a literature survey is given for the DQM and semi-analytical method, toroidal elasticity theory, toroidal shell theory, and the vibration of toroidal shell.

In chapter three, the governing equations of the displacement-based toroidal elasticity theory are set up in the toroidal coordinate system for transversely isotropic material. Fourier series expansions are written in the circumferential direction leading to a series of two-dimensional problems for the individual harmonics. The theory developed is used to solve thick shell problems in which the thickness is uniform. Finally results are determined for natural frequency and local surface loading problems by the DQM and FEM.

In chapter four, the governing equations of the Sanders-Budiansky shell theory [Sanders 1995, Budiansky 1968] are developed in the toroidal coordinate system for a shell with variable thickness and consisting of an orthotropic material. Fourier series are written in the circumferential direction leading to a series of one-dimensional problems for the individual harmonics. The theory developed is used to solve two cases for vibration and statics. In the first case a complete isotropic toroidal shell is considered, and the shell wall is thinner at the extrados. Final results are determined for natural frequency and internal pressure cases by DQM and FEM. In the second case an orthotropic toroidal shell incomplete in the meridional direction is considered, and the shell wall is thicker at the extrados. Final results are determined for natural frequency and pad load problems by DQM and FEM.

In chapter five, conclusions are drawn and suggestions for future study are made.

Chapter 2

LITERATURE SURVEY

2.1 Differential Quadrature Method and Semi-Analytical Method

Bellman [Bellman and Casti 1972] introduced the method of differential quadrature (DQM), where a partial derivatives of a function with respect to a coordinate direction is expressed as a linear weighted sum of all the functional values at all mesh points along that direction. The DQM is gradually emerging as a distinct numerical solution technique for the initial- and/or boundary-value problems of physical and engineering science.

Many researchers have used Bellman's approach to compute the weighting coefficients, such as Hu and Hu [1974], Mingle [1977], Wang [1982], Naadimuthu et al. [1984], Bert et al. [1988,1989], Jang et al. [1989] [Shu 2000]. Shu and Richard [1992] developed a global method called the GDQ method. This new method has overcome the difficulty of DQM in obtaining the weighting coefficients for the first-order derivative discretization with arbitrary distribution of grid points. In the GDQ method, the weighting coefficients of the first-order derivative are given by a simple algebraic formulation, and the weighting coefficients of the second- and higher-order derivative are determined by a recurrence relationship. Shu and Xue [1997] demonstrated that the weighting coefficients could be calculated using explicit formulations by following the Fourier series expansion and the same concept of GDQ method. So the explicit formulations for computing the weighting coefficients in the harmonic differential quadrature (HDQ) have been developed. This method is useful especially for some problems with periodic behavior. A monograph about the DQM was presented by Shu in 2000 which introduced the development and applications of DQM [Shu 2000].

The application of the DQM to the area of structural mechanics was introduced through the vibration analysis [Bert et al. 1988] and static analysis [Bert et al. 1989] of basic structural components such as beams and plates. Then the DQM was applied to the solution of Navier-Stokes equations [Shu and Richard 1992]. More recently, the application of the DQM has been introduced to lubrication problems [Malik and Bert 1994]. Redekop and Xu [1999] represent the first application of this method to problems in shell theory with variable coefficients in the governing equations. In their work the free vibration characteristics of linear elastic toroidal shell panels are determined.

A semi-analytical approach has been shown to be a simple, effective and efficient one in dealing with the problem of a shell of revolution. In this approach, the displacements and rotations of the shell are represented by Fourier series in the circumferential coordinate direction. In 1990, Kim et al. modeled the space shuttle nose-gear tire with semi-analytical finite elements [Kim et al. 1990]. Shell variables are represented in semi-analytical finite elements by Fourier series in the circumferential direction and piecewise polynomials in the meridional direction. In 1996, Bert et al. did the free vibration analysis of thin cylindrical shell by the semi-analytical DQM [Bert and Malik 1996a]. The displacements are represented by trigonometric sine and cosine function of the circumferential coordinate to reduce the size of eigenvalue matrices.

Noor and Peters [1987] carried out the analysis of laminated anisotropic shells of revolution. The Sanders-Budiansky shell theory was developed for the governing equations. Semi-analytical finite elements were used in which the shell variables were represented by Fourier series in the circumferential direction and piecewise polynomials in the meridional direction. A computational procedure was presented for assessing the sensitivity of the shell response to anisotropic material coefficients.

The modeling and analysis of the space shuttle nose-gear tire was developed by Kim et al. [1990] in a similar approach to that of Noor and Peters [1987]. The cord-rubber composite which was treated as a laminated material was used in the tire model. Numerical results were presented for an inflated Space Shuttle tire and the case of localizing loading.

2.2 Toroidal Elasticity Theory

Göhner [1930] was the first researcher to work on the three-dimensional (3D) theory the so-called toroidal elasticity theory (TE). He chose a special co-ordinate system (Fig. 2.1) to investigate the problem of a curved solid circular ring sector subjected to pure twisting and bending moments. His theory was specialized to axisymmetric cases and his co-ordinate system works for the boundary conditions of a solid circular cross-section, but is not good for the boundary conditions of an annular cross-section. Kornecki [1963] and McGill and Lenzen [1967] resumed the work of Göhner in the 1960s. Kornecki used the method of successive approximation to solve the governing equations. McGill and Lenzen applied the theory to the circumferential axisymmetric free vibration of a thick hollow tori by using the finite difference method. Both of these authors solved only the axisymmetric problem.

Lang [1989] developed TE theory in the non-classical toroidal coordinate system (Fig. 2.2). In this approach the boundary conditions of a hollow elbow can easily be represented. He cast the three equilibrium equations and the six compatibility equations into this suitable set of curvilinear coordinates. This theory can be applied both to axisymmetric problems or non-axisymmetric problems. Redekop [1992] extended the work of Lang. He developed the governing Navier equations based on the displacement components variables in this coordinate system. This displacement based TE theory has the advantage of yielding immediately the displacements as well as the stresses. Zhu [1994] used a displacement-based approach to TE to solve three problems. The first one concerns an elbow subjected to an out-of-plane couple at the free end. The second problem concerns an axisymmetric sectorial toroidal ring subjected to loads on its radial surfaces. The third one concerns a toroidal shell subjected to band loads of normal pressure.

The increasing demands of high-performance composite materials in aircraft, aerospace vehicles, large space structure, and shipbuilding have simulated interest in the development of 3D elasticity theory of anisotropic shells of revolution. Grigorenko et al. [1995] used the 3D elasticity theory to investigate the stress-strain state of hollow thick-walled noncircular cylinders for certain types of orthotropy under a local load. Also in their paper noncircular cross-sections were considered. Noor and Burton [1990] have systematically reviewed the available approaches and solutions based on 3D elasticity theory for the analysis of multilayered composite material shells. Chandrashekhara and Rao [1996] presented an approximate 3D elasticity solution for an infinite, thick, orthotropic as well as a laminated circular cylindrical shell of revolution subjected to distributed-pinch loads. Grigorenko et al. [1998] proposed a numerical approach to the dynamic problem with nonaxisymmetric loading based on the model obtained in the 3D linear elasticity theory of an anisotropic body. Jiang and Redekop [2002b] developed a solution based on the linear 3D theory of elasticity for vibration and statics problems of transversely isotropic toroidal shell.

2.3 Toroidal Shell Theory

The various shell theories that are in use at present all stem from the equations of Love developed towards the end of the 19^{th} century. The most popular shell theories are those of Sanders-Budiansky, Flügge, Reissner, Donnell, Reissner-Mindlin, Timoshenko, and Berry and Naghdi [Leissa 1973]. In the past two decades a substantial capability has been developed for the numerical analysis of arbitrary shells of revolution. Most of this capability is for orthotropic and transversely isotropic shells for which the principle material directions coincide with the coordinates lines. Recently, interest in the use of composite shells of revolution has been expanded in aircraft, shipbuilding, and other industries.

Studies of toroidal shells have dealt with the static, dynamic and buckling analysis of partial and complete toroidal shells. Sanders and Liepins [1963] considered the toroidal membrane under uniform normal pressure. Their approach extended from both the nonlinear membrane theory and the bending theory [Rossettos and Sanders 1965]. Sobel and Flügge [1967] derived the buckling and free vibration equations of prestressed thin shells of revolution and applied them to toroidal shells. Thompson and Spence [1983] have presented a solution employing the theorem of minimum potential energy, with suitable kinematically admissible displacements. Whatham [1986] has presented the linear shell equations for analyzing pipe bends or discontinuous toroidal shells. Xia and Zhang [1987] have presented a general solution of toroidal shells subjected to symmetric or nonsymmetric loading based on Novozhilov's complex force equations of thin shell theory. Zhang and Redekop [1991] have provided results for surface loading of a thin-walled toroidal shell using the linear Sanders theory. Leung and Kwok [1994] have studied the free vibration of a toroidal shell using the Donnell-Mushtari-Vlasov (DMV) thin shell theory, Fourier series and Galerkin's method.

The early work on shells dealt mainly with the isotropic, transversely isotropic and orthotropic material. Recently, work has been concentrated on laminated composite materials. Noor and Peters [1987] presents an efficient computational procedure for the free vibration analysis of laminated anisotropic shells of revolution. The analytical formulation is based on a form of the Sanders-Budiansky shell theory in which the transversely shear deformation is not taken into account. Xi et al. [1999] investigated the effects of shear non-linearity on free vibration characteristics of laminated composite shells of revolution using a semi-analytical method based on the Reissner-Mindlin shell theory.

2.4 Vibration of Toroidal Shells

The literature on toroidal shell vibration is relatively sparse, and in the main monograph by Leissa [1973] there is mention only of a few studies. In recent years the study of this topic has intensified considerably.

McGill and Lenzen [1967] studied the circumferential axisymmetric free vibration of thick hollowed tori using the axisymmetric theory of elasticity. In their article, an expansion of the basic equation of elastic motion was made in toroidal coordinates. The effects of thickness and the radii ratio on the natural frequencies of axisymmetric free vibration of tori were considered.

The free vibrations of ring-stiffened toroidal shells were studied by Balderes and Anthony [1973] on the basis of the Love-Reissner shell theory. Their results indicate that the frequency increases with increasing h/R (thickness to radii ratio) and decreasing a/R (radii ratio) and the frequency of the stiffened shell remains the same or increases substantially compared with the frequency of the unstiffened shell.

Kosawada et al. [1986] analyzed the free vibration of thick toroidal shells with circular cross section by using an improved thick shell theory. The governing equations are derived from the stationary conditions of the Lagrangian of the toroidal shell. A power series expansion was used to solve the equations of motion and then the natural frequency and mode shapes were obtained.

Noor and Peters [1987] presented an efficient computational procedure for the free vibration analysis of laminated anisotropic shells of revolution. The governing equations are based on a form of Sanders-Budiansky shell theory including the effects of both the transverse shear deformation and the laminated anisotropic material response. The results of two free vibration problems are discussed — simply supported two-layered cylindrical shell and clamped two-layered anisotropic toroidal shell.

The free vibration of a toroidal shell was studied by Leung and Kwok [1994] using the Donnell-Mushtari-Vlasov (DMV) thin shell theory. Fourier series were written, and Galerkin's method was applied to solve the problem of the vibration of a *90-degree-bend* curved pipe. Results are compared to those obtained by finite element and other methods.

Huang et al. [1995] studied the free vibrations of a curved pipe. The governing equations are based on the linear elastic Sanders shell theory. Natural frequency and mode shapes of a 90° curved pipe were determined. The results are compared with previously published results found using the DMV theory. Frequencies found from the two methods showed good agreement.

Xi et al. [1999] investigated effects of shear non-linearity on free vibration of a laminated composite shell of revolution using a semi-analytical method based on the Reissner-Mindlin shell theory. The numerical results showed that the fundamental frequencies of composite shells of revolution might reduce significantly because of shear non-linearity.

Jiang and Redekop [2002c] studied the polar axisymmetric vibrations of isotropic toroidal shells with circular cross-section and uniform thickness using the 3D theory of elasticity. Numerical results were found using the new differential quadrature method (DQM). The solution was partially validated by comparing with results from the finite element method (FEM).

Chapter 3

VIBRATION AND STATICS OF TRANSVERSELY ISOTROPIC THICK-WALLED TOROIDAL SHELLS

3.1 Introduction

Toroidal shells are popular structural forms, finding such applications in engineering as reactor vessels, protective devices for nuclear fuel pellets, buoyancy units, and vehicle and aircraft tires. With the development of new materials there is a demand for new analytical tools to model toroidal shells made of anisotropic, orthotropic, and transversely isotropic materials [Naboulsi et al. 2000].

Considerable work has been done on the vibration and statics of thin-walled isotropic toroidal shells [Redekop and Xu 2000, Leung and Kwok 1994, Redekop 1994]. Studies have also been conducted for thin-walled laminated toroidal shells with application to vehicle and aircraft tires [Kim et al. 1990, Gall et al. 1995, Zhang et al. 1997, and Darnell et al.1997]. In some of these latter studies it has been pointed out that the shells approach a thick-walled status. There is thus a need for an analytical tool to cover thick-walled toroidal shells (hollow toroids), for tire as well as other applications. As well it is desirable to have an independent method available that can serve to validate the finite element method, which is the main tool available in the area at present.

In this chapter the three-dimensional theory of elasticity is developed to consider the vibration and statics of hollow transversely isotropic toroids with annular cross-sections of arbitrary but uniform thickness. Fourier series expansions are written in the circumferential direction leading to a set of two-dimensional problems for the individual harmonics. The problems are solved using the differential quadrature method (DQM) and the finite element method (FEM). Validation of the method is through the comparison of results with previously published work. Finally results are determined for local surface loading problems, and conclusions are drawn.

3.2 Toroidal Elasticity Theory

The hollow toroid (thick-walled toroidal shell) has a bend radius R, and an annular cross-section bounded by radii a and b (Fig. 3.1). A general point P in the shell is defined by circumferential and meridional angular coordinates θ and ϕ, and a radial coordinate r. The shell is complete and thus extends through 360° in the circumferential and meridional directions.

A derivation of the governing equations for linear three-dimensional theory of elasticity in toroidal coordinates has been given for isotropic materials by Redekop [1992]. The theory is extended here to cover the case for transversely isotropic materials. The equations of motion in this theory are given by [Redekop 1992, Grigorenko et al 1998]

$$\frac{\partial \sigma_r}{\partial r}+\frac{1}{r}\frac{\partial \sigma_{r\phi}}{\partial \phi}+\frac{1}{r}(\sigma_r-\sigma_\phi)+\frac{1}{\rho}[\frac{\partial \sigma_{r\theta}}{\partial \theta}+\cos\phi(\sigma_r-\sigma_\theta)-\sin\phi\sigma_{r\phi}]=\hat{\rho}\frac{\partial^2 u}{\partial \hat{t}^2}$$

$$\frac{\partial \sigma_{r\phi}}{\partial r}+\frac{1}{r}\frac{\partial \sigma_{\phi}}{\partial \phi}+\frac{2}{r}\sigma_{r\phi}+\frac{1}{\rho}[\frac{\partial \sigma_{\phi\theta}}{\partial \theta}+\cos\phi\sigma_{r\phi}-\sin\phi(\sigma_\phi-\sigma_\theta)]=\hat{\rho}\frac{\partial^2 v}{\partial \hat{t}^2} \tag{3.1}$$

$$\frac{\partial \sigma_{r\theta}}{\partial r}+\frac{1}{r}\frac{\partial \sigma_{\phi\theta}}{\partial \phi}+\frac{1}{r}\sigma_{r\theta}+\frac{1}{\rho}[\frac{\partial \sigma_{\theta}}{\partial \theta}+2\cos\phi\sigma_{r\theta}-2\sin\phi\sigma_{\phi\theta}]=\hat{\rho}\frac{\partial w^2}{\partial \hat{t}^2}$$

where $\rho = R + r\cos\phi$, $\hat{\rho}$ is the mass density and $\hat{t}$ is the time. The convention for the displacement and stress components present in this theory is given in Fig. 3.2. The kinematics relations for the case of small displacements are [Redekop 1992]

$$\varepsilon_r=\frac{\partial u}{\partial r}; \qquad \varepsilon_{r\phi}=\frac{1}{2}(\frac{1}{r}\frac{\partial u}{\partial \phi}+\frac{\partial v}{\partial r}-\frac{v}{r})$$

$$\varepsilon_\phi=\frac{u}{r}+\frac{1}{r}\frac{\partial v}{\partial \phi}; \qquad \varepsilon_{r\theta}=\frac{1}{2}(\frac{\partial w}{\partial r}+\frac{1}{\rho}\frac{\partial u}{\partial \theta}-\frac{w}{\rho}\cos\phi) \tag{3.2}$$

$$\varepsilon_\theta=\frac{1}{\rho}(u\cos\phi-v\sin\phi+\frac{\partial w}{\partial \theta}); \qquad \varepsilon_{\phi\theta}=\frac{1}{2}(\frac{1}{r}\frac{\partial w}{\partial \phi}+\frac{1}{\rho}\frac{\partial v}{\partial \theta}+\frac{w}{\rho}\sin\phi)$$

For transversely isotropic materials the constitutive equations are [Grigorenko et al 1998]

$$\begin{Bmatrix} \sigma_r \\ \sigma_\phi \\ \sigma_\theta \\ \sigma_{r\phi} \\ \sigma_{r\theta} \\ \sigma_{\phi\theta} \end{Bmatrix} = \begin{bmatrix} d_{11} & d_{12} & d_{13} & 0 & 0 & 0 \\ d_{21} & d_{22} & d_{23} & 0 & 0 & 0 \\ d_{31} & d_{32} & d_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & d_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & d_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & d_{66} \end{bmatrix} \begin{Bmatrix} \varepsilon_r \\ \varepsilon_\phi \\ \varepsilon_\theta \\ \varepsilon_{r\phi} \\ \varepsilon_{r\theta} \\ \varepsilon_{\phi\theta} \end{Bmatrix} \tag{3.3}$$

where the d_{ij} are given by

$$\begin{aligned} d_{11} &= d_{22} = \frac{E' - E(\nu')^2}{E(E')^2\Delta};\ d_{33} = \frac{1-\nu^2}{E^2\Delta};\ d_{12} = \frac{\nu E' + E(\nu')^2}{E(E')^2\Delta} \\ d_{13} &= \frac{\nu'(1+\nu)}{EE'\Delta};\ d_{23} = \frac{\nu'(1+\nu)}{EE'\Delta};\ d_{44} = 2G;\ d_{55} = d_{66} = 2G' \\ \Delta &= \frac{E'(1-\nu^2) - 2E(\nu')^2(1+\nu)}{(EE')^2};\ G = \frac{E}{2(1+\nu)} \end{aligned} \tag{3.4}$$

The engineering constants ν, ν', E, E', G, G' are respectively the Poisson ratios, Young's moduli, and shear moduli in the in-plane and transverse (θ) directions.

The solution is subjected to boundary conditions on the surfaces of the toroid. On the r = *cnst* surfaces the requirements are

$$\sigma_r = p_i(\theta, \phi); \quad \sigma_{r\theta} = \sigma_{r\phi} = 0 \tag{3.5}$$

where p_i is a specified pressure with i = 1 for the inside surface and i = 2 for the outside surface. For some problems symmetry exists about the planes $\phi = 0^o$, $\theta = 0^o$, and $\theta = 90^o$. This symmetry allows consideration of one-eight of the toroid only. In these cases the symmetry conditions

$$v = 0; \quad \sigma_{\phi r} = \sigma_{\phi\theta} = 0 \tag{3.6}$$

are satisfied on the plane $\phi = 0^o$, and the symmetry conditions

$$w = 0; \quad \sigma_{\theta r} = \sigma_{\phi\theta} = 0 \tag{3.7}$$

are satisfied on the planes $\theta = 0^o$, and $\theta = 90^o$.

For calculation purposes the three-dimensional problem is reduced to a set of two-dimensional ones. In this semi-analytical approach Fourier series are written in the circumferential direction, and then the various harmonics so defined are dealt with separately. Thus the displacement components and loading are expanded as

$$\begin{aligned}
u &= U_0 + \sum U_m \cos m\theta + \sum \overline{U}_m \sin m\theta \\
v &= V_0 + \sum V_m \cos m\theta + \sum \overline{V}_m \sin m\theta \\
w &= \sum W_m \sin m\theta + \sum \overline{W}_m \cos m\theta \\
p &= P_0 + \sum P_m \cos m\theta + \sum \overline{P}_m \sin m\theta
\end{aligned} \tag{3.8}$$

where $m = 1, \cdots, M$. Governing equations are obtained by combining Eqns. (3.1— 3.3) with (3.8). Two equations are obtained for the unknowns U_0, V_0

$$\begin{aligned}
&\left(v_1 \frac{\partial^2}{\partial r^2} + v_2 \frac{\partial}{\partial r} + v_3 \frac{\partial^2}{\partial \phi^2} + v_4 \frac{\partial}{\partial \phi} + v_5 \right) U_0 \\
&+ \left(v_6 \frac{\partial^2}{\partial r \partial \phi} + v_7 \frac{\partial}{\partial r} + v_8 \frac{\partial}{\partial \phi} + v_9 \right) V_0 = \hat{\rho} \frac{\partial^2 U_0}{\partial \hat{t}^2} \\
&\left(v_6 \frac{\partial^2}{\partial r \partial \phi} + v_{12} \frac{\partial}{\partial r} + v_{13} \frac{\partial}{\partial \phi} + v_{14} \right) U_0 \\
&+ \left(v_{15} \frac{\partial^2}{\partial r^2} + v_{16} \frac{\partial}{\partial r} + v_{17} \frac{\partial^2}{\partial \phi^2} + v_{18} \frac{\partial}{\partial \phi} + v_{19} \right) V_0 = \hat{\rho} \frac{\partial^2 V_0}{\partial \hat{t}^2}
\end{aligned} \tag{3.9}$$

three equations for each of the N sets of unknowns U_m, V_m, W_m of the ‘even’ harmonics

$$\left(\nu_1\frac{\partial^2}{\partial r^2}+\nu_2\frac{\partial}{\partial r}+\nu_3\frac{\partial^2}{\partial\phi^2}+\nu_4\frac{\partial}{\partial\phi}+\nu_5\right)U_m$$
$$+\left(\nu_6\frac{\partial^2}{\partial r\partial\phi}+\nu_7\frac{\partial}{\partial r}+\nu_8\frac{\partial}{\partial\phi}+\nu_9\right)V_m+\left(\nu_{10}+\nu_{11}\frac{\partial}{\partial r}\right)W_m=\hat{\rho}\frac{\partial^2 U_m}{\partial\hat{t}^2}$$
$$\left(\nu_6\frac{\partial^2}{\partial r\partial\phi}+\nu_{12}\frac{\partial}{\partial r}+\nu_{13}\frac{\partial}{\partial\phi}+\nu_{14}\right)U_m$$
$$+\left(\nu_{15}\frac{\partial^2}{\partial r^2}+\nu_{16}\frac{\partial}{\partial r}+\nu_{17}\frac{\partial^2}{\partial\phi^2}+\nu_{18}\frac{\partial}{\partial\phi}+\nu_{19}\right)V_m+\left(\nu_{20}\frac{\partial}{\partial\phi}+\nu_{21}\right)W_m=\hat{\rho}\frac{\partial^2 V_m}{\partial\hat{t}^2} \tag{3.10}$$
$$\left(\nu_{11}\frac{\partial}{\partial r}+\nu_{22}\right)U_m+\left(\nu_{23}\frac{\partial}{\partial\phi}+\nu_{21}\right)V_m+$$
$$\left(\nu_{24}\frac{\partial^2}{\partial r^2}+\nu_{25}\frac{\partial}{\partial r}+\nu_{26}\frac{\partial^2}{\partial\phi^2}+\nu_{27}\frac{\partial}{\partial\phi}+\nu_{28}\right)W_m=\hat{\rho}\frac{\partial^2 W_m}{\partial\hat{t}^2}$$

and three equations for each of the N sets of unknowns $\overline{U}_m$, $\overline{V}_m$, $\overline{W}_m$ of the 'odd' harmonics

$$\left(\nu_1\frac{\partial^2}{\partial r^2}+\nu_2\frac{\partial}{\partial r}+\nu_3\frac{\partial^2}{\partial\phi^2}+\nu_4\frac{\partial}{\partial\phi}+\nu_5\right)\overline{U}_m$$
$$+\left(\nu_6\frac{\partial^2}{\partial r\partial\phi}+\nu_7\frac{\partial}{\partial r}+\nu_8\frac{\partial}{\partial\phi}+\nu_9\right)\overline{V}_m+\left(\nu_{10}+\nu_{11}\frac{\partial}{\partial r}\right)\overline{W}_m=\hat{\rho}\frac{\partial^2 \overline{U}_m}{\partial\hat{t}^2}$$

$$\left(\nu_6\frac{\partial^2}{\partial r\partial\phi}+\nu_{12}\frac{\partial}{\partial r}+\nu_{13}\frac{\partial}{\partial\phi}+\nu_{14}\right)\overline{U}_m$$
$$+\left(\nu_{15}\frac{\partial^2}{\partial r^2}+\nu_{16}\frac{\partial}{\partial r}+\nu_{17}\frac{\partial^2}{\partial\phi^2}+\nu_{18}\frac{\partial}{\partial\phi}+\nu_{19}\right)\overline{V}_m+\left(-\nu_{20}\frac{\partial}{\partial\phi}-\nu_{21}\right)\overline{W}_m=\hat{\rho}\frac{\partial^2 \overline{V}_m}{\partial\hat{t}^2} \tag{3.11}$$

$$\left(-\nu_{11}\frac{\partial}{\partial r}+\nu_{22}\right)\overline{U}_m+\left(\nu_{23}\frac{\partial}{\partial\phi}-\nu_{21}\right)\overline{V}_m+$$
$$\left(\nu_{24}\frac{\partial^2}{\partial r^2}+\nu_{25}\frac{\partial}{\partial r}+\nu_{26}\frac{\partial^2}{\partial\phi^2}+\nu_{27}\frac{\partial}{\partial\phi}+\nu_{28}\right)\overline{W}_m=\hat{\rho}\frac{\partial^2 \overline{W}_m}{\partial\hat{t}^2}$$

The coefficients ν_1—ν_{28} are given in Appendix A.

The basic equations are used to solve alternatively the problems of statics and vibration for the shell. In solving the statics problem the $\hat{\rho}$ is taken as zero and the boundary term p_i represents a static pressure. For the vibration problem the $\hat{\rho}$ is non-zero, the p_i are zero, and the displacements are taken as harmonics in the natural frequency $\omega\hat{t}$.

3.3 Differential Quadrature Method

The DQM approach is used to calculate the solution to the governing equations. This method of numerical analysis was introduced to problems of shell vibration by [Shu 2000]. As a semi-analytical approach is used in the current study the two-dimensional DQM is required, with all unknown functions expressed in terms of the variables r and ϕ. A brief overview of the method is presented in the following for completeness.

The basis of the DQM is the meshing of the domain and the representation in the domain of the derivatives of a function f(x) by a weighted sum of trial function values at a set of sampling points, i.e.

$$\left.\frac{d^k f}{dx^k}\right|_{x=x_i} = \sum_{j=1}^{N} A_{ij}^{(k)} f(x_j) \tag{3.12}$$

Here the $f(x_j)$ are the function values at the sampling points j, the $A_{ij}^{(k)}$ are weighting coefficients for the *k-th* order derivative at the *i-th* sampling point in the domain, and N is the number of sampling points. The weighting coefficients are determined prior to the analysis on the basis of a chosen set of trial functions. Enforcement of the governing equations and boundary conditions at the sampling points together with the use of the relation (3.12) converts the problem of differential equations into one of linear simultaneous equations in terms of the unknown function values at the defined sampling points of the mesh. The resulting equations are solved using standard matrix packages.

For the present problem polynomial trial functions are selected for the radial direction, and the variable x represents r. The functions are taken as

$$f(r) = 1, r, r^2, \ldots, r^{N1-1} \tag{3.13}$$

For these functions explicit formulas for the weighting coefficients are available in the literature [Bert and Malik 1996b]. For the first order derivative the formulas are

$$A_{ij}^{(1)} = \frac{\pi(r_i)}{(r_i - r_j)\pi(r_j)}; \quad i, j = 1, 2, \ldots, N1; i \neq j$$

$$\pi(r_i) = \prod_{j=1}^{N1} (r_i - r_j); \quad i \neq j$$

while for the higher order derivatives the formulas are

$$\begin{aligned} A_{ij}^{(k)} &= k[A_{ii}^{(k-1)} A_{ij}^{(1)} - \frac{A_{ij}^{(k-1)}}{r_i - r_j}]; \; i, j = 1, 2, \ldots, N1; \; i \neq j; \; 2 \leq k \leq (N1 - 1) \\ A_{ij}^{(k)} &= A_{ii}^{(k)} = -\sum_{s=1}^{N1} A_{is}^{(k)}; \; i = 1, 2, \ldots, N1; \; i \neq s; \; 1 \leq k \leq (N1 - 1) \end{aligned} \tag{3.14}$$

For the sampling points in the radial direction of the mesh the Chebyshev-Gauss-Lobatto spacing (Shu 2000) is used. In this scheme the coordinates r_i are taken as $r_i \equiv x_i = (b - a)\, y_i$ where the y_i are given by

$$y_1 = 0;\ y_i = \frac{1 - \cos\dfrac{\pi(i-2)}{N1-3}}{2};\ 2 < i < (N1-1);\ y_{N1} = 1 \tag{3.15}$$

At each sampling point either the DQM analogue of a governing equation for the domain is represented, or a boundary equation.

Harmonic trial functions (Redekop and Xu 2000) are used in the meridional direction in this problem having cyclic periodicity. Continuity conditions across $\phi = 180, 360^o$ are then identically satisfied. The trial function are thus taken as

$$\begin{aligned} f(\phi) &= \cos[2(K-1)\pi\phi];\ K = 1,2,3,\ldots,N/2+1 \\ f(\phi) &= \sin[2(K-N/2-1)\pi\phi];\ K = N/2+2, N/2+3,\ldots,N \end{aligned} \tag{3.16}$$

where N is an even number. For equally spaced sampling points in the ϕ direction the weighting coefficients, labeled B_{ij} in the following, may be found explicitly from the inverse of the Vandermonde matrix (Shu 2000).

Use of the quadrature rules (3.12—3.16) for the derivatives in the governing equations (3.10) leads to the transformed DQM domain equations for the nth even harmonic set

$$\begin{aligned} &(v_1 \sum A_{hl}^{(2)} + v_2 \sum A_{hl}^{(1)}) U_{mlj} + (v_3 \sum B_{ji}^{(2)} + v_4 \sum B_{ji}^{(1)}) U_{mhi} + v_5 U_{mhj} \\ &+ v_6 \sum\sum B_{ji}^{(1)} A_{hl}^{(1)} V_{mli} + v_7 \sum A_{hl}^{(1)} V_{mlj} + v_8 \sum B_{ji}^{(1)} V_{mhi} + v_9 V_{mhj} + v_{10} W_{mhj} \\ &+ v_{11} \sum A_{ml}^{(1)} W_{nlj} = \hat{\rho} \frac{\partial^2 U_m}{\partial \hat{t}^2} \\ &v_6 \sum\sum B_{ji}^{(1)} A_{hl}^{(1)} U_{mli} + v_{12} \sum A_{hl}^{(1)} U_{mli} + v_{13} \sum B_{ji}^{(1)} U_{mhi} + v_{14} U_{mhj} \\ &+ (v_{15} \sum A_{hl}^{(2)} + v_{16} \sum A_{hl}^{(1)}) V_{mlj} + (v_{17} \sum B_{ji}^{(2)} + v_{18} \sum B_{ji}^{(1)}) V_{mhi} \\ &+ v_{19} V_{mhj} + v_{20} \sum B_{ji}^{(1)} W_{mhi} + v_{21} W_{mhj} = \hat{\rho} \frac{\partial^2 V_m}{\partial \hat{t}^2} \end{aligned} \tag{3.17}$$

$$
\begin{aligned}
&\nu_{11}\sum A_{hl}^{(1)}U_{mlj}+\nu_{22}U_{mhj}+\nu_{23}\sum B_{ji}^{(1)}V_{mhi}+\nu_{21}V_{mhj}\\
&+(\nu_{24}\sum A_{hl}^{(2)}+\nu_{25}\sum A_{hl}^{(1)})W_{mlj}+(\nu_{26}\sum B_{ji}^{(2)}+\nu_{27}\sum B_{ji}^{(1)})W_{mhi}\\
&+\nu_{28}W_{mhj}=\hat{\rho}\frac{\partial^2 W_m}{\partial\hat{t}^2}
\end{aligned}
$$

where the indices h, j and corresponding dummy indices i, l are used for the displacement components U_m, V_m, W_m (Bert and Malik 1996b). The range of indices is given by

$$h, l=1\cdots N1; \quad i, j=1\cdots N$$

Using the quadrature rules the boundary conditions on the surface of toroidal shell $r = const$ are given by

$$
\begin{aligned}
&d_{11}\Sigma A_{hl}^{(1)}U_{mli}+(\frac{d_{12}}{r}+\frac{d_{13}\cos\phi}{\rho})U_{mhj}+\frac{d_{12}}{r}\Sigma B_{ji}V_{mhi}\\
&-\frac{d_{13}\sin\phi}{\rho}V_{mhj}+\frac{d_{13}2m}{\rho}W_{mhj}=0\\
&\frac{1}{2}\frac{d_{55}(-2m)}{\rho}U_{mhj}+\frac{1}{2}d_{55}\Sigma A_{hl}^{(1)}W_{mlj}-\frac{1}{2}\frac{d_{55}\cos\phi}{\rho}W_{mhj}=0 \qquad (3.18)\\
&\frac{1}{2}\frac{d_{66}}{r}\Sigma B_{ji}^{(1)}U_{mhi}+\frac{1}{2}d_{66}\Sigma A_{hl}^{(1)}V_{mlj}-\frac{1}{2}\frac{d_{66}}{r}V_{mhj}=0
\end{aligned}
$$

where the d_{ij} have been given in Section 4.2. The range of indices is

$$i=1\cdots N; \quad l=1\cdots N1; \quad h=1, N1; \quad j=1, N$$

Similarly the governing and boundary equations can be set up for the axisymmetric case and odd harmonics.

The assembly of the domain and boundary equations yields a matrix equation for all these cases of the form

$$\begin{bmatrix} S_{bb} & S_{bd} \\ S_{db} & S_{dd} \end{bmatrix} \begin{Bmatrix} \Delta_b \\ \Delta_d \end{Bmatrix} - \beta^2 \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{Bmatrix} \Delta_b \\ \Delta_d \end{Bmatrix} = \begin{Bmatrix} q \\ 0 \end{Bmatrix} \tag{3.19}$$

This equation serves as the basis for the solution of the problems of vibration and statics as discussed in the preceding. The vector q contains the load values at the boundary sampling points, and $\beta^2 = (\hat{\rho}\omega^2\bar{r}^2)/E$ is the frequency parameter, where $\bar{r}$ is the mean radius of the cross-section. For the vibration problem the load term q is zero, while for the static problem the frequency parameter β^2 is zero. The sub-matrices S_{bb} and S_{bd} stem from the boundary conditions, and involve respectively the unknowns of the boundary and domain mesh points. The sub-matrices S_{db} and S_{dd} stem from the governing equations, and similarly involve the unknowns of the boundary and domain points. The vectors Δ_b and Δ_d contain the displacements of the boundary and domain points respectively. For the vibration problem the former vector is eliminated using the static condensation technique (Bathe 1996). The matrix equation (3.19) is then reduced into the form

$$(-S_{db}S_{bb}^{-1}S_{bd} + S_{dd})\{\Delta_d\} = \beta^2\{\Delta_d\} \tag{3.20}$$

The smallest eigenvalue is found directly by solving a standard eigenvalue problem.

The theory presented in the preceding was coded in the *MATLAB*^{TM} programs Bandsta.m, Semista1.m, Semista2.m, Semiom1.m and Semiom2.m to cover respectively the statics and vibration problems. Results from these DQM programs are given in the following.

3.4 Finite Element Method

The commercial FEM program ADINA [ADINA 2000, Bathe 1996] was used to provide a second solution to the problems considered. Both three-dimensional models consisting of brick elements (ET FEM), and two-dimensional models consisting of shell elements (ST FEM) were used. The brick element in ADINA contains twenty nodes with three displacement degrees of freedom at each node. The shell element is an eight-noded isoparametric element with three displacements and three rotation degrees of freedom at each node.

The models adopted account for symmetry of the geometry and thus represent one eight of a toroid. The three-dimensional model is used to provide a second solution for all cases considered, while the shell theory model is used only for a statics problem involving an isotropic material. Sample meshes for both elasticity and shell theory models are discussed in section 3.5.

3.5 Validation and Results

Two problems were solved to obtain partial validation for the DQM solution. The first problem concerns the determination of the natural frequencies of a hollow toroid. The second problem concerns the determination of the stress state in a hollow toroid subjected to a set of band loads.

The geometry of the first validation problem is that of a complete hollow toroid. The polar axisymmetric vibrations of thick toroids were studied earlier by McGill and Lenzen [1967], while thin toroids (toroidal shells) were studied by Balderes and Armenakis [1973]. The β^2 values obtained from the program Semiom1.m for the lowest natural frequency for some thin toroids are compared in Table 3.1 with those given by Balderes and Armenakas [1973]. The sizes given for meshes refer respectively to the radial and meridional directions. In this table and subsequently ET indicates evaluation using the elasticity theory. Also given are frequencies obtained using the ET FEM. The DQM results agree well with the ET FEM results, and there is also good agreement with results given by Balderes and Armenakas [1973].

The geometry for the second validation problem is given in Fig. 3.3. A hollow steel toroid is subjected to a set of four band loads, each of which extends around the cross-section. The loads are spaced at 90° intervals and have an angular width of 2α. The band loadings are on the external surface of the toroid, and consist of uniform unit compression within the bands. Numerical results are found for the case of a = 100mm, b = 200 mm, R = 500mm, α = 8°. The material properties are E = 206GPa, and ν = 0.3. Values for stresses obtained from the program Bandsta.m are compared in Fig. 3.4 with those given by Zhu and Redekop [1995]. The values represented are the meridional stress on the external surface. Also presented are values obtained from the ET FEM. It is seen that the DQM and ET FEM results show very close agreement, and that good agreement is also obtained with the results of Zhu and Redekop [1995].

The DQM and FEM programs are used to obtain natural frequencies, displacements, and stresses for three geometry cases described in Table 3.2. Symbol $\bar{r}$ in this table indicates the mean radius of the cross-section and h indicates the thickness. The first two cases of the table are thin toroids identical to cases 9 and 10 discussed by Zhang and Redekop [1991]. The third case is a thick toroid, outside the range of shell theory. Two materials are considered, steel (isotropic) and M2 (transversely isotropic) [Chandrashekhara and Nanjunda Rao 1997]. The properties of the steel material are E = 206Gpa, ν = 0.3, while for the M2 material the properties are E' = 172.38GPa, E = 6.90GPa, G' = 3.45GPa, G = 2.76GPa, ν' = 0.25, ν = 0.25.

For the static problem three systems of local loading are considered. These systems consist respectively of unit normal pressure at the extrados, crown, and intrados, resembling the systems of loads described by Zhang and Redekop [1991]. The pressures p_1, p_2, and p_3 of Fig. 3.5 show the three types of systems in the first quadrant. Identical loading are understood to be applied in the other three quadrants, and thus planes $\theta = 0^{\circ}$, $\theta = 90^{\circ}$, $\phi = 0^{\circ}$ are planes of symmetry.

For the p_1 load the unit normal pressure extends over the range $30^o \le \theta \le 60^o$, $-30^o \le \phi \le 30^o$. For the p_2 load the range is $30^o \le \theta \le 60^o$, $75^o \le \phi \le 105^o$. This loading represents a 'pinching' of the toroid since a similar load pad exists on the other side of the $\phi = 0^0$ plane. For the p_3 load the range is $30^o \le \theta \le 60^o$, $150^o \le \phi \le 210^o$.

Analyses are carried out using the DQM, the ET FEM, and the ST FEM. Typical meshes are given in Figs. 3.6 - 3.8. For the semi-analytical DQM used here the meshing of a typical cross-section is as shown in Fig. 3.6. The mesh indicated is 48×12, representing respectively points in the meridional and radial directions. Symmetry is accounted for in the FEM analyses, enabling the analysis of one-eight of the toroid only (Fig. 3.7 – 3.8). The ET FEM mesh of Fig. 3.7 is of size $2 \times 24 \times 48$, representing respectively points in the radial, meridional, and circumferential directions. The ST FEM mesh of Fig. 3.8 is of size 24×48, representing respectively points in the meridional and circumferential directions.

The convergence of the solution for the vibration problem is examined in Table 3.3. The frequency parameter $\hat{\rho}\omega^2$ for the three geometry cases of Table 3.2 as found using the DQM and the ET FEM are given for coarse, medium, and fine meshes for each of the two materials. Rapid convergence is observed for both methods. Passing from geometric case 1 to 3 the frequencies increase (as the wall thickness increases). The frequencies for the steel toroids are higher than those of the M2 material due to the greater stiffness. The agreement between DQM and FEM is best for the thick shells where the elasticity theory is most applicable.

Subsequently in this work the medium mesh was used in the ET DQM analysis, and the fine mesh for the ET FEM analysis.

Results for the three types of loading for the static problem are given in Tables 3.4- 3.5 and Fig. 3.9-3.25. Table 3.4 gives results for the isotropic (steel) toroids. The displacement, meridional and circumferential stresses for the three geometric cases, subjected to the three unit loadings, as determined by the DQM, ET FEM, and ST FEM are given. The displacement values given represent the normal displacement u at the center of the load pad on the outside surface. The stress values given represent the maximum stress under the load pad, on the outside surface. Table 3.5 gives similar results for the M2 toroids, but determined only by the DQM and the ET FEM.

In Tables 3.4 and 3.5 it is seen that passing from geometric cases 1 to 3 the displacement and both stresses decrease due to the increase in wall thickness. Displacement and stresses are generally lowest for loading at the extrados where the curvature is positive, i.e. of the spherical kind. There is a major increase in the displacement and circumferential stress level passing from the steel to the M2 material, but a decrease in the meridional stress level. This is because that the Young's modulus in the circumferential direction is larger in the meridional direction for M2 material. In general the DQM and FEM elasticity theory results are in close agreement. The FEM shell theory results generally underestimate the elasticity results.

Fig. 3.9 gives the variation of the meridional stress on the outside surface at the $\theta = 45^{o}$ cross-section for the p_3 loading case as determined by the DQM within a steel toroid for case3. Results are given for a number of different terms in the Fourier series, with convergence indicated at 51 terms.

Figs. 3.10 – 3.21 provide information about the variation of radial displacement u and meridional σ_{ϕ} stress within the toroid for the geometric case 3. The variation of the radial displacement u and meridional stress σ_{ϕ} on the outside surface at the $\theta = 45^{o}$ cross-section for the p_1, p_2, p_3 loading cases as determined by three methods is shown for a steel toroid in Figs. 3.10 – 3.15. Figs. 3.11, 3.13, 3.15 indicate that the maximum stress always appears under the load pad, and that the ST FEM shows significant difference from the other two solutions.

Fig. 3.10 shows that the extrados load causes an outward displacement at the intrados and crown, with the latter being greater in magnitude. There is an inward displacement at the crown of greater magnitude. The shell theory values are seen to underestimate the maximum displacement by approximately 12%.

Fig. 3.12 shows that the pinching effect causes an outward displacement at the extrados and intrados, with the latter being greater in magnitude. Simultaneously there is an inward displacement at the crown of similar magnitude. The shell theory values are seen to underestimate the maximum displacement by approximately 10%.

Fig. 3.14 shows that the intrados load causes an outward displacement at the extrados and crown, with the latter being greater in magnitude. There is an inward displacement at the intrados of greater magnitude. The shell theory values are seen to underestimate the maximum displacement by approximately 8%.

The variation of the radial displacement u and meridional stress σ_ϕ on the outside surface at the $\theta = 45^\circ$ cross-section for the p_1, p_2, p_3 loading cases as determined by two methods is shown for an M2 toroid in Figs. 3.16 – 3.21. The values have the same trends as for the steel toroid discussed in the previous paragraphs. The magnitudes however are different, due to the differences in the material stiffness.

Fig. 3.22 shows the variation of the circumferential stress on the outside surface at the $\phi = 0^0$ plane, i.e. about the 0° – 180° line on the figure, for the p_1 loading case for a M2 toroid. The load has little effect on the $\theta = 0^\circ, 90^\circ$. There is symmetry about the $\theta = 45^\circ$ cross-section.

Contour lines are shown in Figs. 3.23 – 3.25 for the meridional stress on the $\theta = 45^\circ$ cross-section for three loading cases for a M2 toroid as determined by the DQM. For both loading cases, there is symmetry about the $\phi = 0^0$ plane. The higher stress levels clearly are at the extrados for the p_1 loading case, at the crown for the p_2 loading case, and at the intrados for the p_3 loading case, respectively. The variation of stress over the thickness under the loading clearly deviates from the linear pattern assumed in a shell theory solution.

The previous study by Zhang and Redekop [1991] covered the local loading of steel thin-walled 90° elbows with 'shear diaphragm' supports at the ends. Case 9 and Case 10 of that study are geometrically similar to case 1 and 2 of the present study, with identical loading patterns. The displacements and stresses cited in Tables 2 and 3 of Zhang and Redekop [1991] are generally higher than the corresponding values given in Table 3.4 of the current study. This is attributed partially to the more flexible support condition provided by the 'shear diaphragm'. The radial displacement curve for a loading at the extrados shown in Fig. 9 of Zhang and Redekop [1991] is qualitatively similar to the displacement curve given in Fig. 3.10 of the current study.

3.6 Conclusions

Results obtained using the theory developed in this chapter agree well with finite element method results for the vibration and static problems. The new work provides a valuable, efficient tool for the analysis of toroidal shells which can be used to supplement results obtained using the traditional finite element method.

Chapter 4

VIBRATION AND STATICS OF ORTHOTROPIC THIN-WALLED TOROIDAL SHELLS

4.1 Introduction

Toroidal shells find widespread application in complete as well as incomplete geometric form. Complete toroidal shells spanning 360^0 in the circumferential and meridional directions are used for example as pressure vessels. Incomplete toroidal shells (sectional shells) spanning through less than 360^0 in the meridional directions find use mostly as vehicle or aircraft tires. In both of these applications the shell thickness may be variable.

Considerable work has been done on the vibration and statics of thin-walled isotropic toroidal shells for which the wall thickness is uniform. Studies in which the isotropic model is relaxed include those of Grigorenko et al. [1995], Xi et al. [1999], Noor and Peters [1987], and Noor and Burton [1990]. Studies of toroidal shells in which the wall-thickness is considered variable include those of Gaidaichuck et al. [1978], Thomas [1980], Bielski

[1992], and Jiang and Redekop [2002a]. The latter studies deal mostly with problems of shell buckling and vibration.

The majority of the studies on toroidal shells have relied on analytical or finite element solutions. Compared to analytical solutions, the differential quadrature method (DQM) can be more flexible in dealing with boundary conditions since the choice of functions is not specific to boundaries, The DQM can be computationally more efficient than the finite element method (FEM), since the governing equations are dealt with directly. In a recent study, DQM approach was shown to be suitable to set up benchmark problems to check the accuracy of other numerical solutions, and to carry out efficient parametric studies.

In this chapter a study is made of the static and vibration characteristics of orthotropic toroidal shells having variable wall thickness. The solution procedure is tailored to match the complicated requirements of the model. The Sanders-Budiansky shell theory [Sanders 1959, Budiansky 1968] is developed in toroidal coordinates accounting for an orthotropic material and a variable thickness. In the solution a semi-analytical DQM approach is used in which Fourier series are written in the circumferential direction. This approach reduces the work to a series of problems for the individual harmonics which are mathematically one-dimensional.

The theory developed is used to solve two cases of statics and vibration. In the first case a complete toroidal shell is considered, with internal pressure loading in the statics application. An isotropic material model is used, and the shell wall is considered to be thinner at the extrados. Two DQM solutions, having either harmonic or power series as trial functions in the meridional direction, are developed. In the second case a toroidal shell incomplete in the meridional direction is considered, with a pad loading at the extrados in the statics application. An orthotropic material model is used, and the shell wall is considered to be thicker at the extrados. Results obtained for the statics and vibration characteristics in both problems are compared with results obtained using the FEM.

4.2 Sanders-Budiansky Shell Theory

The toroidal shell (Fig. 4.1) has a bend radius R, a mean cross-sectional radius r, and a thickness $h(\phi)$, which, varies with the meridional angle ϕ. The shell is taken as complete in the circumferential direction θ, extending through 360^0. For the meridional direction two alternatives are considered. In the first (Fig. 4.2) the shell is complete, while in the second (Fig. 4.3) the shell extends through an angle of 2β. A general point P on the shell mid-surface is defined by the circumferential and meridional coordinates θ and ϕ. The radius vector **r** to P is given by

$$\mathbf{r} = (R + r\cos\phi)\sin\theta\,\mathbf{i} + (R + r\cos\phi)\cos\theta\,\mathbf{j} + r\sin\phi\,\mathbf{k}, \tag{4.1}$$

where **i**, **j**, **k** are unit vectors in the Cartesian coordinates system. The first fundamental form of the middle surface is given by

$$ds^2 = A_1^2 d\xi_1^2 + A_2^2 d\xi_2^2 \tag{4.2}$$

A_1 and A_2 are the Lamé parameters, while ξ_1 and ξ_2 are the base coordinates. Following the convention of Redekop [1990] the base coordinates are taken as

$$\xi_1 = \phi; \qquad \xi_2 = \eta = \rho\theta \tag{4.3}$$

where $\rho = \dfrac{R}{r}$, the Lamé parameters are

$$A_1 = r; \qquad A_2 = r\varsigma \tag{4.4}$$

where

$$\varsigma = 1 + \gamma\cos\phi; \qquad \gamma = \frac{r}{R} = \rho^{-1}$$

The principal radii of curvature for the meridional (R_1) and circumferential (R_2) directions are

$$R_1 = r; \qquad R_2 = \frac{r\varsigma}{\gamma\cos\phi} \tag{4.5}$$

The subscripts ϕ and η in the following indicate respectively the shell meridional and circumferential directions, while r indicates the outward normal direction.

The wall thickness is defined by

$$h = t_1 + t_2\cos\phi \tag{4.6}$$

where t_1, t_2 are constants. Two models are considered for the wall thickness. For the shell complete in the meridional direction the thickness is assumed least at the extrados, corresponding to a piping elbow geometry. For the shell incomplete in the meridional direction the thickness is assumed greatest at the extrados, corresponding to a vehicle or aircraft tire geometry. The mean thickness or thickness of a shell having a uniform wall is represented by t. It is noted that the shell gross behavior depends strongly on the toroidal aspect ratio $\gamma = \dfrac{r}{R}$, the shell thickness ratio $\dfrac{r}{t}$, and the meridional angle 2β.

The equilibrium equations in the Sanders-Budiansky theory are given by [Sanders 1959 and Budiansky 1968]

$$\frac{\partial(\varsigma N_\phi)}{\partial\phi}+\frac{\partial N_{\phi\eta}}{\partial\eta}+\gamma\sin\phi N_\eta+\frac{1}{r}\frac{\partial(\varsigma M_\phi)}{\partial\phi}+\frac{1+(2\varsigma)^{-1}}{r}\frac{\partial M_{\phi\eta}}{\partial\eta}$$
$$+\frac{\gamma\sin\phi}{r}M_\eta=0$$

$$\frac{\partial(\varsigma N_{\phi\eta})}{\partial\phi}+\frac{\partial N_\eta}{\partial\eta}-\gamma\sin\phi N_{\phi\eta}+\frac{\gamma\cos\phi}{r\varsigma}\frac{\partial M_\eta}{\partial\eta}+\frac{2\gamma\cos\phi-1}{2r}\frac{\partial M_{\phi\eta}}{\partial\phi}$$
$$-\frac{2\gamma^2\sin 2\phi+\gamma\sin\phi}{2r\varsigma}M_{\phi\eta}=0 \tag{4.7}$$

$$\frac{N_\phi}{r}+\frac{\gamma\cos\phi}{\varsigma r}N_\eta-\frac{1}{r^2}\frac{\partial^2 M_\phi}{\partial\phi^2}-\frac{1}{(r\varsigma)^2}\frac{\partial^2 M_\eta}{\partial\eta^2}-\frac{2}{(r\varsigma)^2}\frac{\partial}{\partial\theta}(\varsigma\frac{\partial M_{\phi\eta}}{\partial\phi})$$
$$-\frac{\gamma\sin\phi}{r^2\varsigma}\frac{\partial}{\partial\phi}(M_\eta-2M_\phi)-\frac{\gamma\cos\phi}{r^2\varsigma}(M_\eta-M_\phi)=p_r$$

where N_ϕ, N_η, $N_{\phi\eta}$, M_ϕ, M_η and $M_{\phi\eta}$ are the membrane stress resultants and bending stress resultants defined in the theory, and Q_ϕ, Q_η are shear resultants. For a single generally orthotropic layer of thickness, h, and lamina stiffness, Q_{ij}, stresses are defined as

$$\begin{bmatrix}\sigma_\phi\\ \sigma_\eta\\ \tau_{\phi\eta}\end{bmatrix}=\begin{bmatrix}Q_{11} & Q_{12} & 0\\ Q_{12} & Q_{22} & 0\\ 0 & 0 & Q_{66}\end{bmatrix}\begin{bmatrix}\varepsilon_\phi\\ \varepsilon_\eta\\ \varepsilon_{\phi\eta}\end{bmatrix} \tag{4.8}$$

where

$$Q_{11}=\frac{E_1}{1-\nu_{12}\nu_{21}};\quad Q_{12}=\frac{E_1\nu_{21}}{1-\nu_{12}\nu_{21}};\quad Q_{22}=\frac{E_2}{1-\nu_{12}\nu_{21}};\quad Q_{66}=G_{12};\quad \frac{\nu_{12}}{E_1}=\frac{\nu_{21}}{E_2}$$

The E_1, E_2, G_{12}, ν_{12},ν_{21} are Young's modulus, shear moduli, and Poisson's ratios, respectively. ε_ϕ, ε_η, $\varepsilon_{\phi\eta}$ are meridional strain, circumferential strain, and shear strain, and σ_ϕ, σ_η, $\tau_{\phi\eta}$ are meridional stress, circumferential stress, and shear stress.

The lamina stiffness are

$$A_{ij} = Q_{ij}\,t \qquad\qquad D_{ij} = \frac{Q_{ij}t^3}{12}$$

There is no coupling between bending and extension, so the force and moment resultants are

$$\begin{bmatrix} N_\phi \\ N_\eta \\ N_{\phi\eta} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{16} \\ A_{12} & A_{22} & A_{26} \\ A_{16} & A_{26} & A_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_\phi \\ \varepsilon_\theta \\ \varepsilon_{\phi\theta} \end{bmatrix} \qquad \begin{bmatrix} M_\phi \\ M_\eta \\ M_{\phi\eta} \end{bmatrix} = \begin{bmatrix} D_{11} & D_{12} & D_{16} \\ D_{12} & D_{22} & D_{26} \\ D_{16} & D_{26} & D_{66} \end{bmatrix} \begin{bmatrix} \kappa_\phi \\ \kappa_\eta \\ \kappa_{\phi\eta} \end{bmatrix} \tag{4.9}$$

The strain – displacement and curvature – displacement relations in this theory, expressed in toroidal coordinates [Redekop 1992, Leung and Kwok 1994], are

$$\begin{aligned}
\varepsilon_\phi &= \frac{1}{r}\left(\frac{\partial u}{\partial \phi} + w\right); \\
\varepsilon_\eta &= \frac{1}{r\varsigma}\left(-\gamma \sin\phi + \frac{\partial v}{\partial \eta} + \gamma \cos\phi w\right); \\
\varepsilon_{\phi\eta} &= \frac{1}{2r\varsigma}\left(\frac{\partial u}{\partial \eta} + \gamma \sin\phi v\right) + \frac{1}{2r}\frac{\partial v}{\partial \phi}; \\
\kappa_\phi &= \frac{1}{r^2}\left(\frac{\partial u}{\partial \phi} - \frac{\partial^2 w}{\partial \phi^2}\right) \\
\kappa_\eta &= -\frac{\sin\phi}{rR\varsigma}\left(u - \frac{\partial w}{\partial \phi}\right) + \frac{\cos\phi}{rR\varsigma^2}\frac{\partial v}{\partial \eta} - \frac{1}{(r\varsigma)^2}\frac{\partial^2 w}{\partial \eta^2} \\
\kappa_{\phi\eta} &= \frac{(1+2\varsigma)}{(2r\varsigma)^2}\frac{\partial u}{\partial \eta} + \frac{1}{2r^2\varsigma}\left(\gamma\cos\phi - \frac{1}{2}\right)\frac{\partial v}{\partial \phi} \\
&+ \frac{2\gamma^2 \sin 2\phi + (1-2\varsigma)\gamma \sin\phi}{(2r\varsigma)^2}v - \frac{1}{r^2\varsigma}\frac{\partial^2 w}{\partial \phi \partial \eta} - \frac{1}{rR\varsigma^2}\frac{\partial w}{\partial \eta}
\end{aligned} \tag{4.10}$$

where u, v, w are the mid-surface displacements components in the meridional, circumferential, and radial directions. The displacement components and stress resultants are represented in their positive sense in Fig. 4.4.

The kinematics' relations (4.10) are substituted into the expressions for the stress resultants (4.9). To account for inertial effects in the vibration analysis the D'Alembert principle is invoked. Substituting then the stress resultants into the equilibrium equations (4.7) yields the governing equations as

$$
\begin{aligned}
L_{11}u + L_{12}v + L_{13}w &= \frac{\partial^2 u}{\partial \hat{t}^2} \\
L_{21}u + L_{22}v + L_{23}w &= \frac{\partial^2 v}{\partial \hat{t}^2} \qquad (4.11) \\
L_{31}u + L_{32}v + L_{33}w &= q_r + \frac{\partial^2 w}{\partial \hat{t}^2}
\end{aligned}
$$

where $q_r = p_r /(r\varsigma^2 \hat{\rho} h)$, $\hat{\rho}$ is mass density, $\hat{t}$ is the time, and the L_{ij} are differential operators given by

$$
\begin{aligned}
L_{11} &= f_1 \frac{\partial^2}{\partial \phi^2} + f_2 \frac{\partial}{\partial \phi} + f_3 \frac{\partial^2}{\partial \eta^2} + f_4 \\
L_{12} &= f_5 \frac{\partial^2}{\partial \phi \partial \eta} + f_6 \frac{\partial}{\partial \eta} \\
L_{13} &= f_7 \frac{\partial^3}{\partial \phi^3} + f_8 \frac{\partial^2}{\partial \phi^2} + f_9 \frac{\partial}{\partial \phi} + f_{10} \frac{\partial^4}{\partial^2 \phi \partial^2 \eta} + f_{11} \frac{\partial^2}{\partial \eta^2} + f_{12} \\
L_{21} &= f_{13} \frac{\partial^2}{\partial \phi \partial \eta} + f_{14} \frac{\partial}{\partial \eta} \qquad (4.12) \\
L_{22} &= f_{15} \frac{\partial^2}{\partial \phi^2} + f_{16} \frac{\partial}{\partial \phi} + f_{17} \frac{\partial^2}{\partial \eta^2} + f_{18} \\
L_{23} &= f_{19} \frac{\partial^3}{\partial \phi \partial \eta^2} + f_{20} \frac{\partial^2}{\partial \phi \partial \eta} + f_{21} \frac{\partial^3}{\partial \eta^3} + f_{22} \frac{\partial}{\partial \eta}
\end{aligned}
$$

$$L_{31} = f_{23} \frac{\partial^3}{\partial \phi^3} + f_{24} \frac{\partial^2}{\partial \phi^2} + f_{25} \frac{\partial}{\partial \eta} + f_{26} \frac{\partial^3}{\partial \phi \partial \eta^2} + f_{27} \frac{\partial^2}{\partial \eta^2} + f_{28}$$

$$L_{32} = f_{29} \frac{\partial^3}{\partial \phi^2 \partial \eta} + f_{30} \frac{\partial^2}{\partial \phi \partial \eta} + f_{31} \frac{\partial^3}{\partial \eta^3} + f_{32} \frac{\partial}{\partial \eta}$$

$$L_{33} = f_{33} \frac{\partial^4}{\partial \phi^4} + f_{34} \frac{\partial^3}{\partial \phi^3} + f_{35} \frac{\partial^2}{\partial \phi^2} + f_{36} \frac{\partial}{\partial \phi} + f_{37} \frac{\partial^4}{\partial \phi^2 \partial \eta^2}$$
$$+ f_{38} \frac{\partial^3}{\partial \phi \partial \eta^2} + f_{39} \frac{\partial^4}{\partial \eta^4} + f_{40} \frac{\partial^2}{\partial \eta^2} + f_{41}$$

The functions $f_1 - f_{41}$ are given in the Appendix B. It is clear that for the static problem the time derivatives are zero, while for the vibration problem $q_r = 0$.

In the case of a shell incomplete in the meridional direction, boundary conditions must be satisfied on the $\phi = cnst$ surfaces. For a toroidal shell with clamped meridional edges the boundary conditions at the edges $\phi = \pm\beta$ are

$$u = 0; \quad v = 0; \quad w = 0; \quad \frac{\partial w}{\partial \phi} = 0; \tag{4.13}$$

In the case of a shell complete in the meridional direction, continuity conditions over the $\phi = 360^o$ position are automatically satisfied if harmonic trial functions are used. Alternatively if polynomial trial functions are used continuity over the $\phi = 360^o$ position is obtained by enforcing the four displacements relations

$$u_1 = u_N; \quad v_1 = v_N; \quad w_1 = w_N; \quad \frac{\partial w_1}{\partial \phi} = \frac{\partial w_N}{\partial \phi} \tag{4.14}$$

and the four force resultants relations

$$
\begin{aligned}
&\hat{L}_{11}u_1 + \hat{L}_{12}v_1 + \hat{L}_{13}w_1 = \hat{L}_{11}u_N + \hat{L}_{12}v_N + \hat{L}_{13}w_N \\
&\hat{L}_{21}u_1 + \hat{L}_{22}v_1 + \hat{L}_{23}w_1 = \hat{L}_{21}u_N + \hat{L}_{22}v_N + \hat{L}_{23}w_N \\
&\hat{L}_{31}u_1 + \hat{L}_{32}v_1 + \hat{L}_{33}w_1 = \hat{L}_{31}u_N + \hat{L}_{32}v_N + \hat{L}_{33}w_N \\
&\hat{L}_{41}u_1 + \hat{L}_{42}v_1 + \hat{L}_{43}w_1 = \hat{L}_{41}u_N + \hat{L}_{42}v_N + \hat{L}_{43}w_N
\end{aligned}
\tag{4.15}
$$

In eqns. (4.14-4.15) u_1, v_1, w_1 are the displacements at the position $\phi = 0^o$, while u_N, v_N, w_N are the displacements at the position $\phi = 360^o$. The differential operators in eqn. (4.15) are given by

$$
\begin{aligned}
&\hat{L}_{11} = g_1 \frac{\partial^2}{\partial \phi^2} + g_2 \frac{\partial}{\partial \phi} + g_3 \frac{\partial^2}{\partial \eta^2} + g_4 \\
&\hat{L}_{12} = g_5 \frac{\partial^2}{\partial \phi \partial \eta} + g_6 \frac{\partial}{\partial \eta} \\
&\hat{L}_{13} = g_7 \frac{\partial^3}{\partial \phi^3} + g_8 \frac{\partial^2}{\partial \phi^2} + g_9 \frac{\partial}{\partial \phi} + g_{10} \frac{\partial^3}{\partial \phi \partial \eta^2} + g_{11} \frac{\partial^2}{\partial \eta^2} \\
&\hat{L}_{21} = g_{12} \frac{\partial}{\partial \eta} \\
&\hat{L}_{22} = g_{13} \frac{\partial}{\partial \phi} + g_{14} \\
&\hat{L}_{23} = g_{15} \frac{\partial^2}{\partial \phi \partial \eta} + g_{16} \frac{\partial}{\partial \eta} \\
&\hat{L}_{31} = g_{17} \frac{\partial}{\partial \phi} + g_{18} \\
&\hat{L}_{32} = g_{19} \frac{\partial}{\partial \eta} \\
&\hat{L}_{33} = g_{20}
\end{aligned}
\tag{4.16}
$$

$$\hat{L}_{41} = g_{21}\frac{\partial}{\partial \phi} + g_{22}$$

$$\hat{L}_{42} = g_{23}\frac{\partial}{\partial \eta}$$

$$\hat{L}_{43} = g_{24}\frac{\partial^2}{\partial \phi^2} + g_{25}\frac{\partial}{\partial \phi} + g_{26}\frac{\partial^2}{\partial \eta^2}$$

The functions $g_1 - g_{26}$ are given in Appendix C.

Taking a semi-analytical approach Fourier series expansions are written in the circumferential direction, and the various harmonics are handled separately. Periodic harmonics are chosen to automatically satisfy the continuity conditions in the circumferential direction. The series for the displacements in the statics problem thus are taken as

$$u = U_0 + \Sigma U_m(\phi)\cos m\theta; \quad v = \Sigma V_m(\phi)\sin m\theta$$

$$w = W_0 + \Sigma W_m(\phi)\cos m\theta; \quad p_r = P_{r0} + \Sigma P_{rm}(\phi)\cos m\theta \tag{4.17}$$

where m is the number of the harmonic $(m = 1, \cdots, M)$, $U_m \equiv U_m(\phi)$, $V_m \equiv V_m(\phi)$, $W_m \equiv W_m(\phi)$ are unknown displacement functions of the mth harmonic, and $P_{rm} = P_{rm}(\phi)$ is a known loading coefficients. The U_0, W_0, P_{r0} set defines the zeroth harmonic term, which is a special simpler case. For the vibration problem the expressions for the mth harmonic are taken as

$$u = U_m(\phi)\sin m\theta e^{i\omega\hat{t}}; \quad v = V_m(\phi)\cos m\theta e^{i\omega\hat{t}}; \quad w = W_m \sin m\theta e^{i\omega\hat{t}} \tag{4.18}$$

where ω is the natural frequency. Solely modes which are symmetrical with respect to the toroidal equator ($z \equiv 0$) are represented in these expansions. Antisymmetrical modes are readily obtained by reversing the selection of the trigonometric terms.

4.3 Differential Quadrature Method

The DQM approach [Bert and Malik1996b, Shu 2000] is used to determine the unknown functions U_m, V_m, W_m. As a semi-analytical approach is used in the current study the one-dimensional DQM is required, and all unknown functions can be expressed in terms of one variable (ϕ) only. A brief overview of the method is presented in the following for completeness.

The basis of the DQM is the meshing of the domain and the representation in the domain of the derivatives of unknown functions by a weighted sum of function values, i.e. in the present case

$$\left.\frac{d^k f}{d\phi^k}\right|_{\phi=\phi_i} = \sum_{l=1}^{N} A_{il}^{(k)} f(\phi_l) \tag{4.19}$$

Here the $f(\phi_l)$ are the function values at the sampling points l, the $A_{il}^{(k)}$ are weighting coefficients of the $k-th$ order at the $i-th$ sampling point of the mesh in the ϕ direction, and N is the number of sampling points in this direction.

In the current study for the geometric case where $2\beta < 360^o$ the use of polynomial trial functions is appropriate. Thus for this case the choice of trial function is

$$f(\phi) = 1, \phi, \phi^2, ..., \phi^{N-1} \tag{4.20}$$

For these functions explicit formulas for the weighting coefficients may be used. For the first order derivative the formulas are [Bert and Malik 1996b]

$$\begin{aligned} A_{il}^{(1)} &= \frac{\pi(\phi_i)}{(\phi_i - \phi_l)\pi(\phi_l)}; \quad i,l = 1,2,...,N; i \neq l \\ \pi(\phi_i) &= \prod_{l=1}^{N} (\phi_i - \phi_l); \quad i \neq l \end{aligned} \tag{4.21}$$

while for the higher order derivatives the formulas are

$$\begin{aligned} A_{il}^{(k)} &= k[A_{il}^{(k-1)} A_{il}^{(1)} - \frac{A_{il}^{(k-1)}}{\phi_i - \phi_l}]; \; i,l = 1,2,\ldots,N; \; i \neq l; \; 2 \leq k \leq (N-1) \\ A_{il}^{(k)} &= A_{ii}^{(k)} = -\sum_{s=1}^{N} A_{is}^{(k)}; \; i = 1,2,\ldots,N; \; i \neq s; \; 1 \leq k \leq (N-1) \end{aligned} \tag{4.22}$$

For the sampling points ϕ_l in the meridional direction the Chebyshev-Gauss-Lobatto spacing (Shu 2000) is used. In this scheme the coordinates ϕ_l are taken as $\phi_l = \phi_L + (\phi_H - \phi_L) y_l$ where ϕ_L, ϕ_H represent the low and high values of ϕ in the domain. The y_l is given by

$$\begin{aligned} & y_1 = 0; \quad y_2 = \delta; \quad y_{N-1} = 1 - \delta; \quad y_N = 1 \\ & y_l = \frac{1 - \cos\frac{\pi(l-2)}{N-3}}{2}; \; 2 < l < (N-1); \;\; \delta \approx 10^{-5} \end{aligned} \tag{4.23}$$

In the case when there is cyclic periodicity in the domain (meridional direction) it is customary to choose uniformly spaced points and harmonic trial functions. With such harmonic functions the periodicity conditions are identically satisfied. Thus for the geometric case where $2\beta = 360^o$ in the current study the trial function are taken as

$$f(\phi) = \cos[2(K-1)\pi\phi];\ K = 1,2,3,\ldots,N/2+1$$

$$f(\phi) = \sin[2(K-N/2-1)\pi\phi];\ K = N/2+2, N/2+3,\ldots,N \qquad (4.24)$$

where N is an even number.

As polynomial functions are reputed to lead to faster convergence rates than harmonic functions it was decided to attempt in the current study also the use of polynomial functions for the meridionally complete case. Such an approach necessitates the satisfying of continuity conditions over the position $\phi = 0^o \equiv 360^o$. Results found using harmonic trial functions are termed DQM1 in the following, while results found using polynomial trial functions are termed DQM2.

At each sampling point either the DQM analogue of a governing equation for the domain is represented, or of a boundary equation. For the model there are four boundary conditions at each boundary, while there are only three governing equations. It is necessary to enforce one of the boundary equations at an interior point [Bert and Malik 1996b]. This point, a 'δ' point, is taken a short distance ($\delta \cong 10^{-5}$ on a unit domain) from the boundary point. The boundaries of the current study are located on the meridional edges of the shell, arising due to the use of fixed boundary conditions or continuity boundary conditions in the solution.

Substitution of (4.17) or (4.18) and the quadrature rule (4.19) for the ϕ derivatives into the governing equations (4.11), leads to three DQM domain equations. For the typical ith sampling point these equations have the form

$$\begin{aligned}
&(h_1\Sigma A_{il}^{(2)} + h_2\Sigma A_{il}^{(1)})U_{ml} + h_3U_{mi} + h_4\Sigma A_{il}^{(1)}V_{ml} \\
&+ h_5V_{mi} + (h_6\Sigma A_{il}^{(3)} + h_7\Sigma A_{il}^{(2)} + h_8\Sigma A_{il}^{(1)})W_{ml} + h_9W_{mi} = \lambda U_{mi} \\
&h_{10}\Sigma A_{il}^{(1)}U_{ml} + h_{11}U_{mi} + (h_{12}\Sigma A_{il}^{(2)} + h_{13}\Sigma A_{il}^{(1)})V_{ml} \\
&+ h_{14}V_{mi} + (h_{15}\Sigma A_{il}^{(2)} + h_{16}\Sigma A_{il}^{(1)})W_{ml} + h_{17}W_{mi} = \lambda V_{mi} \\
&(h_{18}\Sigma A_{il}^{(3)} + h_{19}\Sigma A_{il}^{(2)} + h_{20}\Sigma A_{il}^{(1)})U_{ml} + h_{21}U_{mi} + (h_{22}\Sigma A_{il}^{(2)} + h_{23}\Sigma A_{il}^{(1)})V_{ml} + h_{24}V_{mi} \\
&+ (h_{25}\Sigma A_{il}^{(4)} + h_{26}\Sigma A_{il}^{(3)} + h_{27}\Sigma A_{il}^{(2)} + h_{28}\Sigma A_{il}^{(1)})W_{nl} + h_{29}W_{ni} = q_r + \lambda W_{ni}
\end{aligned} \tag{4.25}$$

where the coefficients $h_1 - h_{29}$ are given in Appendix C, and the indices i and corresponding dummy indices l are used for the displacement components U_m, V_m, W_m [Bert and Malik 1996b]. The range of indices is given by

$$i, l = 1, 2, \cdots, N;$$

The boundary conditions and continuity conditions, where required, can similarly be transformed into a DQM form.

The assembly of the domain and boundary equations yields a set of simultaneous linear equations of the form

$$\begin{bmatrix} S_{bb} & S_{bd} \\ S_{db} & S_{dd} \end{bmatrix} \begin{Bmatrix} \Delta_b \\ \Delta_d \end{Bmatrix} + \lambda \begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix} \begin{Bmatrix} \Delta_b \\ \Delta_d \end{Bmatrix} = \begin{Bmatrix} 0 \\ q \end{Bmatrix} \tag{4.26}$$

This equation serves as the basis for the solution of the problem of vibration and statics as discussed in the preceding. The vector q contains the load values, and $\lambda = \omega^2$ is the frequency parameter. The sub-matrices S_{bb} and S_{bd} stem from the boundary conditions, and involve respectively the unknowns of the boundary and domain mesh points. The sub-matrices S_{db} and S_{dd} stem from the governing equations, and similarly involve the unknowns of the boundary and domain points. The vectors Δ_b and Δ_d contain the displacements of the boundary and domain points respectively. For the static problem $\lambda = 0$ and $q \neq 0$, while for the vibration problem $q = 0$ and $\lambda \neq 0$. The static problem can be solved for the displacements using standard Gauss elimination. For the vibration problem the static condensation technique is used [Bathe 1996]. The matrix equation (4.26) is then reduced into the form

$$[-S_{db}S_{bb}^{-1}S_{bd} + S_{dd}]\{\Delta_d\} = \lambda\{\Delta_d\} \tag{4.27}$$

This equation represents a standard eigenvalue problem from which the eigenvalue parameters λ and the corresponding mode shapes can be determined.

The theory presented in the preceding was coded in the *MATLAB™* programs Freqiso1.m, Afreqiso1.m, Freqiso2.m, Afreqiso2.m and Pvessel.m to cover respectively the vibration and static problems of isotropic toroidal shells, and programs Freqc.m, Afreqc.m and Orthsta.m to cover respectively the vibration and static problems of orthotropic toroidal shells. Results from these DQM programs are given in the numerical results section.

4.4 Finite Element Method

The commercial FEM program ADINA (ADINA 2000, Bathe 1996) was used to provide a second numerical solution for both the statics and vibration cases. For axisymmetric analyses the isobeam element was employed. For non-axisymmetric analyses the isoparametric shell element was used. This element has eight nodes, with a total of 48 degrees of freedom. In the definition of the material properties the quantities E_1, E_2, G_{12}, G_{13}, G_{23}, ν_{12}, ν_{21} are provided as data. To specify the partial loads in the statics analysis of case 2 the loaded elements are identified through the graphics, and then given a uniform normal pressure. The symmetry about the $z = 0$ plane was taken into account in most analyses, permitting substantial computational savings.

4.5 Validation and Results

The DQM method and ADINA FEM program are used to obtain natural frequencies, displacements and stresses for two separate sample cases.

4.5.1 Case 1

This case concerns an isotropic shell complete in the meridional direction defined by the parameters:

$$E = 2.07\times10^{11}\,Pa; \qquad \nu = 0.3; \quad G = \frac{E_1}{2(1+\nu_{12})}; \quad \hat{\rho} = 7770kg/m^3$$

$$R = 0.762m; \quad r = 0.25m; \quad t_1 = 0.009525; \quad t_2 = -0.001; \quad P_r = 1MPa$$

where t_1, t_2 refer to the thickness parameter of eqn. (4.6). For static analyses an internal pressure is considered, thus the static analysis is for an axisymmetric case.

Analyses are carried out using the DQM and FEM. The one-dimensional DQM model required is of a typical cross-section as shown in Fig. 4.5. The mesh indicated is of size 48, representing points in the meridional direction only. In the FEM analyses a two-dimensional model consisting of shell elements (Fig. 4.6) was used to determine the natural frequency. The mesh indicated is of size 24×96, representing respectively points in the meridional and circumferential directions. Axisymmetry is accounted for in the static analysis, enabling the analysis of one-dimensional model consisting of isobeam elements. The FEM mesh of Fig. 4.7 indicated is of size 192, representing points in the meridional direction. Results for the case of isotropic toroidal shells are given in Tables 4.1 – 4.2 and Figs. 4.8 – 4.16.

A study of the convergence of the FEM and DQM solutions in the vibration analysis of case 1 is given in Table 4.1. The six lowest frequencies are given for four meshes in each case. The identifying number m indicates the number of harmonic, while A and S represents symmetry or antisymmetry about the $z = 0$ plane. The DQM1 results are for harmonic trial functions, while the DQM2 are for polynomial functions. The three sets of converged values show very close agreement. It is seen that for coarse meshes the DQM2 results are closer to the final results than the DQM1 results, indicating an advantage for the polynomial trial functions. Mode shapes determined by the FEM and DQM are shown in Figs. 4.8-4.13, respectively.

A study of the convergence of the FEM and DQM solutions in the static analysis is given in Table 4.2. The DQM solution utilizes harmonic trial functions. The FEM and DQM results show close agreement for both displacement and stress values at the extrados ($\phi = 0^o$). A mesh of 36 nodes for the FEM or 24 sampling points for the DQM in the meridional direction gives convergence for engineering purpose. Results from the static solution given in Figs 4.15 – 4.16 indicate that thinning at the extrados ($\phi = 0^o$) leads to an increased stress level, and thickening at the intrados ($\phi = 180^o$) leads to a reduced stress level. It is seen that the change in stresses is closely proportional to the change in thickness. These observations agree with the conclusion reached by Thomas [Thomas 1980].

4.5.2 Case 2

This case concerns an orthotropic shell incomplete in the meridional direction consisting of a matrix and a set of circumferential cords. The values of the elastic constants of the material constituents simulated those of cord-rubber composites used in tires (NASA Technical Paper 2977) are given by

Constituent	Young's modulus, pa	Shear modulus, pa	Poisson's ratio
Matrix	3.15×10^6	1.057×10^6	0.49
Cord	2.45×10^9	4.9×10^6	0.66

The orthotropic elastic constants are computed by [Walter 1982]

$$E_1 = E_c f_c + E_r(1-f_c)$$
$$\nu_{12} = \nu_c f_c + \nu_r(1-f_c)$$

$$E_2 = \frac{E_r[E_c(1+2f_c)+E_r(1-f_c)}{E_c(1-f_c)+2E_r(1+0.5f_c)}$$
$$G_{12} = G_{13} = \frac{G_r[G_c+G_r+(G_c-G_r)f_c]}{[G_c+G_r-(G_c-G_r)f_c]}$$
$$G_{23} = 0.6G_{12}$$
$$\nu_{21} = \frac{\nu_{12}E_2}{E_1}$$

where subscripts r and c represent the quantities of the matrix and the cord, respectively, and f_c is the volume fraction of the cord. In the current study, f_c=0.359, and the other parameters have the values:

$$E_1 = 0.88157GPa; \quad \nu_{12} = 0.551; \quad E_2 = 0.0084GPa; \quad G_{12} = G_{13} = 0.00785GPa$$

$$G_{23} = 0.00471GPa; \quad \nu_{21} = 0.00525; \quad \hat{\rho} = 1190kg/m^3; \quad P_r = 1Pa$$

$$R = 0.38m; \quad r = 0.12m; \quad t_1 = 0.008; \quad t_2 = 0.0008;$$

where '1' represents the meridional coordinate (ϕ) and '2' represents the circumferential coordinate (η). The loading in the static analysis for case 2 is a nonaxisymmetric pad of uniform pressure centered at $\phi = 0^o, \theta = 0^o$, and extending through 30^0 in the circumferential direction and 40^0 in the meridional direction.

Analyses are again carried out using the DQM and FEM. For the semi-analytical DQM used here the meshing of a typical cross-section is as shown in Fig. 4.17. The mesh indicated is of size 48, representing points in the meridional direction only. Symmetry is accounted for in the FEM analyses, enabling the analysis of half of the cross-section only. The sample ST FEM mesh of Fig. 4.18 is of size 15×72, representing respectively points in the meridional and circumferential directions. Results of vibration and statics analyses are given in Table 4.3 and Figs. 4.19 – 4.32.

A study of convergence of the FEM and DQM solutions in the vibration analysis is given in Table 4.3. The six lowest frequencies are again given for four meshes in each case, and the single set of DQM results corresponding to the choice of polynomial trial functions. The two sets of converged values show close agreement. The mode shapes given in Figs. 4.19 – 4.24 are found from DQM and FEM. The symmetric modes and the antisymmetric modes are illustrated. All figures show the deformation of the toroidal shell and indicate the number of the circumferential wave.

The comparison of displacement and stress variation along the $\theta = 0^o$ cross section determined by DQM and FEM are given in Fig. 4.25 – 4.29. Fig. 4.25 gives the variation of radial displacement w determined by DQM. The maximum inward displacement appears at $\phi = 0^o$, and the maximum outward displacement appears at the $\phi = 90^o$. Results are given for a number of meshes in the DQM, with convergence indicated at a mesh size of 27.

Fig. 4.26 gives the variation of meridional stress σ_ϕ on the outside surface determined by DQM. The two largest compression stresses are at $\phi = 0^o$ and $\phi = 150^o$, and the largest tensile stress is close to $\phi = 90^o$. Results are given for a number of different terms in the Fourier series, with convergence indicated at the term 25.

Fig. 4.27 gives the variation of σ_ϕ determined by FEM. The tendency of the plot is similar to the plot determined by DQM given in Fig. 4.26. Also results are given for a number meshes in the FEM, with convergence indicated at mesh 15×72.

Figs. 4.28 – 4.29 present the comparisons of radial displacement w and circumferential stress σ_η on the outside surface and inside surface determined by DQM and FEM. The agreement between DQM and FEM is good. Fig. 4.29 shows that the circumferential stress is smaller than the meridional stress at the same point because the Young's modulus in circumferential direction is smaller than the Young's modulus in meridional direction in the matrix-cord orthotropic composite material. On the outside surface, the maximum compression stress is at $\phi = 0^o$, and the maximum tensile stress is at $\phi = 90^o$. On the inside surface, the maximum compression stress is at $\phi = 0^o$, and the maximum tensile stress is at $\phi = 150^o$.

The comparisons of displacement and stress variation along the $\phi = 0^o$ cross section determined by DQM and FEM are given in Fig. 4.30– 4.32. There is generally good agreement between the DQM and FEM. The radial displacement and stress components reach a maximum value at $\theta = 0^o$ and almost become zero at $\theta = 60^o$.

Fig. 4.31 shows the variation of meridional stress σ_ϕ on the outside surface and inside surface. The stress on the outside surface almost equals to the stress on the inside surface.

Fig. 4.32 shows the variation of circumferential stress σ_η on the outside surface and inside surface. The circumferential stress on the outside surface is greater than the circumferential stress on the inside surface.

4.6 Conclusions

A DQM solution has been presented for the problems of statics and vibration of orthotropic toroidal shells of variable thickness. The superiority of a set of power series trial functions over harmonic functions in the DQM approach has been demonstrated. Applications have been made for two geometries involving materials of different properties. Sample numerical results obtained using the DQM compare well with those obtained using a finite element solution. The semi-analytical DQM has an undeniable computational advantage over the standard finite element analysis. Thus the DQM approach provides an accurate, efficient means of solving problems for orthotropic toroidal shells of variable thickness.

Chapter 5
CONCLUSIONS

5.1 Thick-Walled Toroidal Shell

A viable solution based on the three-dimensional theory of elasticity has been developed for the vibration and static behavior of a transversely isotropic thick-walled toroidal shell. The solution was based on the semi-analytical approach and the differential quadrature method (DQM) was used to solve the governing equations.

- Two sample problems were considered to validate the DQM solution. Results from the DQM solution agreed well with calculated ET FEM results and with previously published results.
- Natural frequency parameters for three geometric cases found using the DQM and ET FEM are given for steel and M2 toroid. The agreement between the two methods is best for the third geometric case where the three-dimensional elasticity theory is most applicable. The results show that the natural frequencies for the steel toroids as expected are higher than those of M2 material due to the greater stiffness.

- Results for three types of loading for the static problem are given. ET DQM, ET FEM and finite element method using two-dimensional model (ST FEM) are used to get the results. The three sets of results show good agreement for the first two cases where wall thickness ratio is near the thin shell range $h / r < 1 / 10$. For the greater wall thickness results from ET DQM and ET FEM still agree well with each other different from the results from ST FEM. These results indicate that shell theory is not accurate for thick-walled structures.
- The DQM provides a valuable, efficient tool for the analysis of thick-walled toroidal shells which can be used to supplement results obtained using the traditional finite element method.

5.2 Thin-Walled Toroidal Shell

Viable governing equations based on the Sanders thin shell theory has been developed for the vibration and static behavior of orthotropic toroidal shells with variable thickness. The semi-analytical method is again employed and solutions were found using the DQM. A commercial FEM program was used to determine alternative results. Two cases are considered in detail. The first one is a steel toroidal shell complete in the meridional direction. The second one is an orthotropic toroidal shell incomplete in the meridional direction. Conclusions are drawn as following:

- Natural frequency and corresponding mode shapes for both cases are found. Results from the DQM and FEM agree well with each other. For case1, the natural frequency obtained from DQM using harmonic trial functions which are termed as DQM1 leads to slower convergence rated than the natural frequency obtained using the polynomial trial functions which are termed as DQM2.

- Internal pressure is considered in the static problem of case1. Comparing the results for the steel toroidal shell with variable thickness with the steel toroidal shell with constant thickness, it is seen that the changes in stresses is closely proportional to the change in the thickness.
- An extrados pad loading is considered in the static problem of case2. Radial displacements, meridional stress components and circumferential stress components are given. There is a good agreement between two methods. Results show that the circumferential stress is smaller than the meridional stress since the stiffness in the circumferential direction is smaller than it is in the meridional direction. On the $\phi=0^o$ plane, the value of meridional stress on the outside surface almost equals to the values on the inside surface, while the value of circumferential stress on the outside surface is greater than the value on the inside surface.
- According to the results found using DQM and FEM, the set of meshing points used in the DQM is smaller than that for the FEM due to the fact that the DQM is a high order method. Thus the running time of the DQM is shorter. The semi-analytical DQM has an undeniable computational advantage over the standard finite element analysis because it reduces the dimensions of problems by using Fourier series expansions in

circumferential direction. The DQM approach provides an accurate, efficient means of solving problems for orthotropic toroidal shells of variable thickness.

5.3 Suggestions for Further Research

The current study concerns the vibration and static analyses of one-layered transversely isotropic and orthotropic toroidal shells. Future work could be conducted on multi-layer laminated anisotropic composite materials which have applications in aircraft, aerospace vehicles, large space structures, and recreational structures. The linear elasticity theory and shell theory have the limits in their application. Consideration of nonlinear elasticity and shell theories would extend those limits.

In the current study, the free vibration case and the statics case were considered separately. Also vibration of a toroidal shell often occurs with the shell loaded, thus the vibration of prestressed toroidal shell will be considered in the future work.

The thickness of the toroidal shell considered here is a function of the meridional angle only, and the cross-section is given by a circle or a circular arc. In future work, consideration could be given to other thickness variations, and to non-circular arcs more closely representing tire geometries.

References

ADINA, User's manual (2000), v. 7.4, ADINA Engineering, Inc., Worcester, MA, USA.

Balderes T and Armenakas AE (1973), "Free vibrations of ring-stiffened toroidal shells," *AIAA J.,* **11**, 1637 – 1644.

Bathe KJ (1996), *Finite Element Procedures*, Prentice Hall, Englewood Cliffs.

Bellman RE and Casti J (1972), "Differential quadrature and long-term integration," *J. Math. Anal. Appl.*, **34**, 235 – 238.

Bert CW and Malik M (1996a), "Free vibration analysis of thin cylindrical shells by the differential quadrature method," *ASME J. Press. Ves. Tech.*, **118**, 1 – 12.

Bert CW and Malik M (1996b), "Differential quadrature method in computational mechanics: a review, " *ASME, Appl. Mech.*, **49**, no 1.

Bert CW, Jang SK, and Striz AG (1989), "Nonlinear bending analysis of orthotropic rectangular plated by the method of differential quadrature," *Comput. Mech.*, **5**, 217 – 226.

Bert CW, Jang SK, and Striz AG (1988), "Two new approximate methods for analyzing free vibration of structural components," *AIAA J.*, **26**, 612 – 618.

Bielski J (1992), "Influence of geometrical imperfection on buckling pressure and post-buckling behavior of elastic toroidal shells," *Mech. Struct. & Mach.*, **20**, 145 – 154.

Budiansky B (1968), "Notes on nonlinear shell theory," *J. Appl. Mech.* **40**, 393 – 401.

Chandrashekhara K and Nanjunda Rao KS (1997), "Approximate elasticity solution for a long and thick laminated circular cylindrical shell of revolution," *Int. J. Solids Struct.*, **34**, 1327 – 1341.

Darnell I., Hulbert GM, and Mousseau CW (1997), "An efficient three-dimensional tire model for vehicle dynamics simulations," *Mech. Stuct. & Mach.*, **25**, 1 – 9.

Eringen AC and Suhubi ES (1975), *Elastodynamics*, Academic Press, New York.

Gaidaichuk VV, Gotsulyak EA, and Guyyaev VI (1978), "Nonlinear stability of toroidal shells of variable thickness acted upon by external pressure," *Mech. of Sol.*, **13**(3), 95 – 100.

Gall R, Tabaddor F, Robbins D, Majors P, Sheperd W, and Johnson S (1995), " Some notes on the finite element analysis of tires," *Tire Sci. Technol.*, **23**, 175 – 188.

Göhner O (1930), "Schubspannungsverteilung im Querschnitt einer Schraubenfeder," *Ing. Arch.*, **1**, 619 – 664.

Grigorenko YaM, Dyak II, and Makar VM (1998), "Three-dimensional dynamic elasticity-theory problem for anisotropic bodies," *Int. Appl. Mech.* **34**(5), 424 – 430.

Grigorenko YaM, Vlaikov GG, and Shevchenko SN (1995), "Stress state of hollow orthotropic noncircular cylinders under a local load," *Int. Appl. Mech.*, **31**(5), 331 – 337.

Hu LC and Hu CR (1974), "Identification of rate constenls by differential quadrature in partly measurable compartmental models," *Math. Biosci.*, **21**, 71 – 76.

Huang DW, Redekop D, and Xu B (1995), "Natural frequency and mode shapes of curved pipes," *Comput. & Struct.*, **63**(3), 465 – 473.

Jang SK, Bert CW, and Striz AG (1989), "Application of differential quadrature to static analysis of structural components," *Int. J. Numer. Meth. Eng.*, **28**(3), 561 – 577.

Jiang W and Redekop D (2002a), "Static and vibration analysis of orthotropic toroidal shells of variable thickness," Submitted for publication.

Jiang W and Redekop D (2002b), "Polar axisymmetric vibration of a hollow toroid using the differential quadrature method." *J. Sound & Vibr.*, **251**(4), 761 – 765.

Jiang W and Redekop D (2002c), "Analysis of transversely isotropic hollow toroids using the semi-analytical DQM," *Struct. Eng. & Mech.*, **13**(1), 103 – 116.

Kornecki A (1963), "Stress distribution in a pressurized thick-walled toroidal shell – a three dimensional analysis," *College of Aeronautics*, Cranfield, England. Note 137.

Kim DO, Noor AK, and Tanner JA (1990), "Modeling and analysis of the space shuttle nose-gear tire with semi-analytical finite element," *NASA TP2977*, 33 pages.

Kosawada T, Suzuki K, and Takahashi S (1986), "Free vibrations of thick toroidal shells," *Bull JSME*, **29**, 3036 – 3042.

Lang HA (1989), "The theory of toroidal elasticity," *Int. J. of Struct. Mech. and Materials Sci.*, **26**, 289 – 357.

Leissa AW (1973), *Vibration of Shells*, NASA SP-288.

Leung AYT and Kwok NTC (1994), "Free vibration analysis of a toroidal shell," *Thin-Walled Struct.*, **18**, 317 – 332.

Malik M and Bert CW (1994), "Differential quadrature solutions for steady-state incompressible and compressible lubrication problems," *J. Tribol-Trans. ASME.*, **116**(2), 296 – 302.

McGill DJ and Lenzen KH (1967), "Polar axisymmetric free oscillations of thick hollowed tori," *SIAM Journal*, **15**, 678 – 692.

Mingle JO (1977), "The method of differential quadrature for transient nonlinear diffusion," *J. Math. Anal. Appl.*, **60**, 559 – 569.

Naadimuthu G, Bellman RE, Wang KM, and Lee ES (1984), "Differential quadrature and partial differential equations: some numerical results," *J. Math. Anal. Appl.*, **98**, 220 – 233.

Naboulsi SK, Palazotto AN, and Greer JM (2000), "Static-dynamic analyses of toroidal shells," *J. Aero. Eng., ASCE*, **13**, 110 – 121.

Noor AK and Burton WS (1990), "Assessment of computational models for multi-layered composite shells," *ASME Appl. Mech. Review*, **43**, 67 – 96.

Noor AK and Peters JM (1987), "Vibration analysis of laminated anisotropic shells of revolution," *Comp. Mech. Appl. Mech. Eng.*, **61**, 277 – 301.

Redekop D and Xu B (2000), "Dynamic buckling of toroidal shells subject to impulsive local loading," *Proc. ICPVT-9 Conference, Sydney*, **1**, 685 – 692.

Redekop D and Xu B (1999), "Vibration analysis of toroidal panels using the differential quadrature method," *Thin-Walled struct.*, **34**, 217 – 231.

Redekop D (1994), "Natural frequencies of a short curved pipe," *Trans CSME*, **18**, 35 –45.

Redekop D (1992), "A displacement solution in toroidal elasticity," *Int. J. Pres. Vess. & Piping*, **51**, 189 – 209.

Redekop D (1990), "Ram bending of a curved pipe in the elastic range," *Int. J. Pres. Ves. Piping*, **44**, 191 – 206.

Rossettos JN and Sanders JL (1965), "Toroidal shell under internal pressure in the transition range," *AIAA Journal*, **3**(10), 1901 – 1909.

Sanders JL (1959), "An improved first-approximation theory for thin shells," *NASA TR-24*.

Sanders JL and Liepins AA (1963), "Toroidal membrane under internal pressure," *AIAA Journal*, **1**(9), 2105 – 2110.

Shu C (2000), *Differential Quadrature and Its Application in Engineering*, Springer, Berlin.

Shu C and Xue H (1997), "Explicit computation of weighting coefficients in the harmonic differential quadrature," *J. Sound & Vibr.*, *204*(3), 549 – 555.

Shu C and Richard BE (1992), "Application of generalized differential quadrature to solve two-dimensional incompressible Navier-Stokes equations," *Int. J. Numer. Meth. in Fluids*, **15**, 791 – 798.

Sobel LH and Flügge W (1967), "Stability of toroidal shells under uniform external pressure," *AIAA Journal*, **5**(3), 425 – 431.

Thomas K (1980), "The effects of geometric irregularities on the design analysis of thin-walled piping elbows," *ASME PVP-43,* **80-C2**, 1 – 9.

Thompson G and Spence J (1983), "The influence of flanged end constraints on smooth curved tubes under in-plane bending," *Int. J. Pres. Ves. & Piping*, **13**, 65 – 83.

Walter JD (1982), "Cord reinforced rubber," *Mechanics of Pneumatic Tires, SK Clark ed., US Dept. of Transp.*, 123 – 202.

Wang KM (1982), "Solving the model of isothermal reactor -with axial mixing by the differential quadrature method," *Int. J. Numer. Meth. Eng.*, **18**, 111 – 118.

Whatham JF (1986), "Pipe bend analysis by thin shell theory," *J. Appl. Mech.*, **53**, 173 – 180.

Xi ZC, Yam LH, and Leung TP (1999), "Free vibration of laminated composite shell of revolution: effects of shear non-linearity," *Int. J. of Mech. Sci.*, **41**, 649 – 661.

Xia ZH and Zhang W (1987), "The general solution of thin-walled toroidal shells with various boundary conditions and subjected to arbitrary loadings," *Ingenieur-Archiv*, **57**, 166–174.

Zhang F and Redekop D (1991), "Surface loading of a thin-walled toroidal shell," *Computers & Structures*, **43**, 1019 – 1028.

Zhang Y, Palmer T, and Farahani A (1997), "A finite element tire model and vibration analysis: a new approach," *Tire Sci. Technol.*, **26**,149 – 172.

Zhu Y and Redekop D (1995), "Band loading of a thick-walled toroidal shell," *Int. J. Pres. Vess. & Piping*, **61**, 99 – 109.

Zhu Y (1994), *The Development of Displacement Based Toroidal Elasticity*, Thesis of Mechanical Engineering Department, University of Ottawa.

Table 3.1: Thin shell lowest natural frequency parameter β^2

	Shell theory	FEM (ET)		DQM (ET)	
Geometry	Balderes(1973)	Mesh	Value	Mesh	Value
$R/\bar{r} = 10$ $h/\bar{r} = 0.01$	0.0148	2×400	0.0149	6×18	0.0149
		2×800	0.0149	8×24	0.0149
		4×800	0.0149	12×36	0.0148
$R/\bar{r} = 10$ $h/\bar{r} = 0.02$	0.0315	2×400	0.0322	6×18	0.0319
		2×800	0.0322	8×24	0.0319
		4×800	0.0322	12×36	0.0318
$R/\bar{r} = 5$ $h/\bar{r} = 0.02$	0.0150	2×500	0.0158	6×18	0.0152
		2×1000	0.0153	8×24	0.0152
		4×1000	0.0153	12×36	0.0152

Table 3.2: Shell geometric cases

Case	R(mm)	$\bar{r}$ (mm)	h(mm)
1	600	100	10.6
2	600	100	18.0
3	600	100	30.0

Table 3.3: Convergence study – parameter $\hat{\rho}\omega^2$ ($\times 10^9$)

Case	FEM (ET)			DQM (ET)		
	Mesh	Steel	M2	Mesh	Steel	M2
1	$1\times12\times24$	23.28	2.133	4×24	22.30	2.101
	$1\times24\times24$	22.99	2.112	6×36	17.23	1.901
	$1\times24\times48$	22.99	2.112	12×48	17.23	1.900
2	$1\times12\times24$	32.18	2.844	4×24	32.28	2.924
	$1\times24\times24$	32.08	2.836	6×36	28.97	2.712
	$1\times24\times48$	32.08	2.836	12×48	28.94	2.711
3	$1\times12\times24$	37.91	3.523	4×16	37.88	4.910
	$2\times24\times24$	37.84	3.509	8×32	36.32	3.454
	$2\times24\times48$	37.84	3.509	12×48	36.32	3.454

Tables 3.4: Results for unit loading on steel shell

		FEM (ET)			DQM (ET)			FEM (ST)	
Case	Load at	u(mm)	σ_ϕ(MPa)	σ_θ(MPa)	u(mm)	σ_ϕ(MPa)	σ_θ(MPa)	u(mm)	σ_ϕ(MPa)
1	Extrados	-0.0787	-33.86	-35.98	-0.0770	-31.92	-35.00	-0.0749	-25.78
2		-0.0346	-17.31	-16.30	-0.0320	-16.45	-16.04	-0.0320	-13.16
3		-0.0143	-8.21	-7.81	-0.0144	-7.79	-7.72	-0.0126	-5.77
1	Crown	-0.0692	-40.93	-21.77	-0.0652	-40.18	-23.51	-0.0684	-35.69
2		-0.0227	-18.54	-9.19	-0.0214	-17.17	-9.21	-0.0212	-16.35
3		-0.0069	-7.99	-3.89	-0.0070	-7.67	-3.75	-0.0062	-6.95
1	Intrados	-0.1000	-46.35	-14.93	-0.1180	-50.31	-23.71	-0.0974	-35.47
2		-0.0342	-20.32	-7.30	-0.0373	-20.90	-9.89	-0.0326	-14.80
3		-0.0132	-8.87	-3.85	-0.0139	-8.90	-4.59	-0.0121	-5.78

Tables 3.5: Results for unit loading on M2 shell

		FEM (ET)			DQM (ET)		
Case	Load at	u(mm)	σ_ϕ(MPa)	σ_θ(MPa)	u(mm)	σ_ϕ(MPa)	σ_θ(MPa)
1	Extrados	-0.553	-10.15	-59.13	-0.609	-10.45	-57.16
2		-0.292	-7.30	-37.04	-0.318	-7.38	-35.69
3		-0.148	-4.54	-21.76	-0.160	-4.56	-21.21
1	Crown	-0.821	-24.00	-58.73	-0.849	-21.98	-58.15
2		-0.288	-10.98	-26.02	-0.300	-10.30	-25.55
3		-0.109	-5.12	-11.54	-0.109	-4.88	-11.30
1	Intrados	-1.219	-20.80	-57.04	-1.366	-21.84	-64.37
2		-0.405	-9.93	-29.41	-0.438	-10.04	-31.81
3		-0.149	-4.68	-13.92	-0.156	-4.60	-14.99

Table 4.1: Convergence of natural frequency ω (rad/s) for case 1

m	FEM		DQM1		DQM2	
	Mesh size	ω	Mesh size	ω	Mesh size	ω
0A	6×24	0.9088e3	8	1.1722e3	16	1.0572e3
2A		1.0788e3		1.2889e3		1.2702e3
2S		1.1706e3		1.4898e3		1.3218e3
3A		2.1213e3		1.7353e3		1.7956e3
3S		2.1454e3		2.6358e3		2.5385e3
1A		2.9806e3		3.1935e3		3.0977e3
0A	12×48	0.9390e3	16	0.9275e3	24	0.9269e3
2A		1.0959e3		1.1084e3		1.1124e3
2S		1.1916e3		1.2227e3		1.2258e3
3A		2.1421e3		2.1682e3		2.1678e3
3S		2.1660e3		2.1824e3		2.1801e3
1A		3.0109e3		3.0387e3		3.0403e3
0A	12×96	0.9391e3	24	0.9275e3	36	0.9281e3
2A		1.0958e3		1.1084e3		1.1102e3
2S		1.1915e3		1.2227e3		1.2267e3
3A		2.1415e3		2.1682e3		2.1708e3
3S		2.1653e3		2.1824e3		2.1769e3
1A		3.0108e3		3.0387e3		3.0415e3
0A	24×96	0.9385e3	48	0.9275e3	44	0.9281e3
2A		1.0952e3		1.1084e3		1.1095e3
2S		1.1909e3		1.2227e3		1.2274e3
3A		2.1394e3		2.1682e3		2.1708e3
3S		2.1634e3		2.1824e3		2.1769e3
1A		2.9641e7		3.0387e3		3.0415e3

Table 4.2: Convergence of statics solution for case 1

FEM				DQM			
Mesh	$w(meters)$	$\sigma_\theta(MPa)$	$\sigma_\eta(MPa)$	*Mesh*	$w(meters)$	$\sigma_\theta(MPa)$	$\sigma_\eta(MPa)$
36	3.2896e-5	25.610	14.406	12	3.3715e-5	25.666	14.596
72	3.2928e-5	25.668	14.434	24	3.3395e-5	25.713	14.535
96	3.2932e-5	25.680	14.439	48	3.3357e-5	25.713	14.505
192	3.2937e-5	25.698	14.446	72	3.3359e-5	25.713	14.538

Table 4.3: Convergence of natural frequency ω (rad/s) for case 2

m	FEM		DQM	
	Mesh size	ω	Mesh size	ω
0A	3×24	1.2729e2	8	1.7089e2
1A		1.5841e2		1.8895e2
2A		2.4063e2		2.6029e2
1S		3.5933e2		3.0568e2
3A		3.5935e2		3.8069e2
0S		3.7467e2		3.6999e2
0A	6×48	1.2309e2	16	1.1697e2
1A		1.4740e2		1.3424e2
2A		2.2304e2		1.9784e2
1S		3.3182e2		3.0063e2
3A		3.3453e2		3.0621e2
0S		3.4524e2		3.6589e2
0A	12×96	1.2250e2	24	1.1697e2
1A		1.4501e2		1.3424e2
2A		2.1848e2		1.9784e2
1S		3.2493e2		3.0063e2
3A		3.2898e2		3.0621e2
0S		3.4130e2		3.6589e2
0A	24×96	1.2155e2	48	1.1697e2
1A		1.4482e2		1.3424e2
2A		2.1839e2		1.9784e2
1S		3.2433e2		3.0063e2
3A		3.2889e2		3.0621e2
0S		3.4032e2		3.6589e2

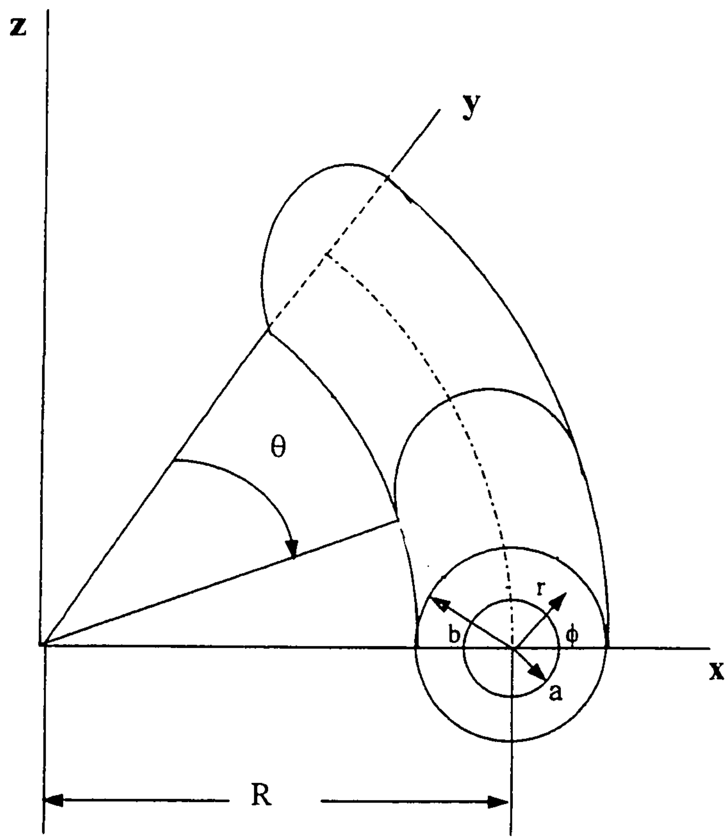


Fig. 3.1: Geometry and coordinate system

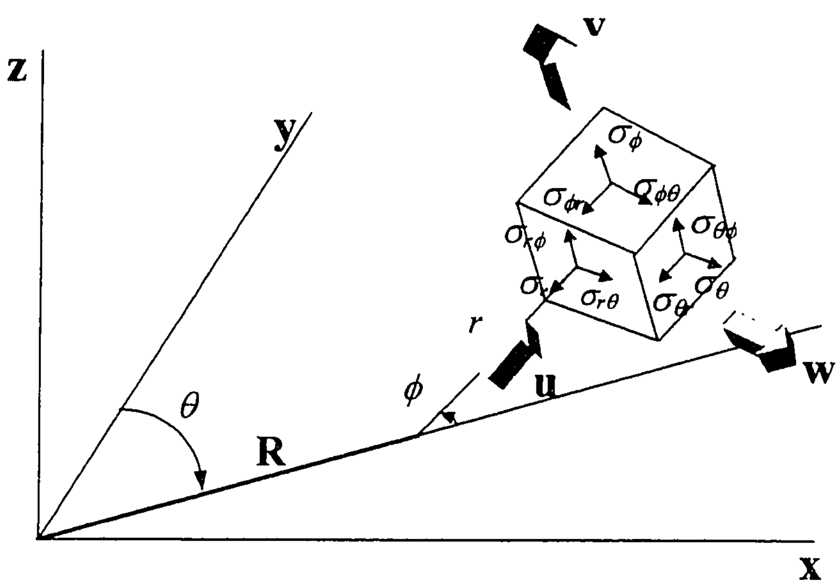


Fig. 3.2: Displacement and stress components

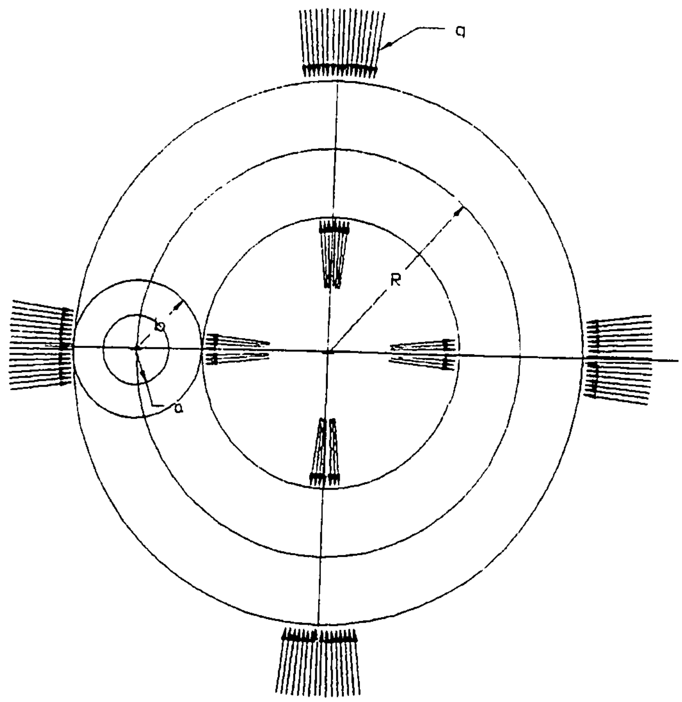


Fig. 3.3: Band loading of hollow toroid

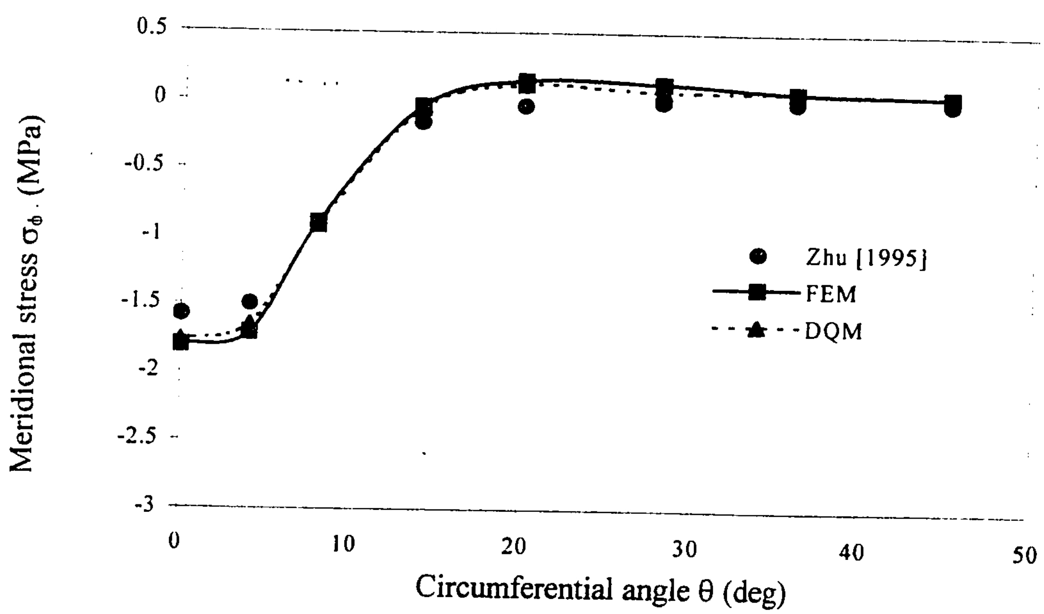


Fig. 3.4: Meridional stresses for band loading

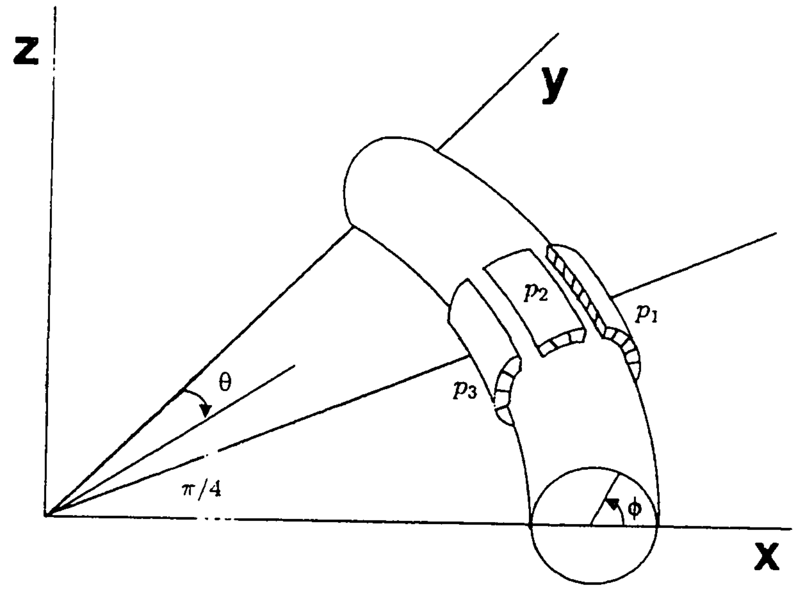


Fig. 3.5: Local loading cases

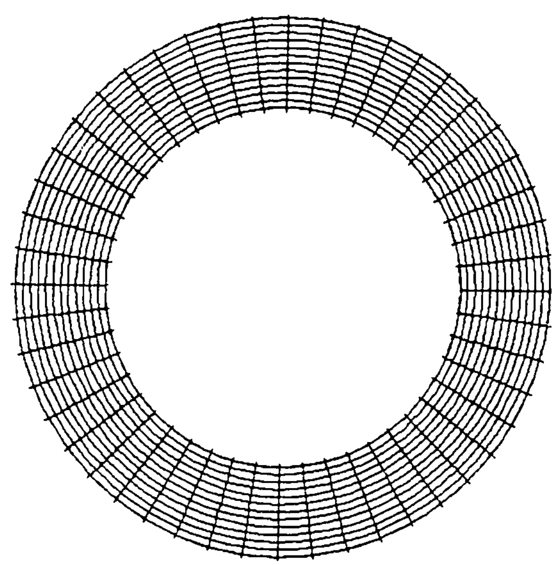


Fig. 3.6: DQM mesh

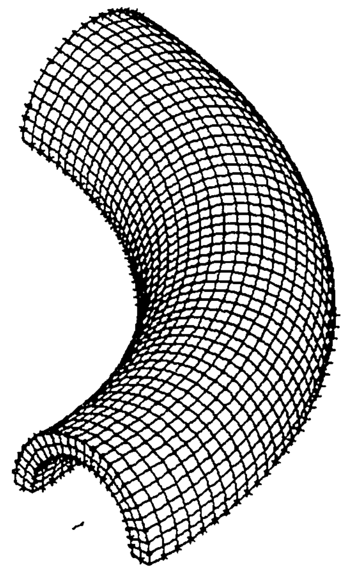


Fig. 3.7: Elasticity theory FEM mesh

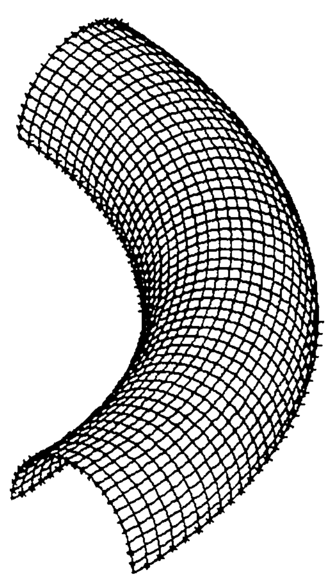


Fig. 3.8: Shell theory FEM mesh

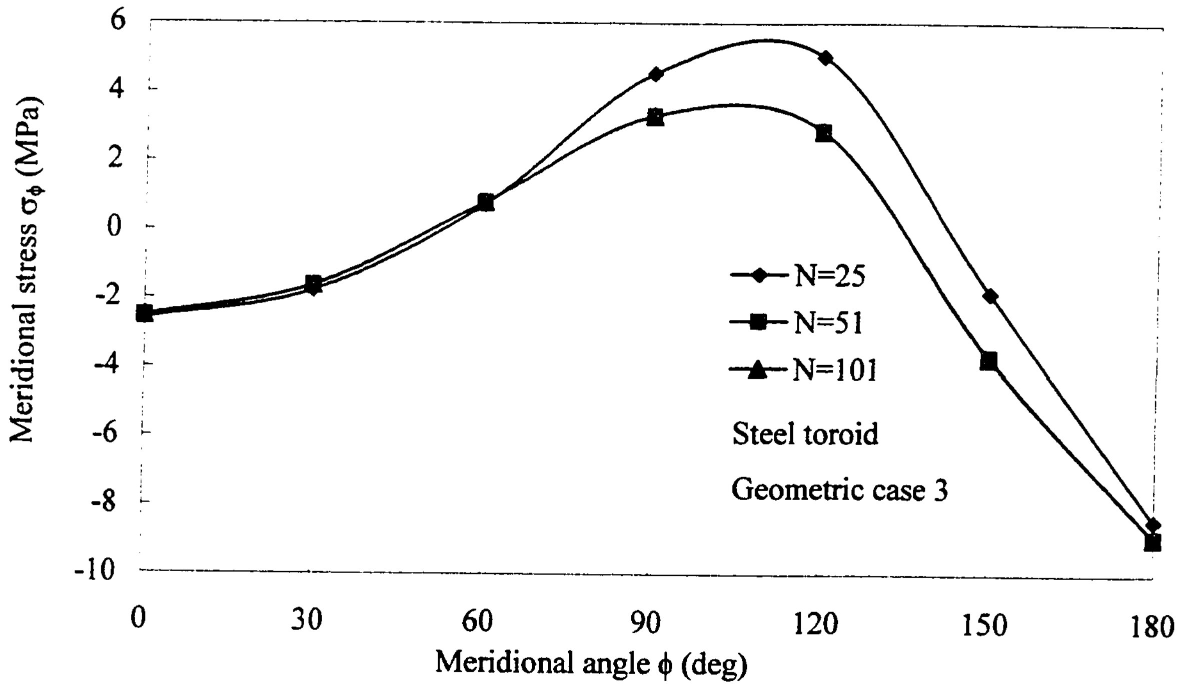


Fig. 3.9: Convergence of Fourier series for p_3 loading case

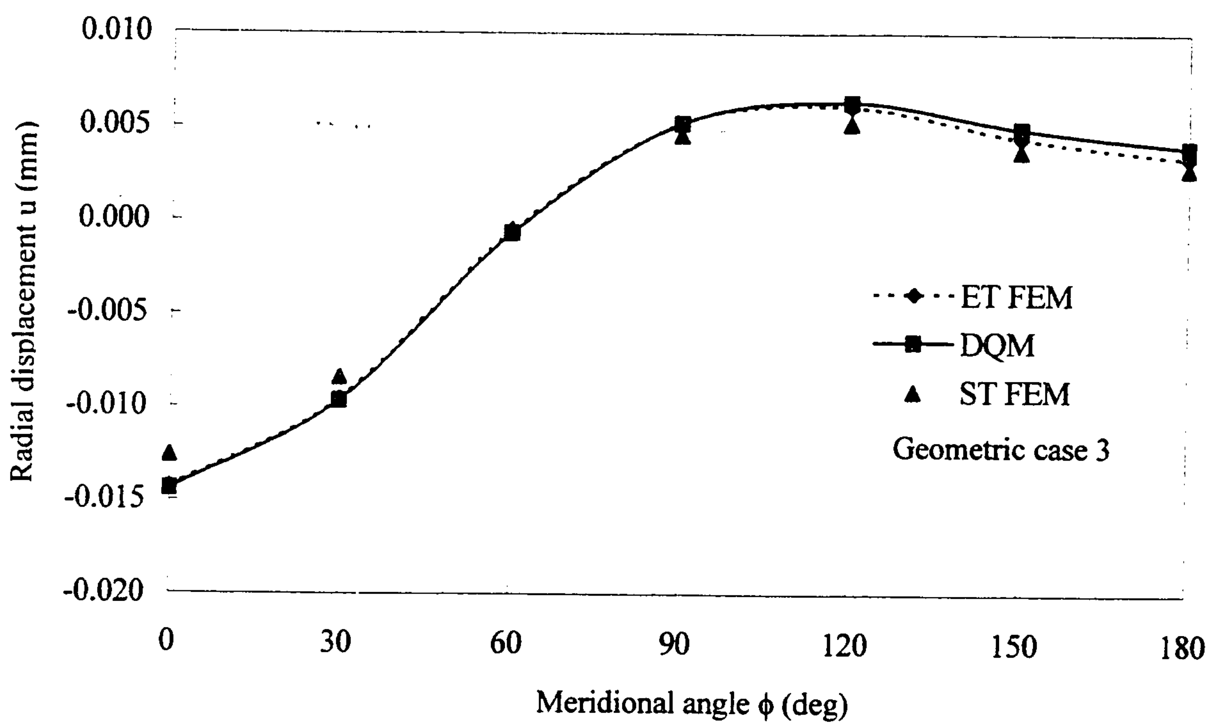


Fig. 3.10: Radial displacement u for p_1 loading on plane θ=45^o – steel toroid

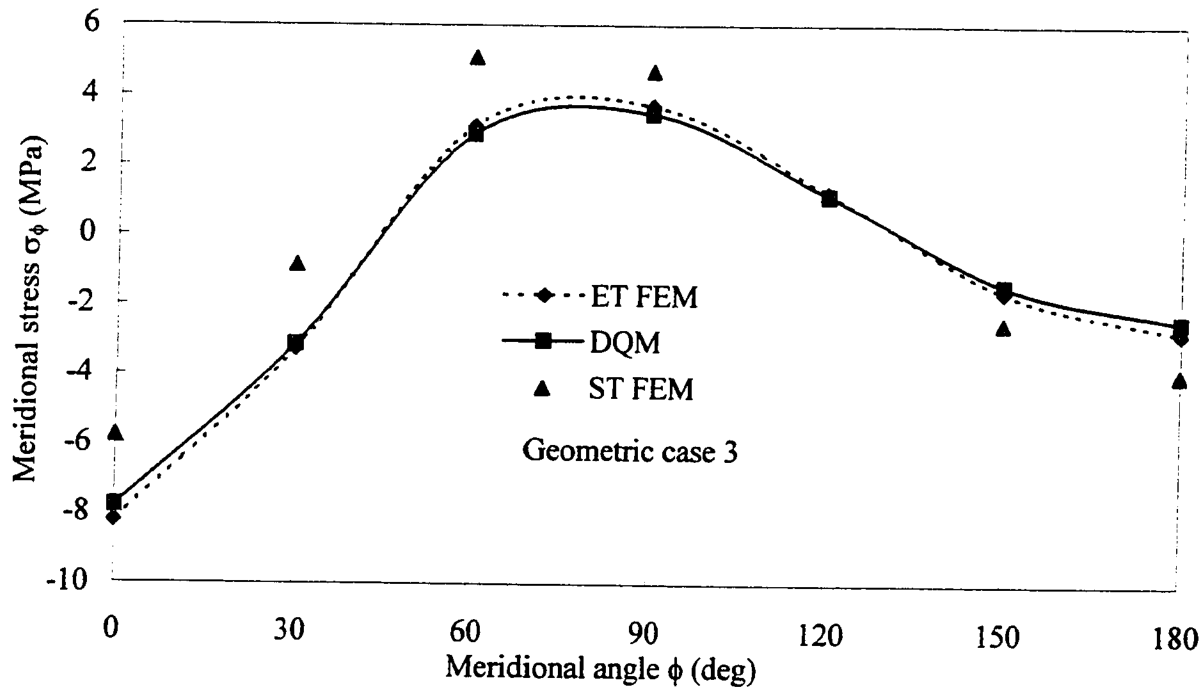


Fig. 3.11: Meridional stress σ_ϕ for p_1 loading on plane θ=45° – steel toroid

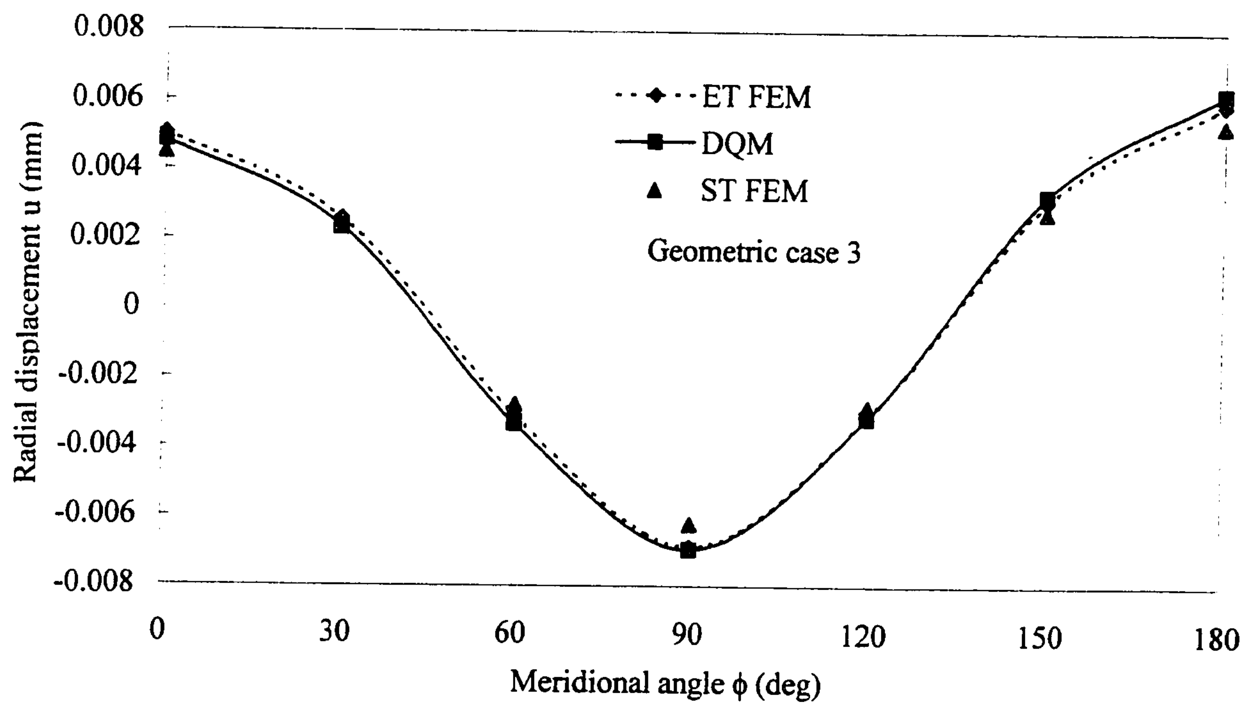


Fig. 3.12: Radial displacement u for p_2 loading on plane θ=45° – steel toroid

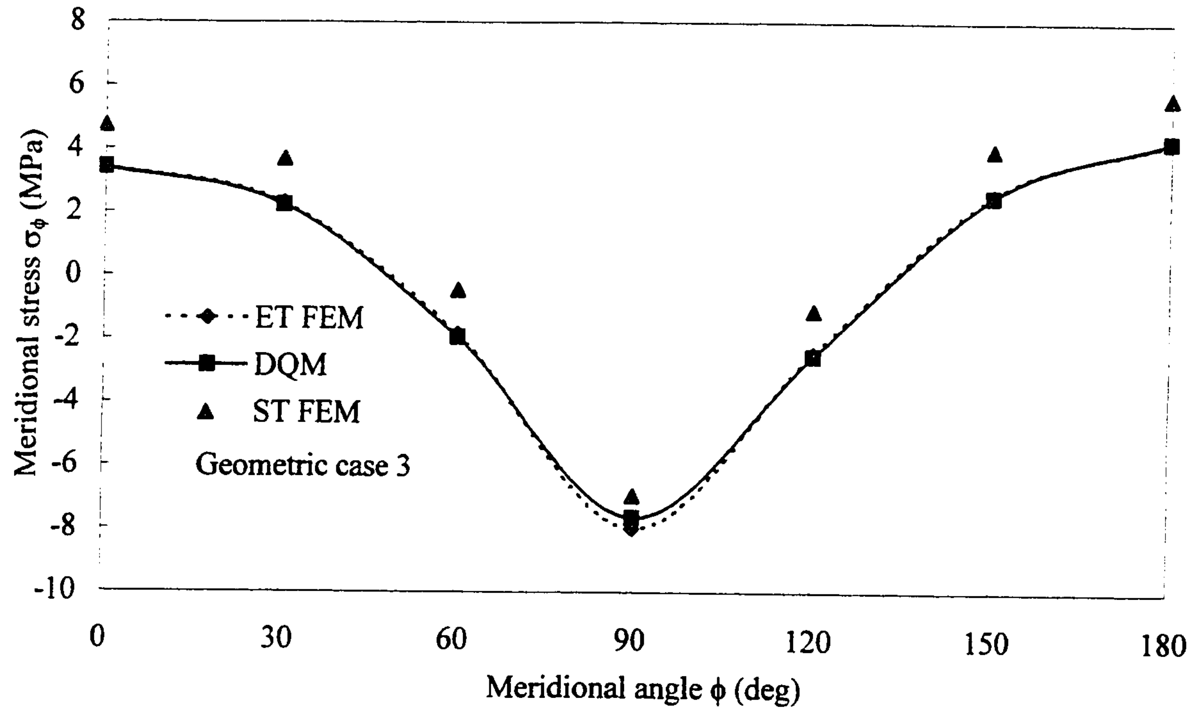


Fig. 3.13: Meridional stress σ_ϕ for p_2 loading on plane θ=45° – steel toroid

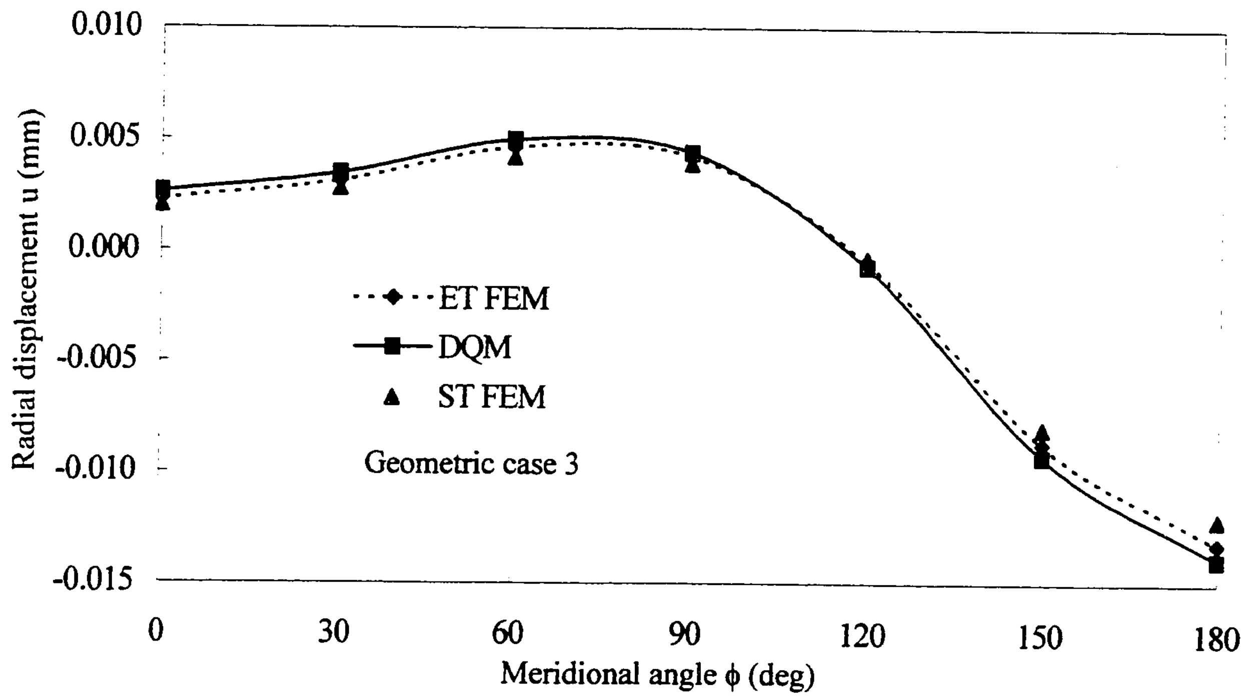


Fig. 3.14: Radial displacement u for p_3 loading on plane θ=45° – steel toroid

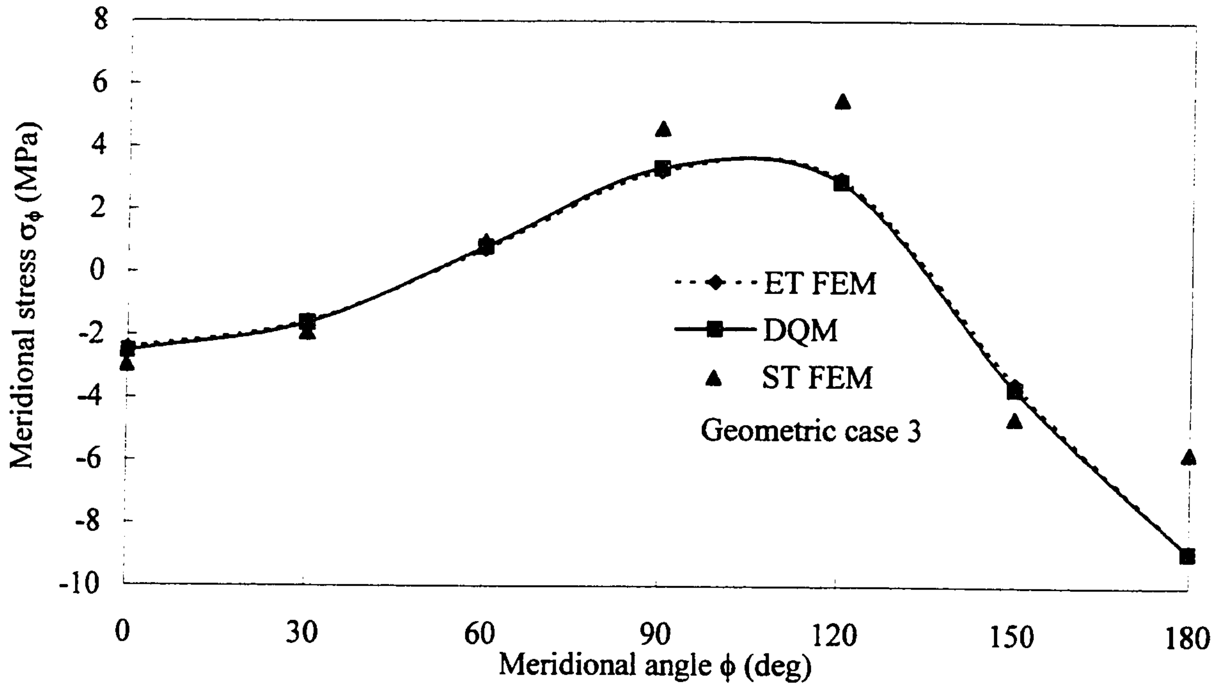


Fig. 3.15: Meridional stress σ_ϕ for p_3 loading on plane θ=45^o – steel toroid

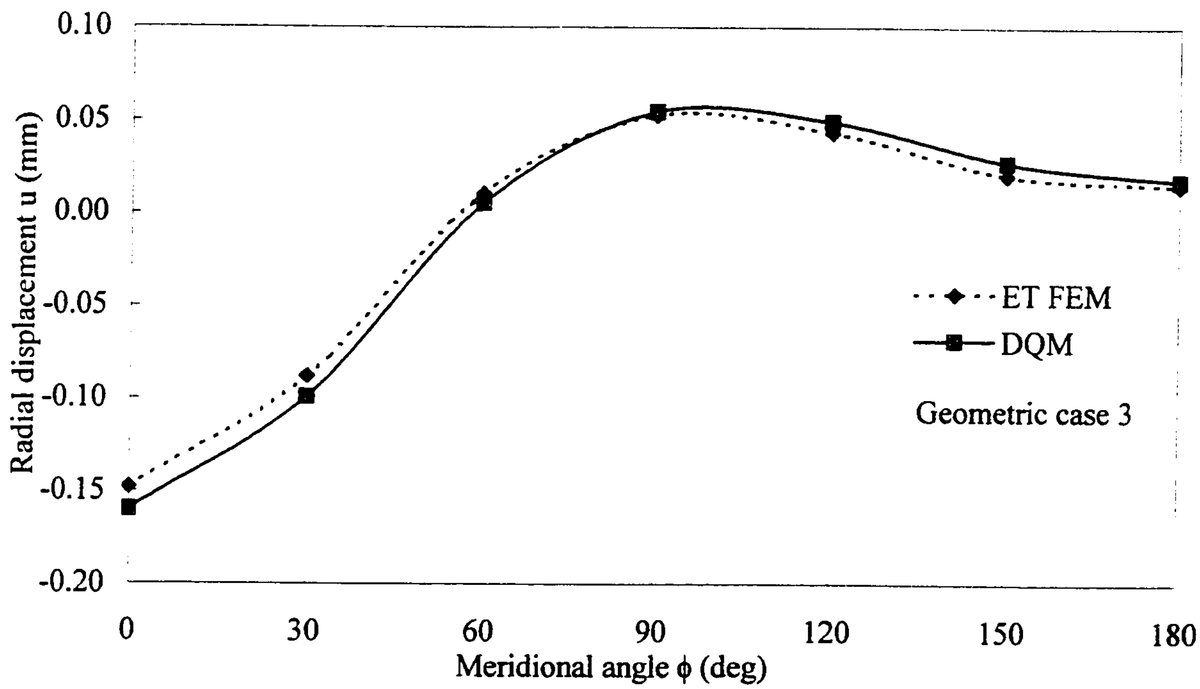


Fig. 3.16: Radial displacement u for p_1 loading on plane θ=45^o – M2 toroid

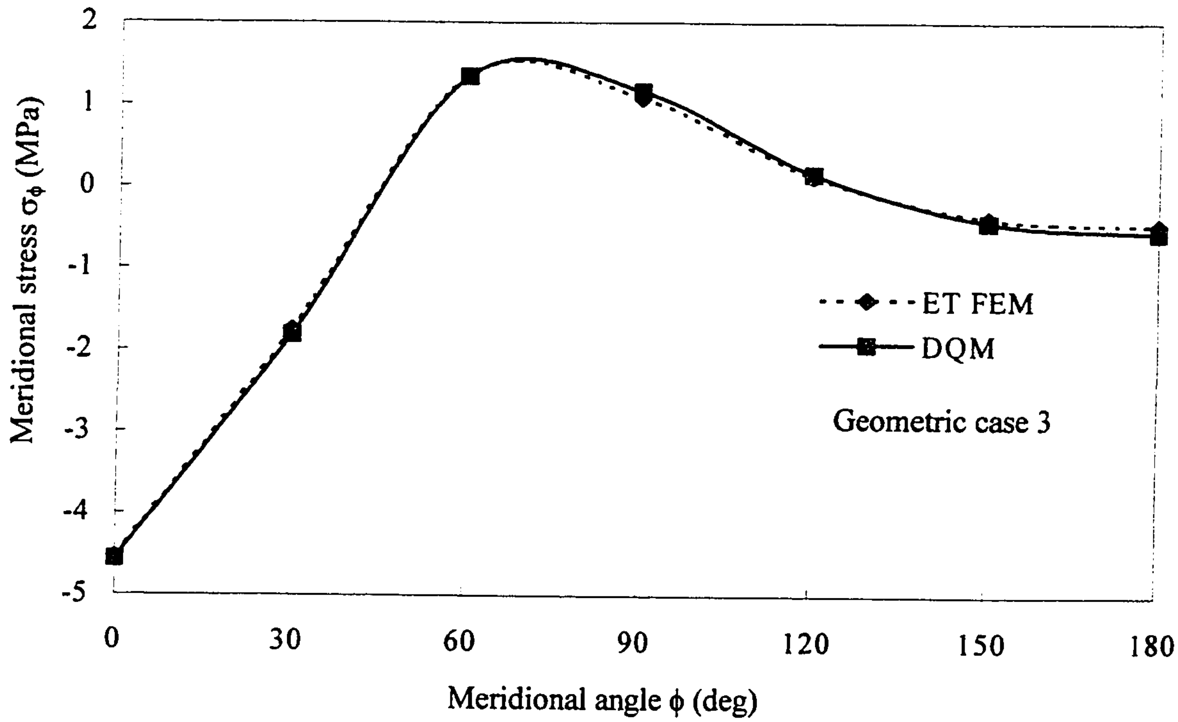


Fig. 3.17: Meridional stress σ_ϕ for p_1 loading on plane θ=45^o – M2 toroid

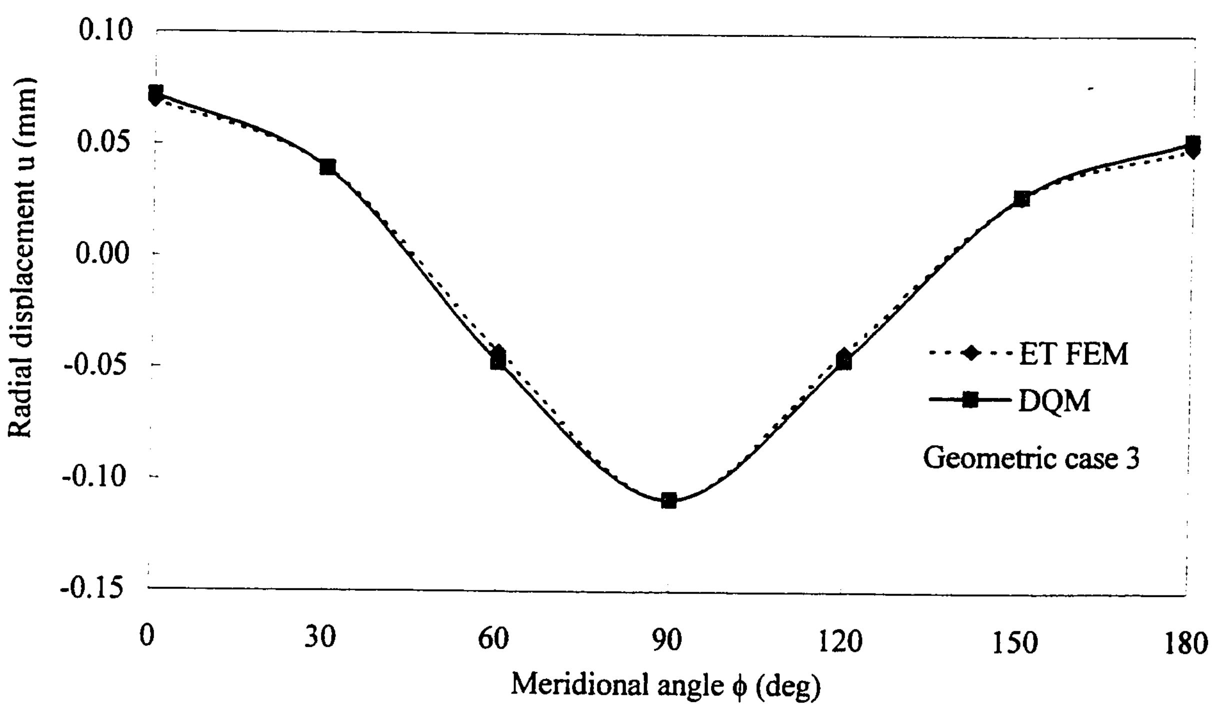


Fig. 3.18: Radial displacement u for p_2 loading on plane θ=45^o – M2 toroid

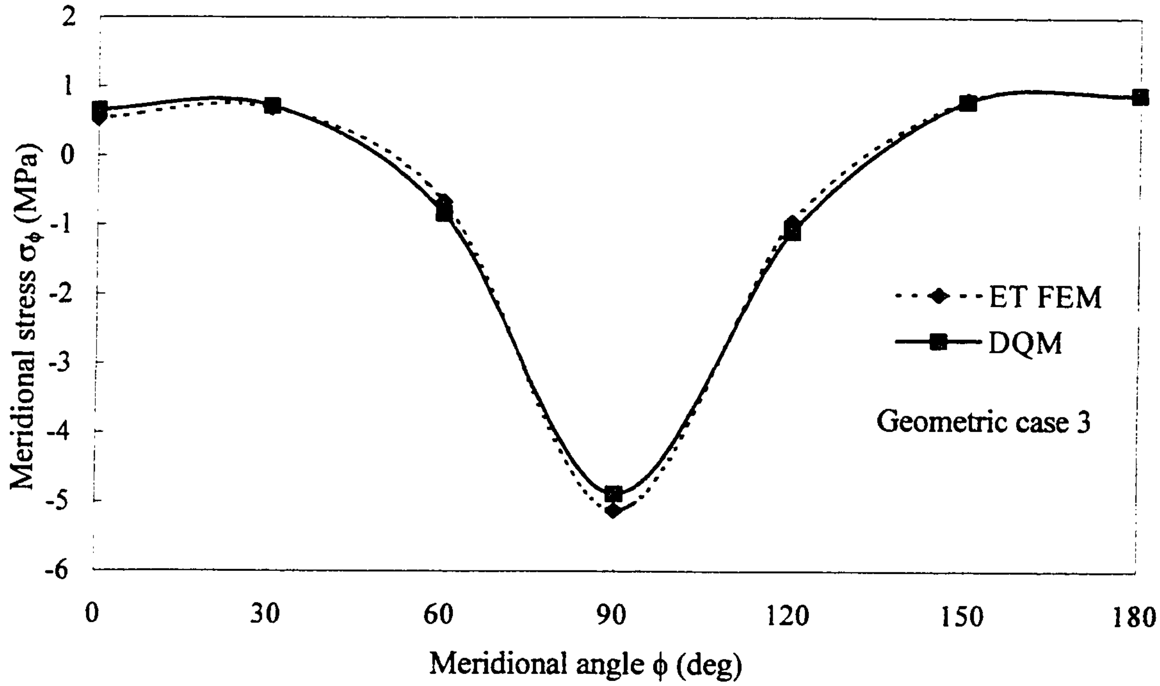


Fig. 3.19: Meridional stress σ_ϕ for p_2 loading on plane θ=45° – M2 toroid

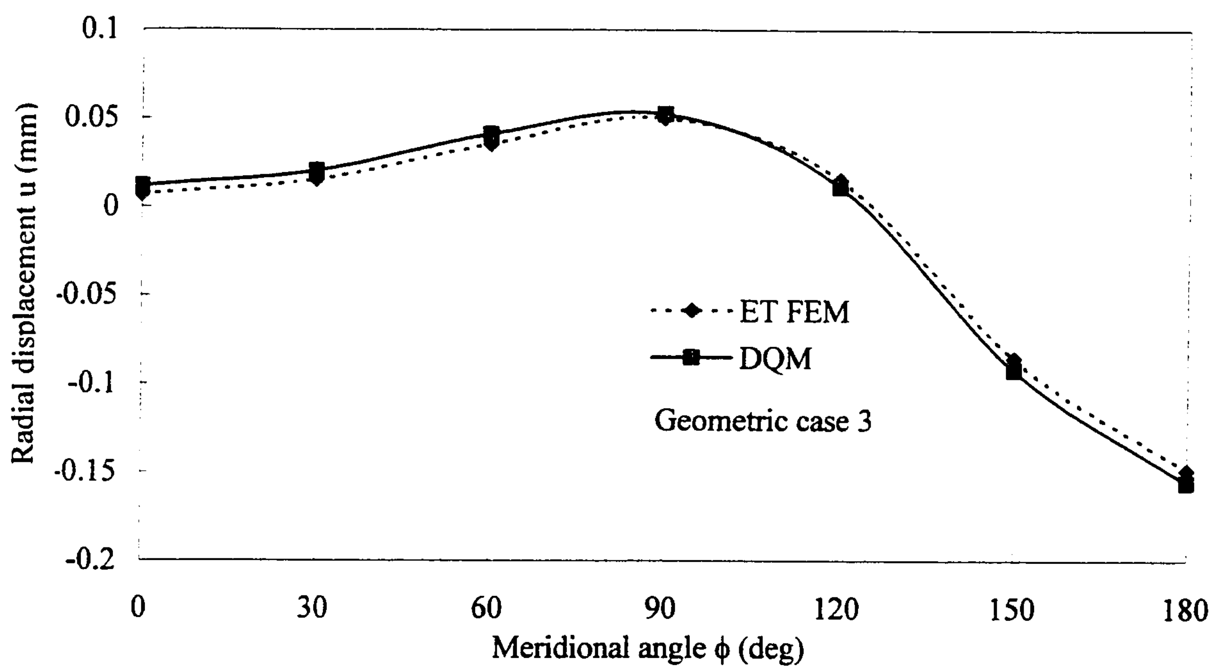


Fig. 3.20: Radial displacement u for p_3 loading on plane θ=45° – M2 toroid

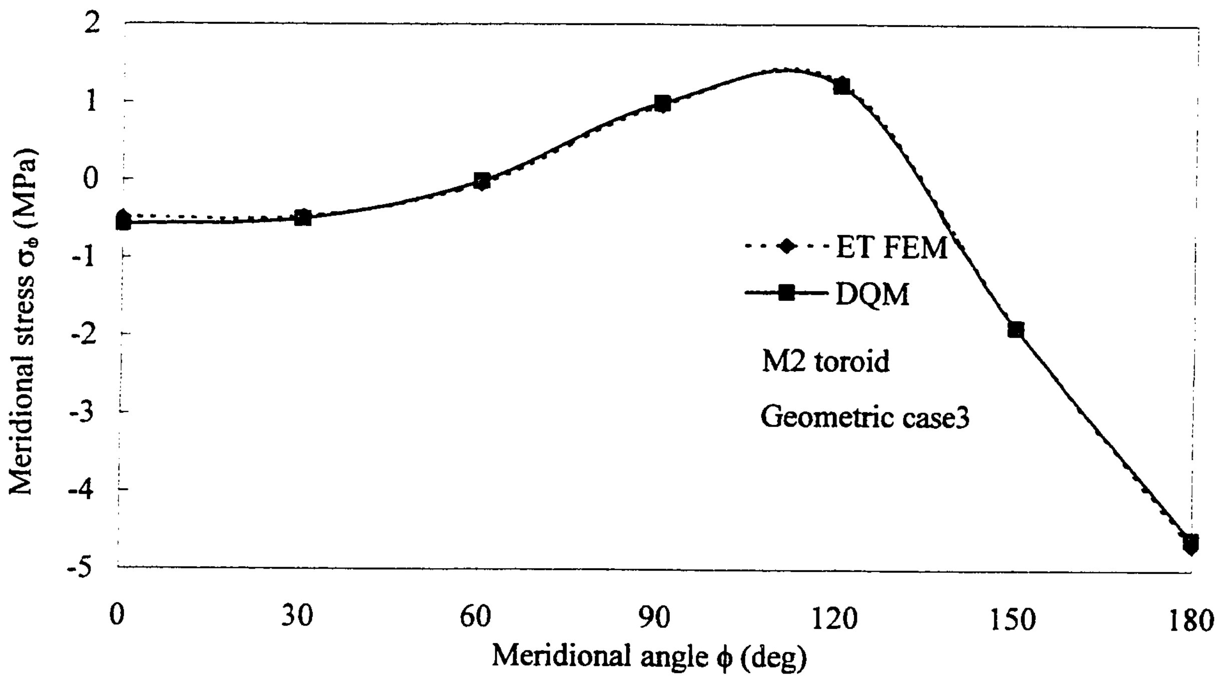


Fig. 3.21: Meridional stress σ_ϕ for p_3 loading on plane θ=45^o

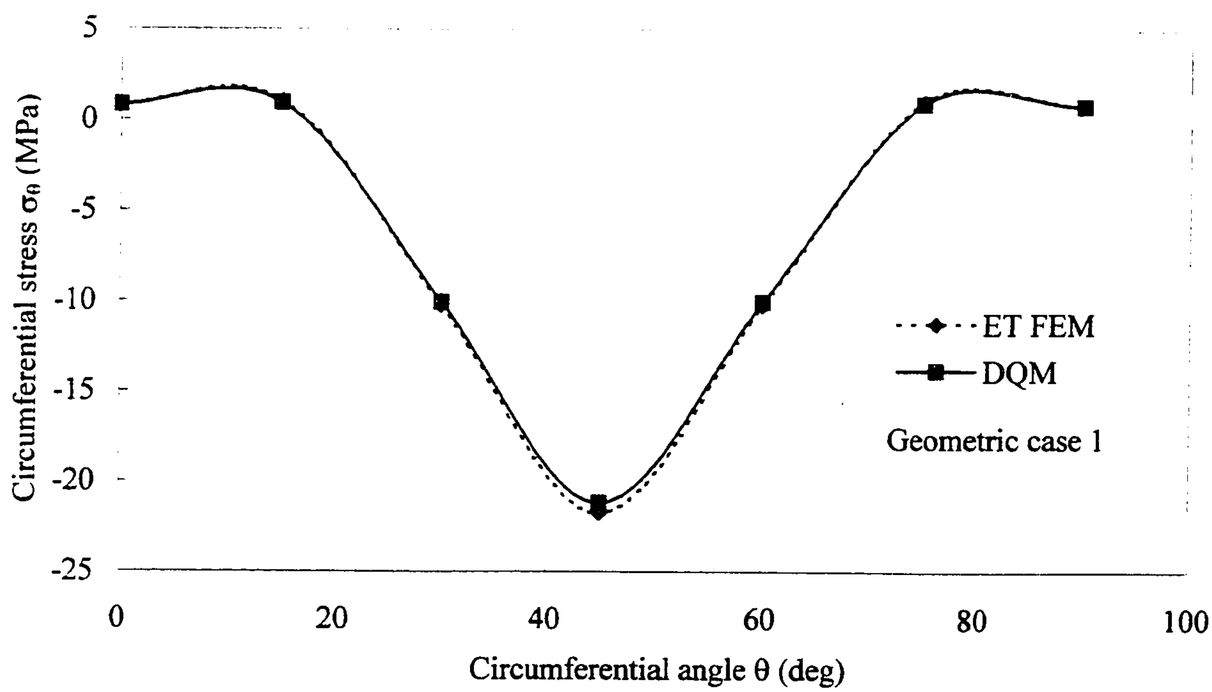


Fig. 3.22: Circumferential stress σ_θ for p_1 loading on plane ϕ=0^o – M2 toroid

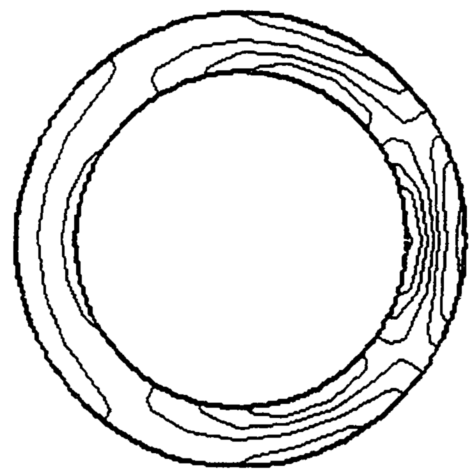


Fig. 3.23: Meridional stress contour line for p_1 loading – M2 toroid

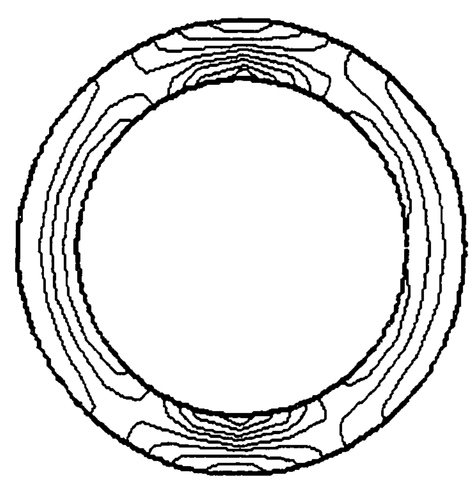


Fig. 3.24: Meridional stress contour line for p_2 loading – M2 toroid

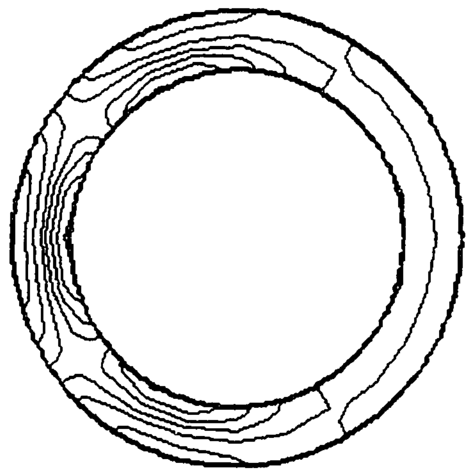


Fig. 3.25: Meridional stress contour line for p_3 loading – M2 toroid

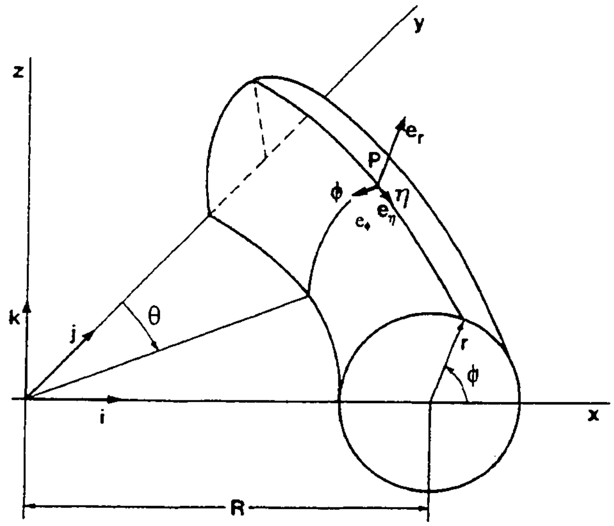


Fig. 4.1: Toroidal coordinate system

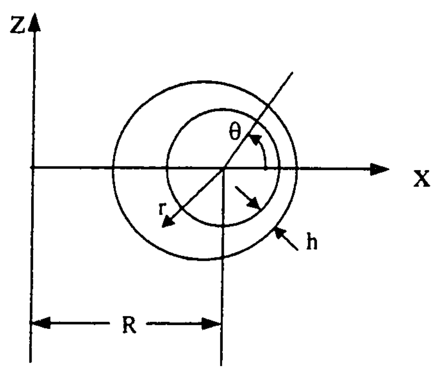


Fig. 4.2: Shell complete in meridional direction

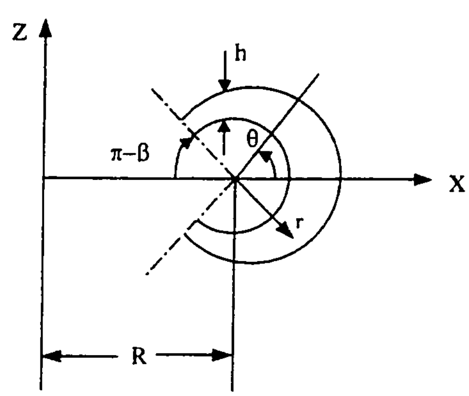


Fig. 4.3: Shell incomplete in meridional direction

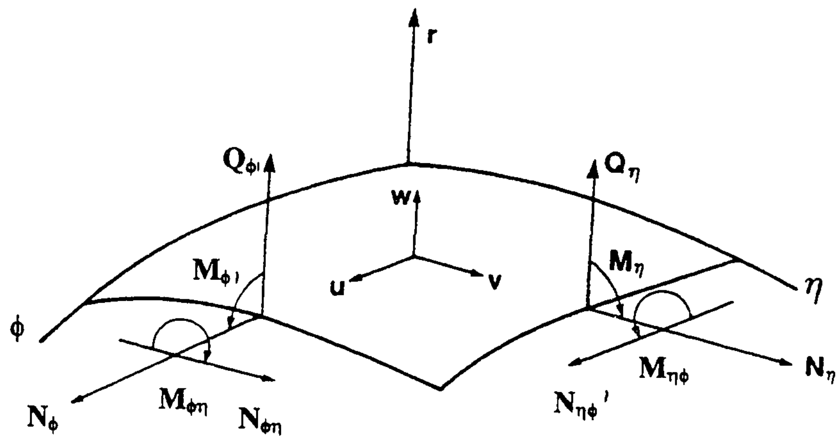


Fig. 4.4: Displacement components and stress resultants

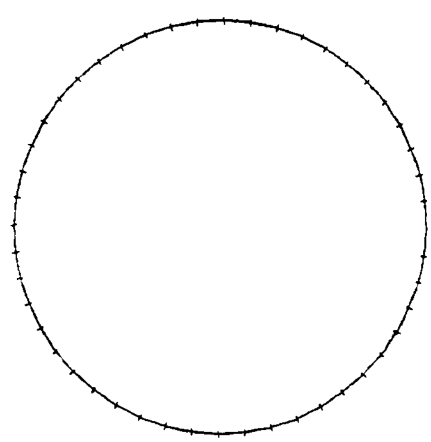


Fig. 4.5: DQM mesh – case 1

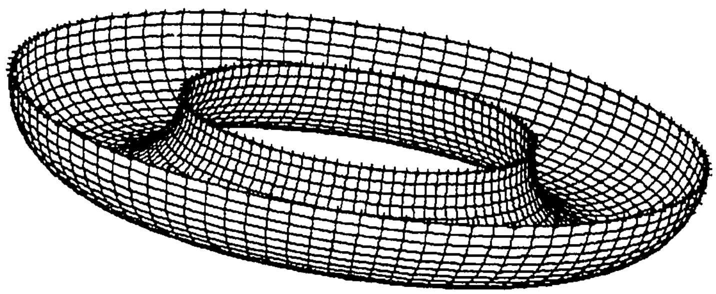


Fig. 4.6: FEM mesh (shell element) – case 1

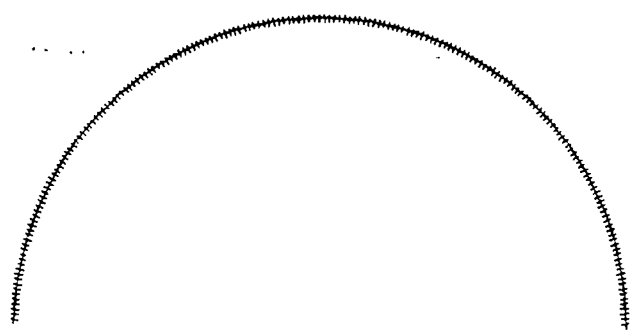
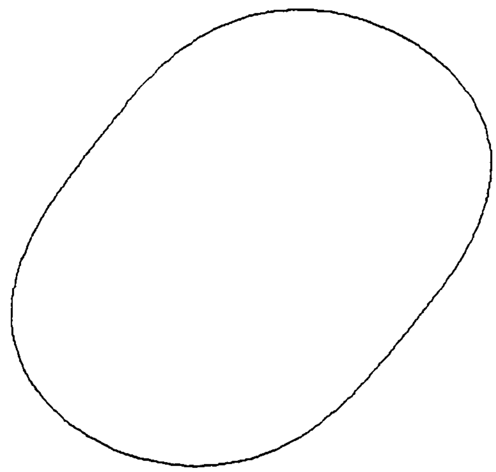
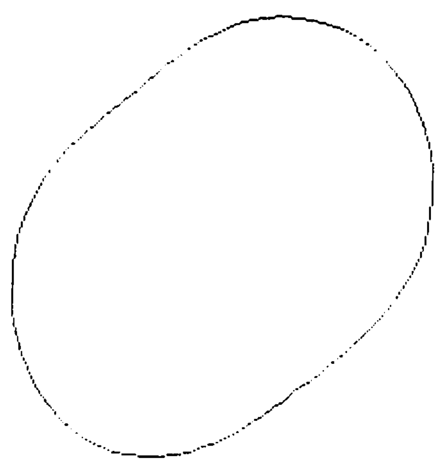


Fig. 4.7: FEM mesh (isobeam element) – case 1

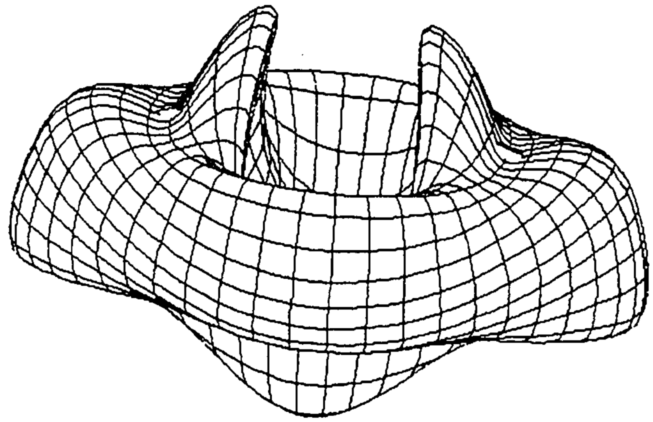


FEM result

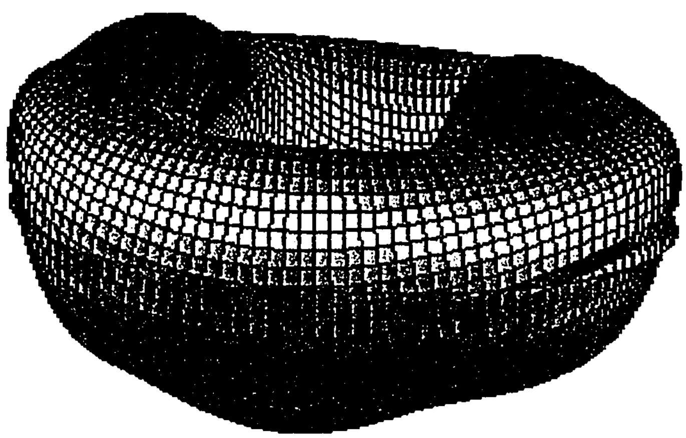


DQM result

Fig. 4.8: FEM and DQM results for mode shape 0A – case 1

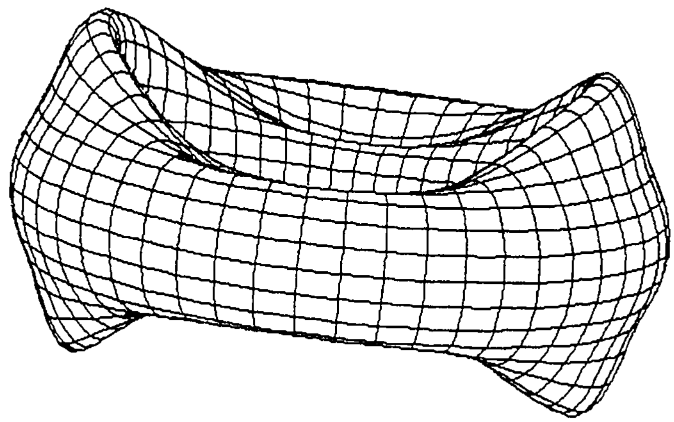


FEM result

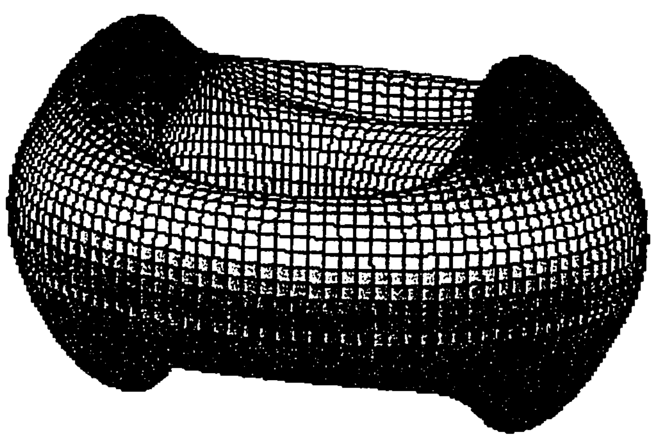


DQM result

Fig. 4.9: FEM and DQM results for mode shape 2A – case 1

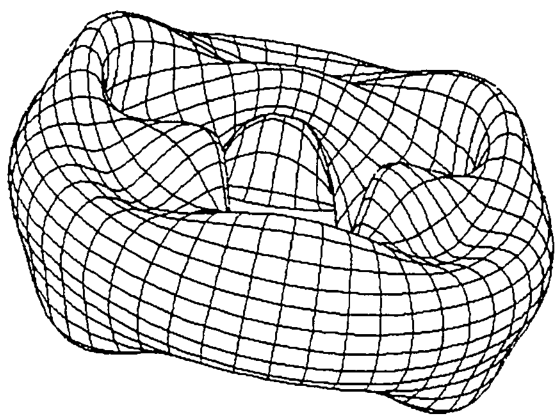


FEM result

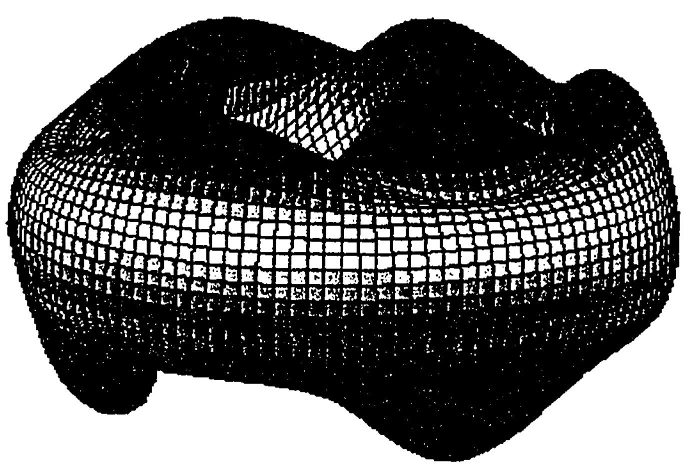


DQM result

Fig. 4.10: FEM and DQM results for mode shape 2S – case 1

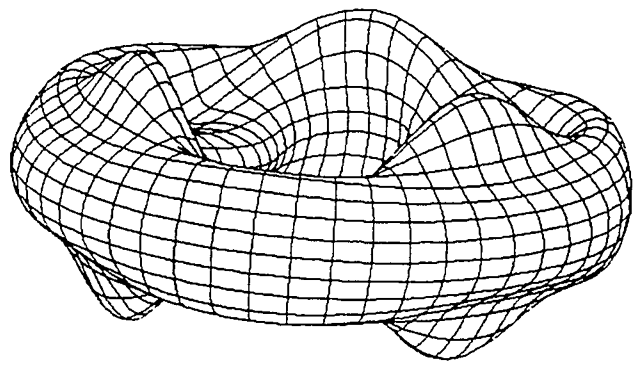


FEM result

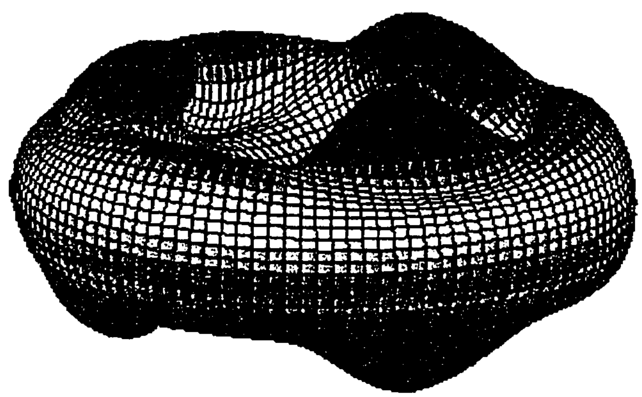


DQM rseult

Fig. 4.11: FEM and DQM results for mode shape 3A – case 1

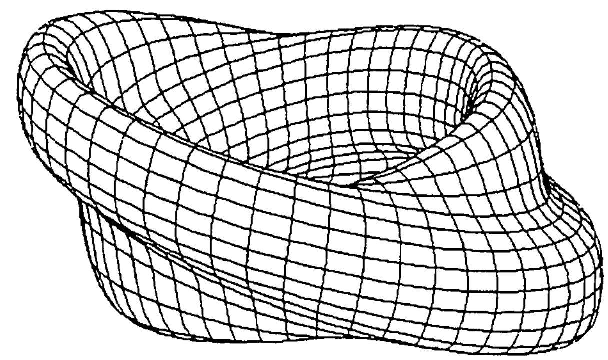


FEM result

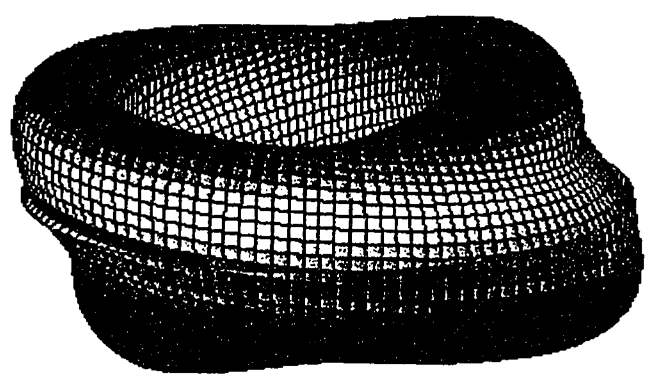


DQM result

Fig. 4.12: FEM and DQM results for mode shape 3S – case 1



FEM result



DQM result

Fig. 4.13: FEM and DQM results for mode shape 1A – case 1

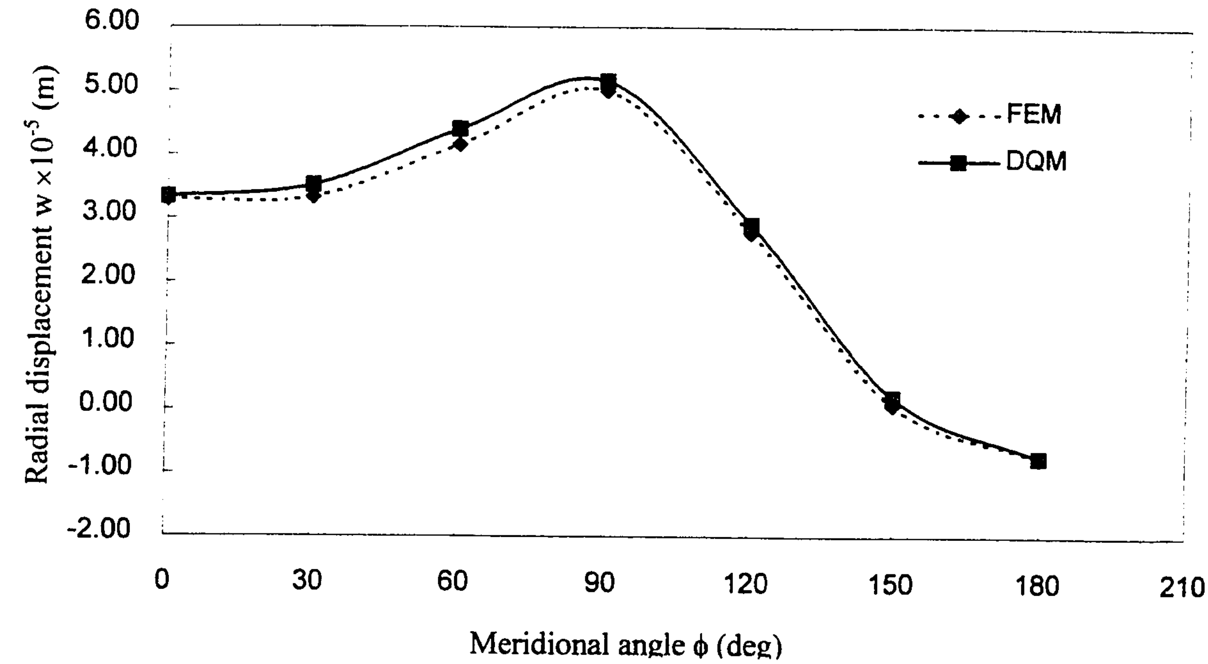


Fig. 4.14: Comparison of results for radial displacement w on plane $\theta = 0^o$ – case 1

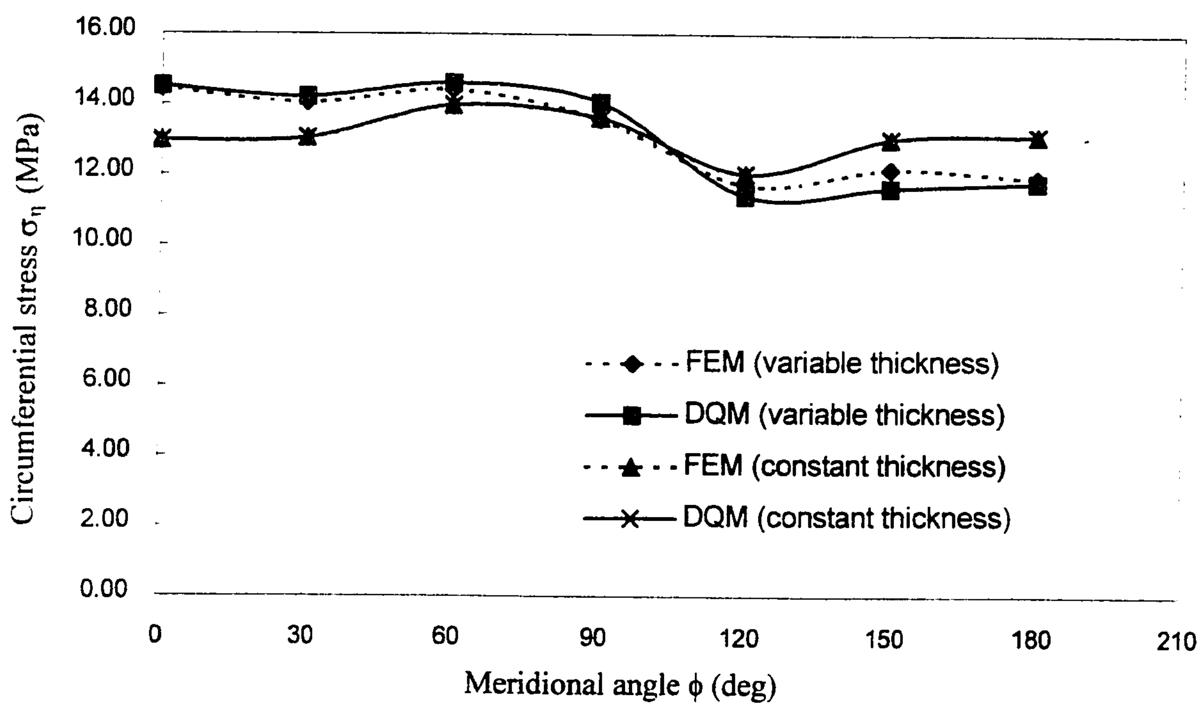


Fig. 4.15: Comparison of results for circumferential stress σ_η on plane $\theta = 0^o$ – case 1

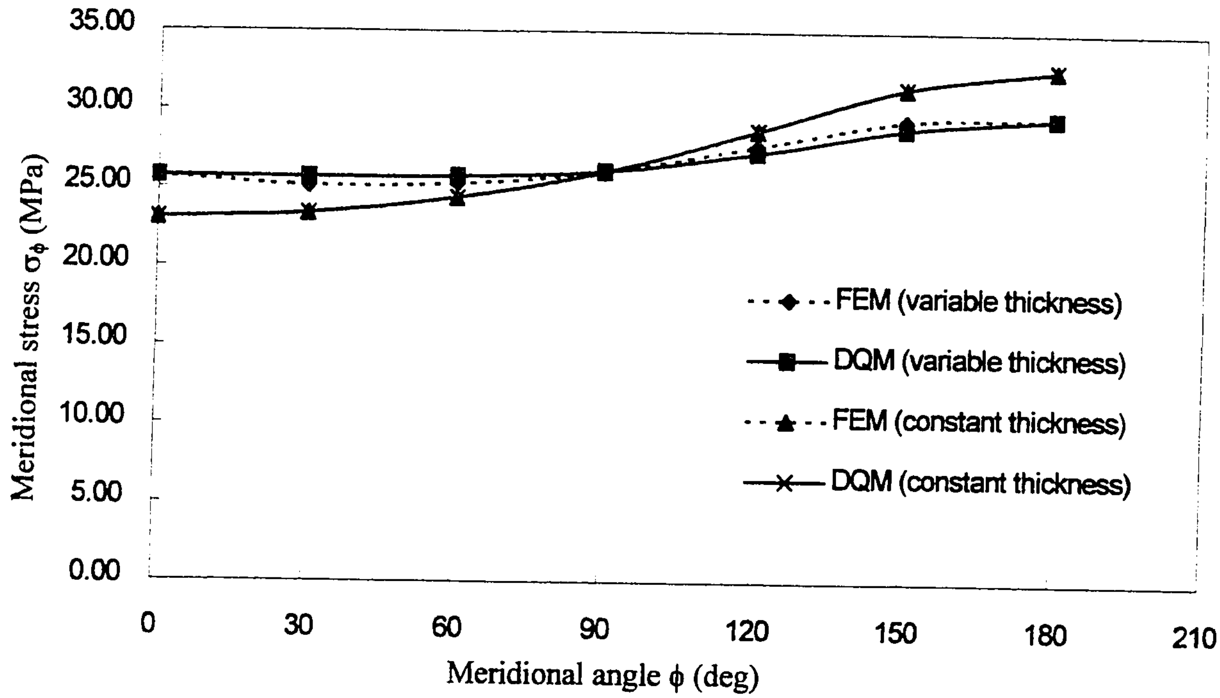


Fig. 4.16: Comparison of results for meridional stress σ_ϕ on plane $\theta = 0^o$ – case 1

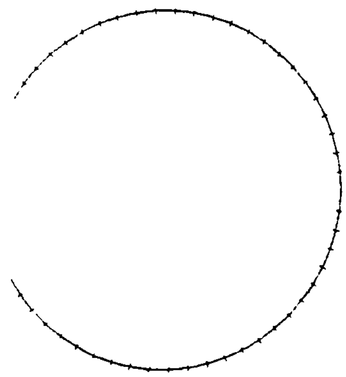


Fig. 4.17: DQM mesh – case2

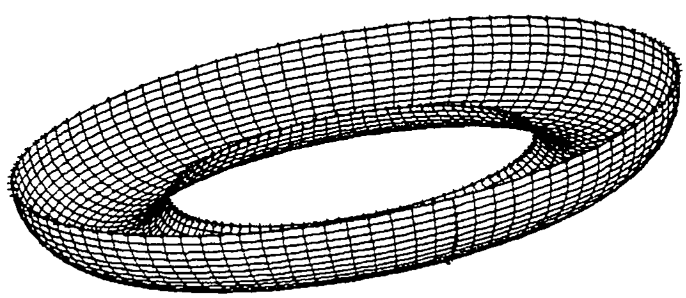
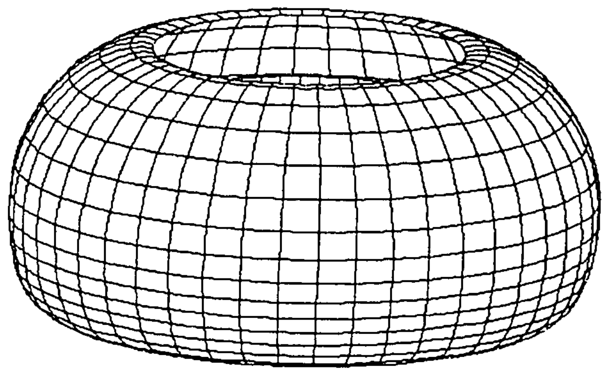
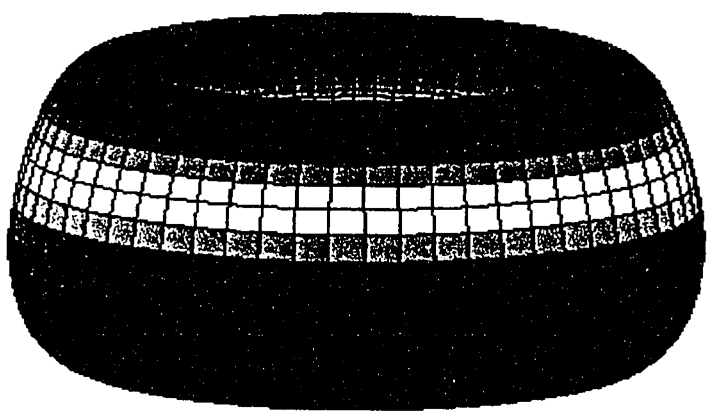


Fig. 4.18: FEM mesh – case2

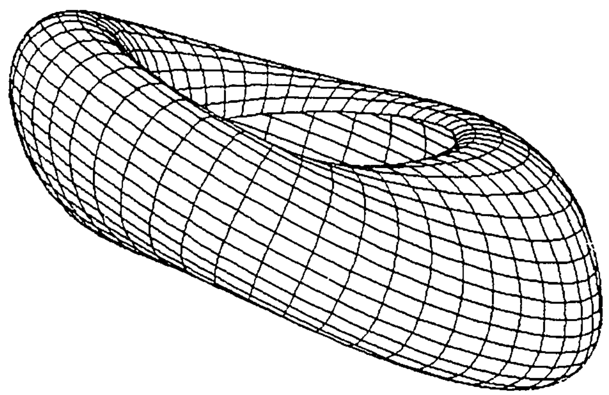


FEM result

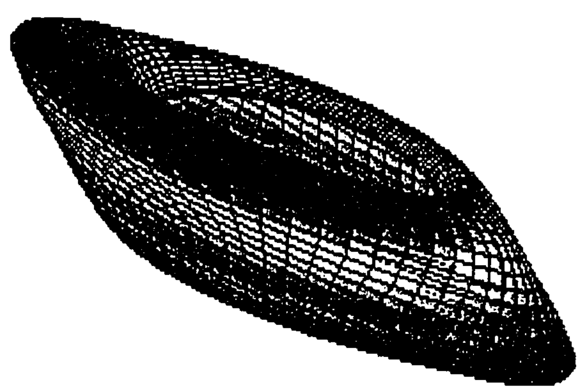


DQM result

Fig. 4.19: FEM and DQM results for mode shape 0A – case2

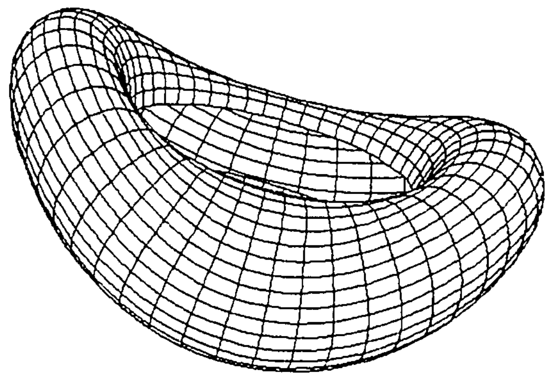


FEM result

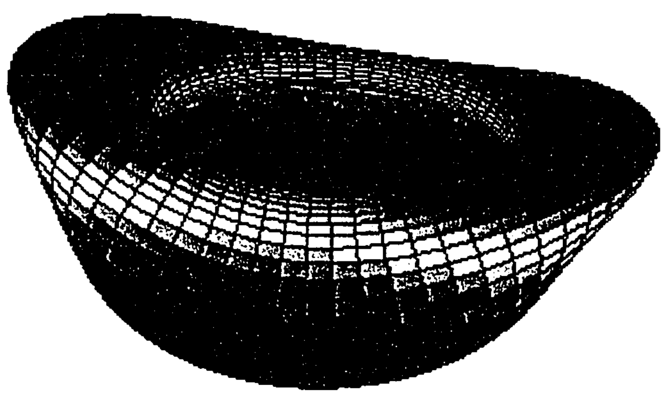


DQM result

Fig. 4.20: FEM and DQM results for mode shape 1A – case2

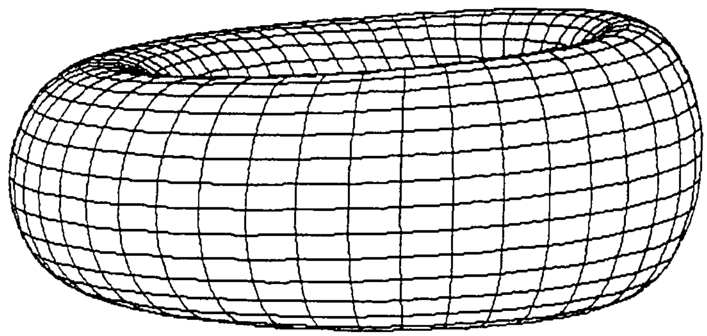


FEM result

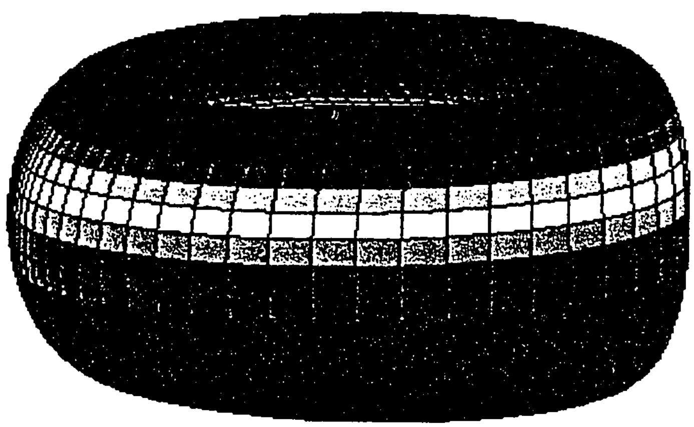


DQM result

Fig. 4.21: FEM and DQM results for mode shape 2A – case2

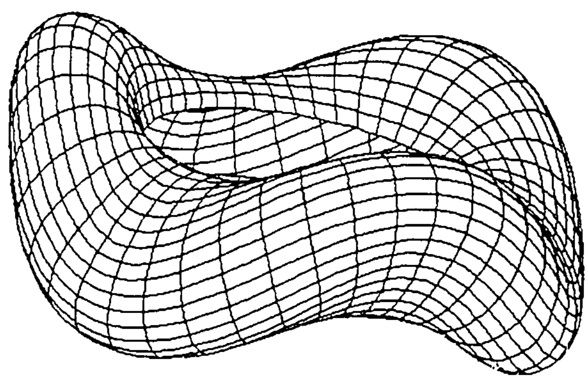


FEM result

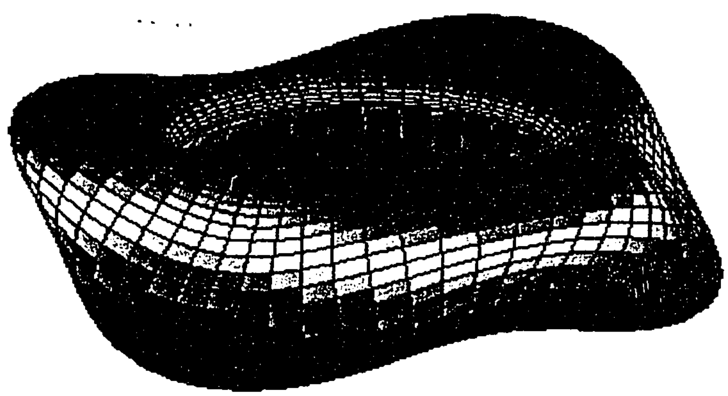


DQM result

Fig. 4.22: FEM and DQM results for mode shape 1S – case2

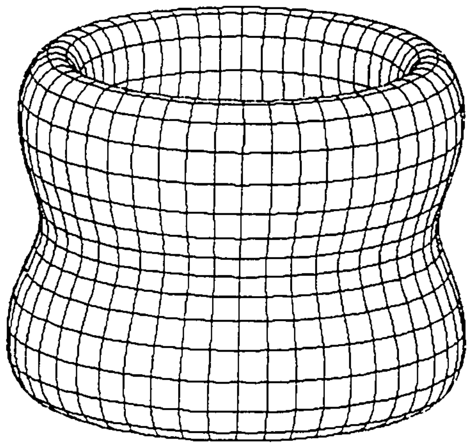


FEM result

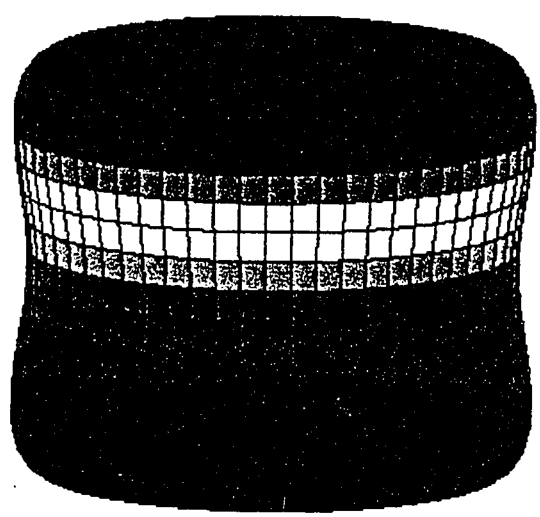


DQM result

Fig. 4.23: FEM and DQM results for mode shape 3A – case2



FEM result



DQM result

Fig. 4.24: FEM and DQM results for mode shape 0S – case2

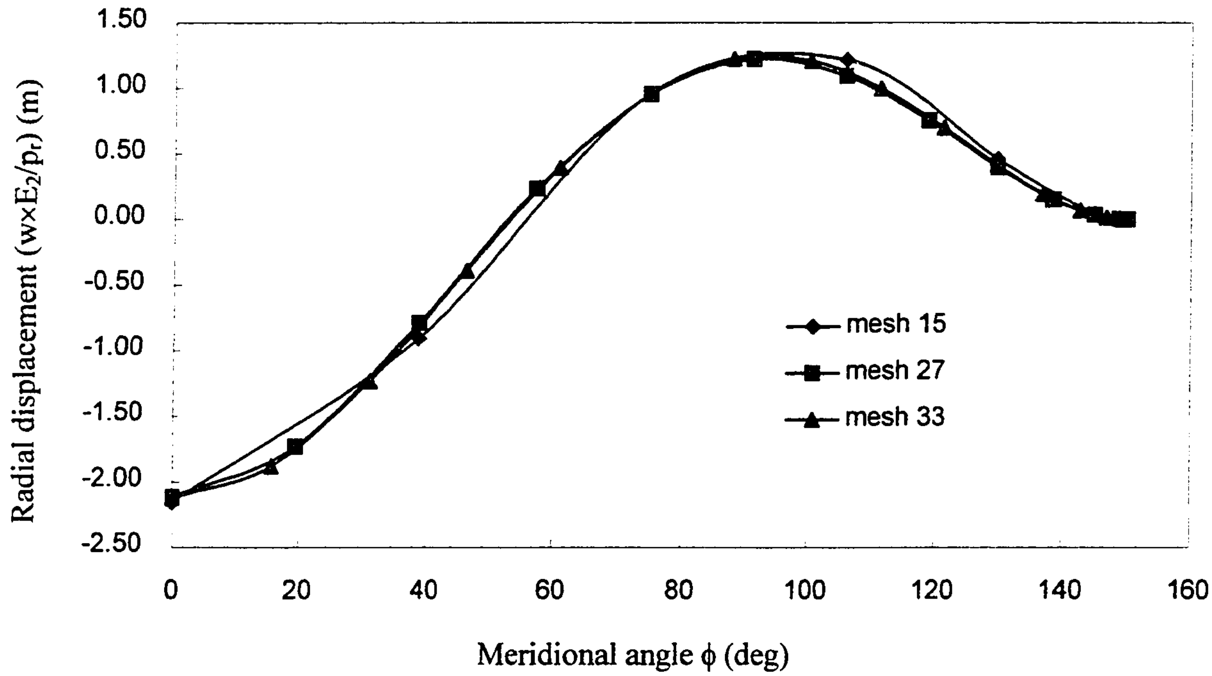


Fig. 4.25: Convergence of DQM – case 2

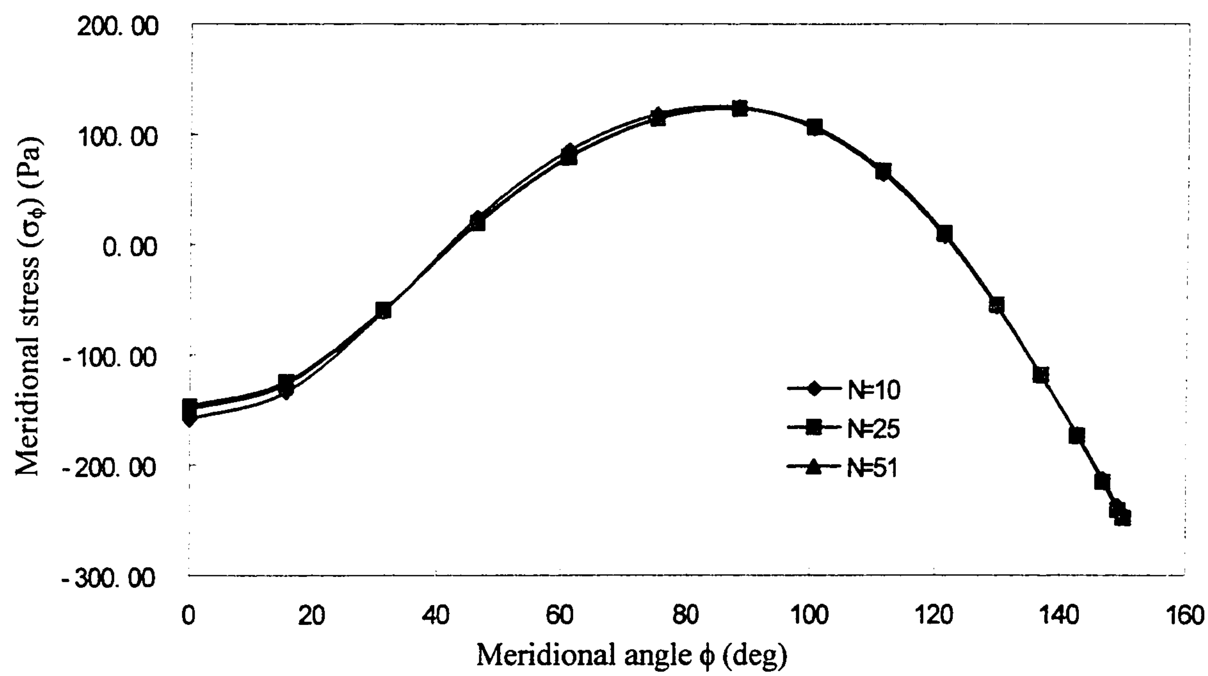


Fig. 4.26: Convergence of Fourier series – case 2

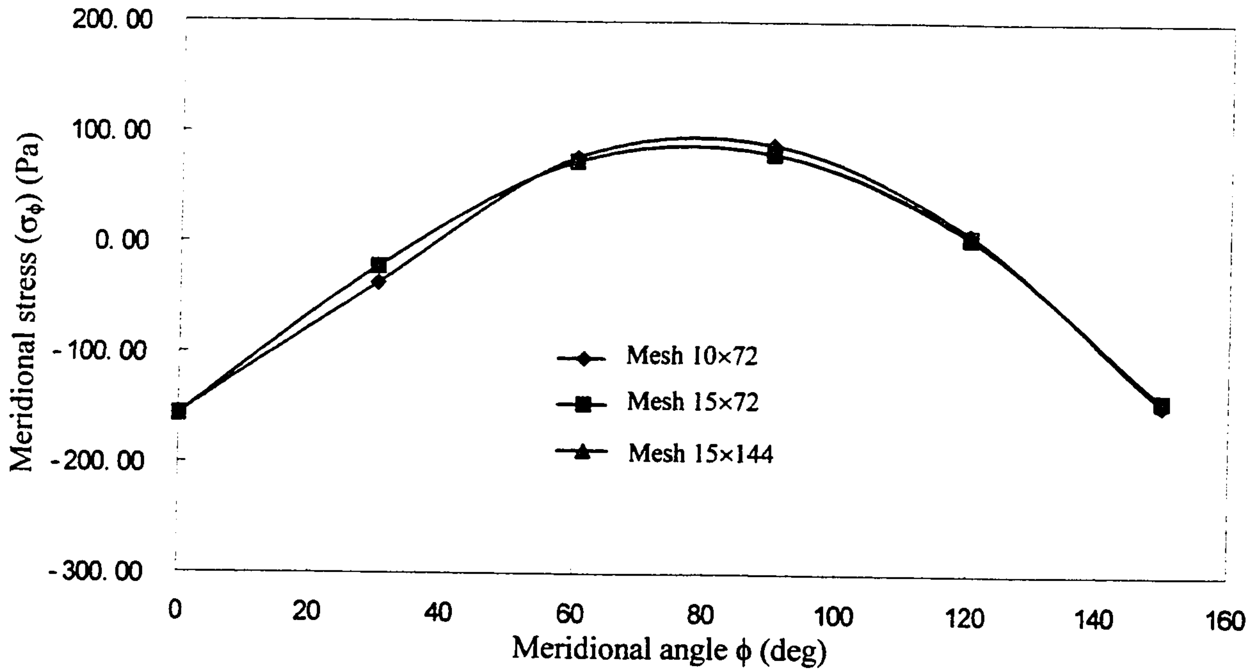


Fig. 4.27: Convergence of FEM – case 2

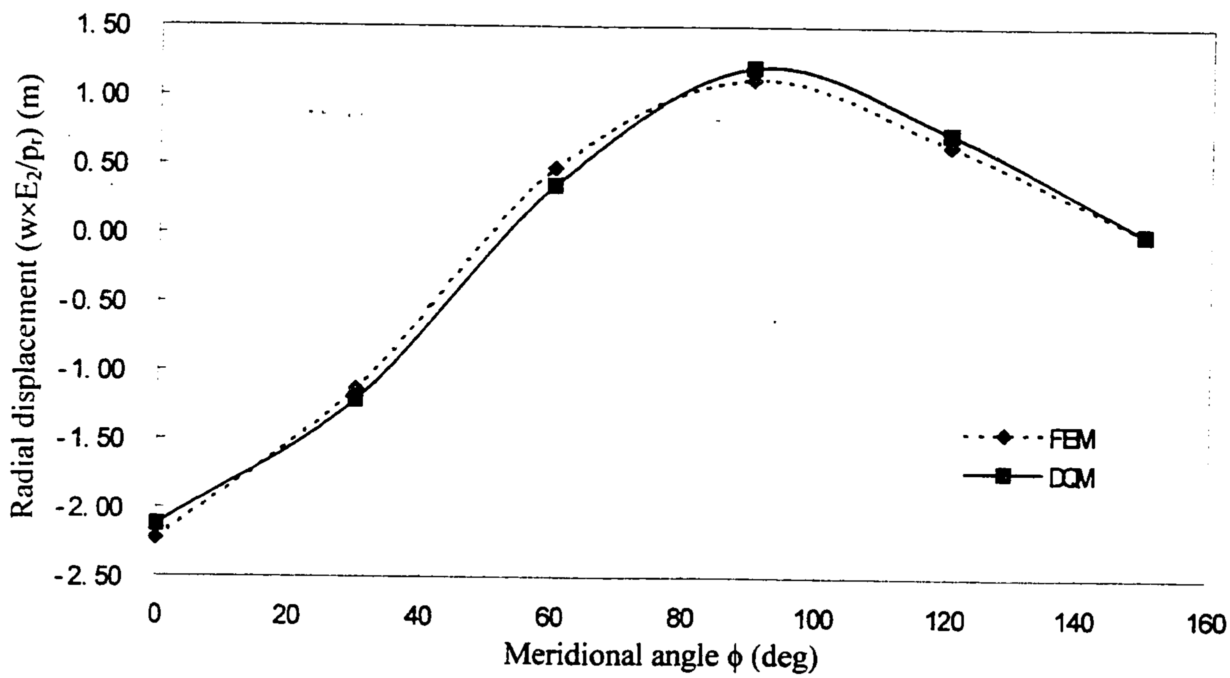


Fig. 4.28: Comparison of results for radial displacement parameter on plane θ=0^o – case 2

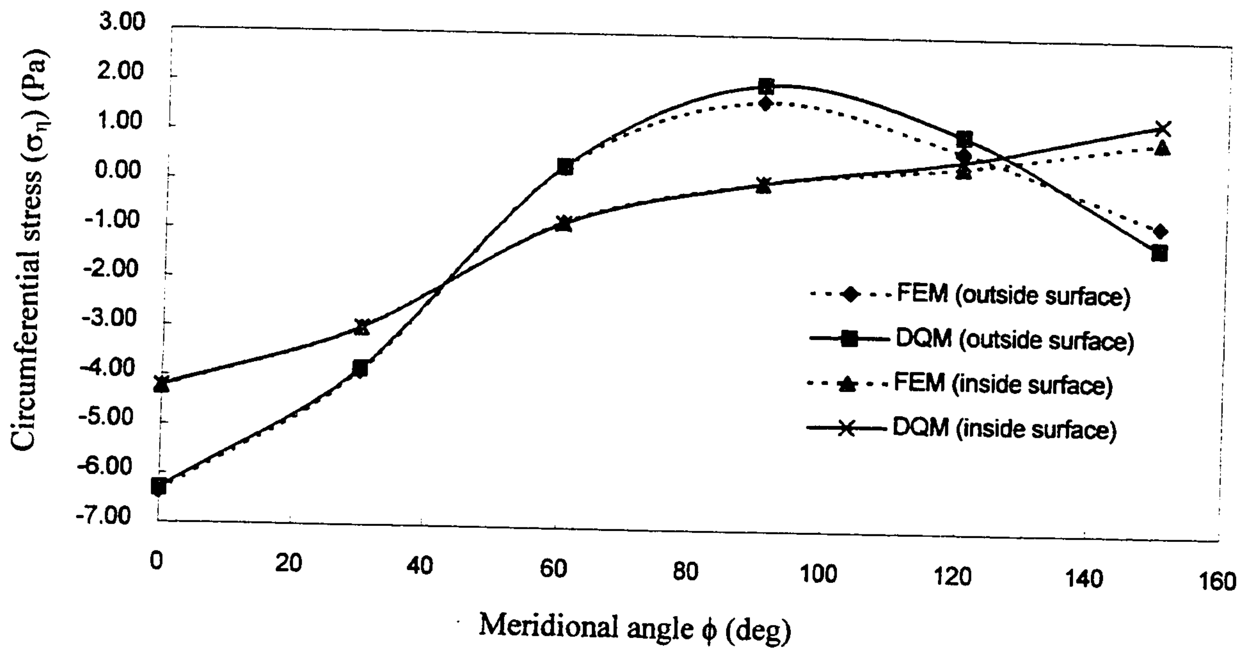


Fig. 4.29: Comparison of results for circumferential stress σ_η on plane θ=0° – case 2

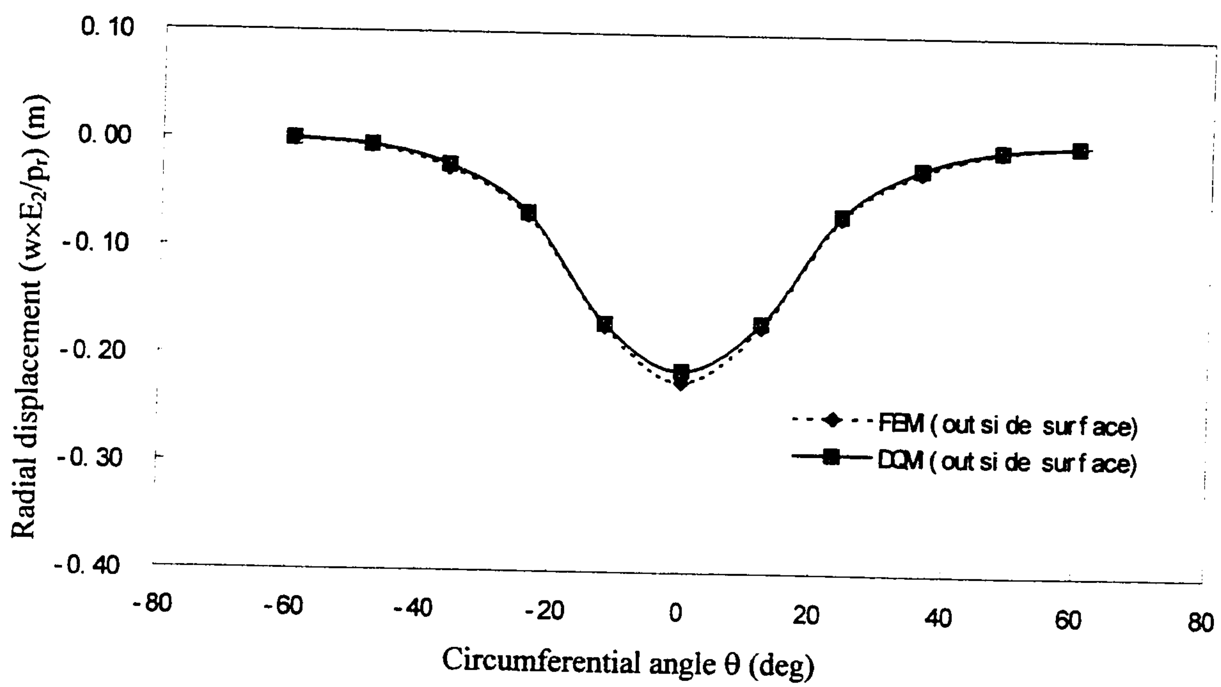


Fig. 4.30: Comparison of results for radial displacement parameter on plane θ=0° – case 2

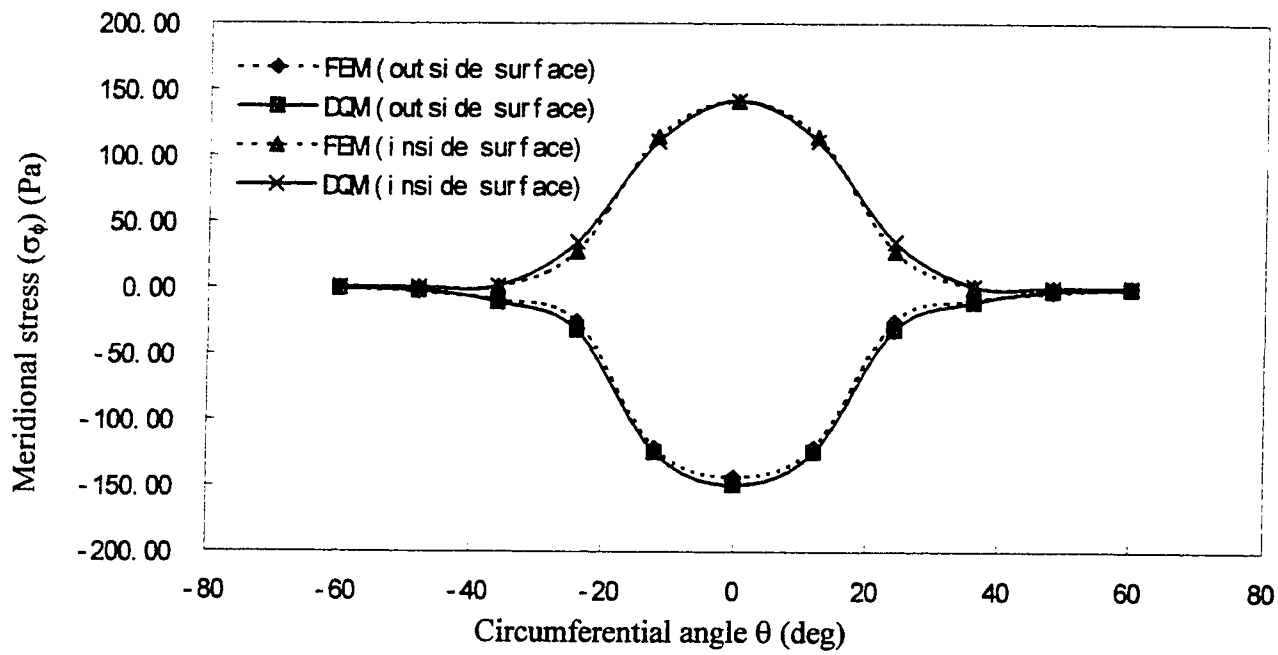


Fig. 4.31: Comparison of results for meridional stress σ_ϕ on plane θ=0° – case 2

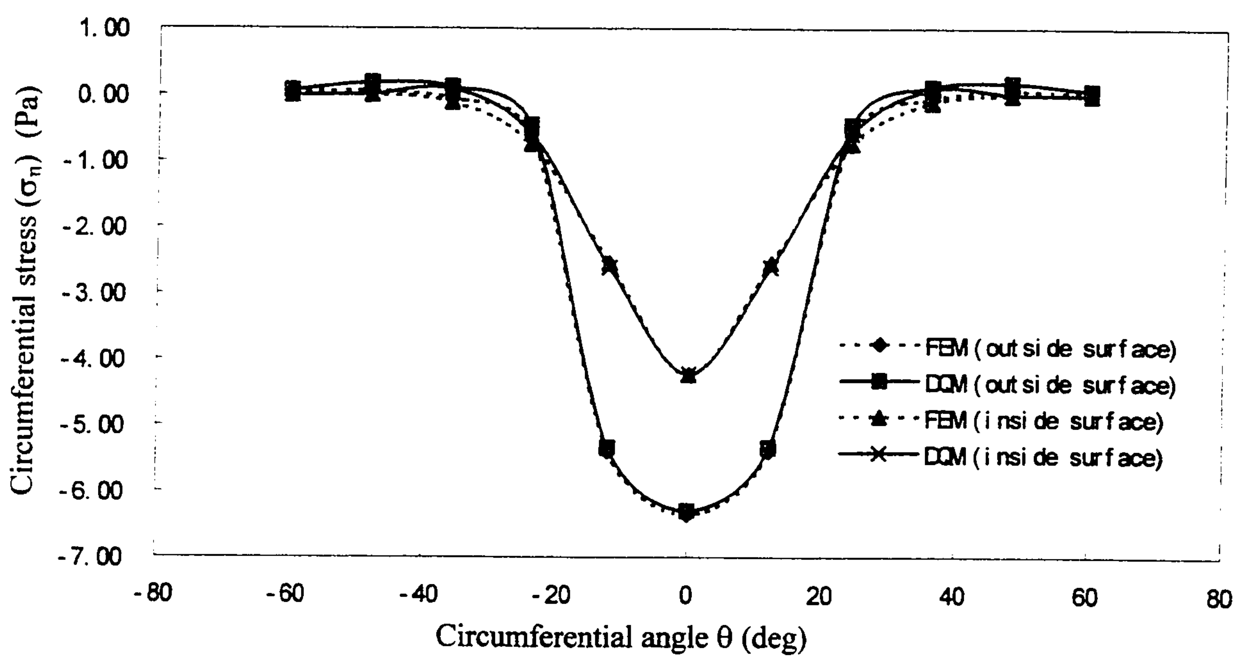


Fig. 4.32: Comparison of results for circumferential stress σ_η on plane θ=0° – case 2

APPENDIX A – Listing of coefficients $\nu_1 - \nu_{28}$

$$\nu_1 = a_1$$

$$\nu_2 = \frac{a_1}{r} + \frac{a_1 \cos\phi}{\rho}$$

$$\nu_3 = \frac{a_2}{r^2}$$

$$\nu_4 = -\frac{a_2 \sin\phi}{r\rho}$$

$$\nu_5 = -(\frac{(2n)^2 a_3}{\rho^2} + \frac{a_9}{r^2} - \frac{(a_4 + a_5)\cos\phi}{r\rho} + \frac{a_6 \cos^2\phi}{\rho^2})$$

$$\nu_6 = \frac{a_2 + a_7}{r}$$

$$\nu_7 = -\frac{(a_2 + a_8)\sin\phi}{\rho}$$

$$\nu_8 = -\frac{1}{r}(\frac{a_2 + a_9}{r} - \frac{a_5 \cos\phi}{\rho})$$

$$\nu_9 = -\frac{\sin\phi}{\rho}(\frac{a_4 - a_2}{r} - \frac{a_6 \cos\phi}{\rho})$$

$$\nu_{10} = \frac{2n}{\rho}(\frac{a_4}{r} - \frac{(a_3 + a_6)\cos\phi}{\rho})$$

$$\nu_{11} = \frac{2n(a_3 + a_8)}{\rho}$$

$$\nu_{12} = -\frac{a_{12} \sin\phi}{\rho}$$

$$\nu_{13} = \frac{1}{r}(\frac{a_2 + a_9}{r} + \frac{(a_2 + a_{13})\cos\phi}{\rho})$$

$$\nu_{14} = -\frac{\sin\phi}{\rho}(\frac{a_9}{r} - \frac{a_6 \cos\phi}{\rho})$$

$$\nu_{15} = a_2$$

$$\nu_{16} = a_2(\frac{1}{r} + \frac{\cos\phi}{\rho})$$

$$\nu_{17} = \frac{a_9}{r^2}$$

$$\nu_{18} = -\frac{a_9 \sin\phi}{r\rho}$$

$$\nu_{19} = -(\frac{(2n)^2 a_{14}}{\rho^2} + \frac{a_2}{r^2} + \frac{(a_2 + a_{13})\cos\phi}{r\rho} + \frac{a_6 \sin^2\phi}{\rho^2})$$

$$\nu_{20} = \frac{2n(a_{13} + a_{14})}{r\rho}$$

$$\nu_{21} = \frac{2n(a_6 + a_{14})\sin\phi}{\rho^2}$$

$$v_{22} = -\frac{1}{\rho}\left(\frac{a_3 \cos\phi}{\rho} + \frac{a_3 + a_{13}}{r} + \frac{a_6 \cos^2\phi}{\rho}\right)$$

$$v_{23} = -\frac{2n(a_{13} + a_{14})}{r\rho}$$

$$v_{24} = a_3$$

$$v_{25} = \frac{a_3}{r} + \frac{a_3 \cos\phi}{\rho}$$

$$v_{26} = \frac{a_{14}}{r^2}$$

$$v_{27} = -\frac{a_{14} \sin\phi}{r\rho}$$

$$v_{28} = -\left(\frac{(2n)^2 a_6 + a_{14} \sin^2\phi}{\rho^2} + \frac{a_3 \cos\phi}{\rho}\left(\frac{\cos\phi}{\rho} + \frac{1}{r}\right)\right)$$

where

$$a_1 = d_{11};\quad a_2 = \frac{d_{44}}{2};\quad a_3 = \frac{d_{55}}{2};\quad a_4 = d_{13} - d_{23};\quad a_5 = d_{12} - d_{23}$$

$$a_6 = d_{33};\quad a_7 = d_{12};\quad a_8 = d_{13};\quad a_9 = d_{22};\quad a_{10} = d_{12} - d_{33};\quad a_{11} = d_{13} - d_{33}$$

$$a_{12} = d_{12} - d_{13};\quad a_{13} = d_{23};\quad a_{14} = \frac{d_{66}}{2}$$

where d_{ij} were given in chapter 3

APPENDIX B - Listing of program Bandsta.m

```
% Bandsta.m last modification - Feb. 20th, 2001
% 3D elasticity theory using 3 gov eqns Int. J. Pres. Ves. & Piping 51 (1992)
% statics of isotropic toroids using semi-analytical DQM - band loading, steel
%---------------------------------------------------------------------------
% section input - input the problem parameters
% m=sampling points in radial direction; n1=trucation integer;
% n=sampling points in phi direction (even)
% pr=Poisson ratio; D=coefficient of material;
% E=young's modulus (MPa); p=pressure (MPa);
% rs=x-sect radius (mm); rarr=bend radius (mm);
% r1=outside radius (mm); r2=inside radius (mm);
% aphi=semi-angular length of an entire band;
% theta=the angle on circumferential;
% format long
m=12
n=24
n1=51
rarr=5; r1=2; r2=1; E=2.07e5; pr=0.3; pr1=1/pr;
G=E*pr1/(2*(pr1+1)); F=2*G/(pr1-2);
D=[F+2*G, F, F, 0, 0, 0; F, F+2*G, F, 0, 0, 0; F, F, F+2*G, 0, 0, 0;
0, 0, 0, 2*G, 0, 0; 0, 0, 0, 0, 2*G, 0; 0, 0, 0, 0, 0, 2*G];
aphi=2/45*pi; p=1;  ipr=4*aphi*p/(pi/2);
ipr=2*aphi/(pi/2); xdif=r1-r2; a1=D(1,1);
a2=D(4,4)/2; a3=D(5,5)/2; a4=D(1,3)-D(2,3);
a5=D(1,2)-D(2,3);a6=D(3,3); a7=D(1,2); a8=D(1,3);
a9=D(2,2); a10=D(1,2)-D(3,3); a11=D(1,3)-D(3,3);
a12=D(1,2)-D(1,3); a13=D(2,3); a14=D(6,6)/2; theta=0;
%
%---------------------------------------------------------------------------
-
% section wccd - find weighting coeffts for radial direction
% unequal spacing with one delta point at each end for radial direction
% output - b1(i,j),b2(i,j)- 1st to 2nd derivatives
%
for i=1:m; tt=pi*(i-1)/(m-1);y(i)=(1-cos(tt))/2; end
for i=1:m; x(i)=r2+y(i)*xdif; end
%
for i=1:m; pai(i)=1; for j=1:m
if i~=j; pai(i)=(x(i)-x(j))*pai(i); end
end;end
%
for i=1:m; for j=1:m
if i~=j; b1(i,j)=pai(i)/((x(i)-x(j))*pai(j)); end
end;end
%
for i=1:m; b1(i,i)=0; for j=1:m
if i~=j; b1(i,i)=b1(i,i)-b1(i,j); end
end;end
%
for i=1:m; for j=1:m
if i~=j; b2(i,j)=2*(b1(i,i)*b1(i,j)-b1(i,j)/(x(i)-x(j))); end
end;end
%
for i=1:m; b2(i,i)=0; for j=1:m
if i~=j; b2(i,i)=b2(i,i)-b2(i,j); end
```

```
end;end
%
%---------------------------------------------------------------------------
% section wcmd - find weighting coeffts for the meridional direction
% equal spacing with no delta points
% output - c1(i,j),c2(i,j) - 1st to 2nd derivatives
n2=n/2;
%
for i=1:n; tet(i)=2*pi*(i-1)/n; ib(i,1)=.5; ib(i,n2+1)=.5; end
%
for i=2:2:n; ib(i,n2+1)=-0.5; end
for i=2:n2; ib(1,i)=1; end
for i=n2+2:n; ib(1,i)=0; end
%
for i=2:n; for j=n2+2:n
ib(i,j)=sin(2*pi*(i-1)*(j-1-n2)/n);
end; end
%
for i=2:n; for j=2:n2
ib(i,j)=cos(2*pi*(i-1)*(j-1)/n);
end; end
%
ib=ib/n2;
%
for i=1:n; for j=1:n2+1
rr1(j)=(1-j)*sin((j-1)*tet(i));
rr2(j)=-1*((j-1)^2)*cos((j-1)*tet(i));
end
for j=n2+2:n
rr1(j)=(j-1-n2)*cos((j-1-n2)*tet(i));
rr2(j)=-1*((j-1-n2)^2)*sin((j-1-n2)*tet(i));
end
c1(i,:)=(ib*rr1')';
c2(i,:)=(ib*rr2')';
end
%++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
% solution for axisymmetric part
%
% section domeqn - set up matrices enforcing the governing equations
%    aii  aim     u      0
%    ami  amm     v  =   0
% output - aii(i,j),...,ann(i,j) - coefft matrices for displacements
mn=m*n;
%
aii(:,:)=zeros(mn);aim(:,:)=zeros(mn);
ami(:,:)=zeros(mn);amm(:,:)=zeros(mn);
%
% loop over sampling points in the r direction
for i=2:m-1
s=x(i);  s1=1/s;  s2=(s1)*(s1);
%
% loop over sampling points in the phi direction
for j=1:n
ir=(i-1)*n+j;
tee=tet(j); ct=cos(tee); st=sin(tee);
cp=rarr+s*ct; cp1=1/cp; cp2=(cp1)*(cp1);
%
```

```
% terms which do not involve derivatives
aii(ir,ir)=-s2*a9+s1*ct*cp1*(a4+a5)-cp2*ct*ct*a6;
aim(ir,ir)=-cp1*st*(s1*a4-cp1*ct*a6-s1*a2);
ami(ir,ir)=-cp1*st*(s1*a9-cp1*ct*a6);
amm(ir,ir)=-a2*s2-cp1*s1*ct*(a13+a2)-st*st*cp2*a6;
%
% terms which involve the r derivatives
for ii=1:m
ic=(ii-1)*n+j;
aii(ir,ic)=aii(ir,ic)+(b2(i,ii)+(s1+cp1*ct)*b1(i,ii))*a1;
aim(ir,ic)=aim(ir,ic)-cp1*st*(a8+a2)*b1(i,ii);
ami(ir,ic)=ami(ir,ic)-cp1*st*a12*b1(i,ii);
amm(ir,ic)=amm(ir,ic)+a2*(b2(i,ii)+(s1+ct*cp1)*b1(i,ii));
%
% terms which involve mixed derivatives
for ij=1:n
ic=(ii-1)*n+ij;
aim(ir,ic)=aim(ir,ic)+s1*(a2+a7)*b1(i,ii)*c1(j,ij);
ami(ir,ic)=ami(ir,ic)+s1*(a2+a7)*b1(i,ii)*c1(j,ij);
end
end
% terms which involve the phi derivatives
for jj=1:n
ic=(i-1)*n+jj;
aii(ir,ic)=aii(ir,ic)+(s2*c2(j,jj)-cp1*s1*st*c1(j,jj))*a2;
aim(ir,ic)=aim(ir,ic)-s1*(a2*s1+s1*a9-cp1*ct*a5)*c1(j,jj);
ami(ir,ic)=ami(ir,ic)+s1*(s1*(a2+a9)+cp1*ct*(a13+a2))*c1(j,jj);
amm(ir,ic)=amm(ir,ic)+s1*a9*(s1*c2(j,jj)-st*cp1*c1(j,jj));
end
end;end
%--------------------------------------------------------------------------------
% section boun - insert the boundary conditions
% band loading applied only on out surface r=b
%
% pressurized conditions on r=r1 line
%               and
% zero loading canditions on r=r2 line
%
ival=[1,m];
% loop over sampling points in the radial direction
% only for i=1 and i=m
for bloop=1:2
i=ival(bloop);
s=x(i); s1=1/s;
% loop over sampling points in the merid direction
for j=1:n
tee=tet(j);  ct=cos(tee);  st=sin(tee);  Ap=rarr+s*ct;
Ap1=1/Ap;
ir=(i-1)*n+j;
% terms which do not involve derivatives
aii(ir,ir)=a7*s1+Ap1*ct*a8;
aim(ir,ir)=-Ap1*st*a8;
amm(ir,ir)=-s1;
% loop for terms involving the r derivatives
for ii=1:m
ic=(ii-1)*n+j;
aii(ir,ic)=aii(ir,ic)+a1*b1(i,ii);
```

```
amm(ir,ic)=amm(ir,ic)+b1(i,ii);
end
% loop for terms involving the phi derivatives
for jj=1:n
ic=(i-1)*n+jj;
aim(ir,ic)=aim(ir,ic)+a7*s1*c1(j,jj);
ami(ir,ic)=ami(ir,ic)+s1*c1(j,jj);
end
end; end
%---------------------------------------------------------------------------
% section loadv - set up the load vector for the right hand side
% pressure on out side only
rhss=zeros(2*m*n,1);
for ij=(m-1)*n+1:mn
rhss(ij,1)=-ipr;
end
%---------------------------------------------------------------------------
% section solu - combine submat - solv matrix eqn - find results
% find the vector of displacement components rslt
stiff=[aii aim; ami amm];
clear aii;  clear aim; clear ami;  clear amm; clear pval; clear tval; clear
test1;
clear test2;
rslt=stiff\rhss; clear stiff;  clear rhss;
%
% displacement components at the sampling points, u,v
for i=1:m; for j=1:n
iju=(i-1)*n+j;  ijv=m*n+(i-1)*n+j;
u0(i,j)=rslt(iju);  v0(i,j)=rslt(ijv);
end;  end
ut=u0; vt=v0; wt=zeros(m,n); wt1=zeros(m,n);
%++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
% solution for nonsymmetric part
for k=1:n1
%
% section domeqn - set up matrices enforcing the governing equations
%    aii  aim  ain     u       0
%    ami  amm  amn     v   =   0
%    ani  anm  ann     w       0
% output - aii(i,j),...,ann(i,j) - coefft matrices for displacements
mn=m*n;
%
aii(:,:)=zeros(mn);aim(:,:)=zeros(mn);ain(:,:)=zeros(mn);
ami(:,:)=zeros(mn);amm(:,:)=zeros(mn);amn(:,:)=zeros(mn);
ani(:,:)=zeros(mn);anm(:,:)=zeros(mn);ann(:,:)=zeros(mn);
%
% loop over sampling points in the r direction
for i=2:m-1
s=x(i);  s1=1/s;  s2=(s1)*(s1);
%
% loop over sampling points in the phi direction
for j=1:n
ir=(i-1)*n+j;
tee=tet(j); ct=cos(tee); st=sin(tee);
cp=rarr+s*ct; cp1=1/cp; cp2=(cp1)*(cp1);
%
% terms which do not involve derivatives
```

```
aii(ir,ir)=-(4*k*k*cp2)*a3-s2*a9+s1*ct*cp1*(a4+a5)-cp2*ct*ct*a6;
aim(ir,ir)=-cp1*st*(s1*a4-cp1*ct*a6-s1*a2);
ain(ir,ir)=2*k*cp1*(s1*a4-cp1*ct*a3-cp1*ct*a6);
ami(ir,ir)=-cp1*st*(s1*a9-cp1*ct*a6);
amm(ir,ir)=-4*k*k*cp2*a14-a2*s2-cp1*s1*ct*(a13+a2)-st*st*cp2*a6;
amn(ir,ir)=st*cp2*(a14+a6)*k*2;
ani(ir,ir)=-cp1*((ct*cp1+s1)*a3+s1*a13+ct*cp1*a6)*k*2;
anm(ir,ir)=st*cp2*(a14+a6)*k*2;
ann(ir,ir)=-cp2*(4*k*k*a6+st*st*a14)-ct*cp1*a3*(ct*cp1+s1);
%
% terms which involve the r derivatives
for ii=1:m
ic=(ii-1)*n+j;
aii(ir,ic)=aii(ir,ic)+(b2(i,ii)+(s1+cp1*ct)*b1(i,ii))*a1;
aim(ir,ic)=aim(ir,ic)-cp1*st*(a8+a2)*b1(i,ii);
ain(ir,ic)=ain(ir,ic)+2*k*cp1*(a8+a3)*b1(i,ii);
ami(ir,ic)=ami(ir,ic)-cp1*st*a12*b1(i,ii);
amm(ir,ic)=amm(ir,ic)+a2*(b2(i,ii)+(s1+ct*cp1)*b1(i,ii));
ani(ir,ic)=ani(ir,ic)-2*k*cp1*(a3+a8)*b1(i,ii);
ann(ir,ic)=ann(ir,ic)+(b2(i,ii)+(ct*cp1+s1)*b1(i,ii))*a3;
%
% terms which involve mixed derivatives
for ij=1:n
ic=(ii-1)*n+ij;
aim(ir,ic)=aim(ir,ic)+s1*(a2+a7)*b1(i,ii)*c1(j,ij);
ami(ir,ic)=ami(ir,ic)+s1*(a2+a7)*b1(i,ii)*c1(j,ij);
end
end
% terms which involve the phi derivatives
for jj=1:n
ic=(i-1)*n+jj;
aii(ir,ic)=aii(ir,ic)+(s2*c2(j,jj)-cp1*s1*st*c1(j,jj))*a2;
aim(ir,ic)=aim(ir,ic)-s1*(a2*s1+s1*a9-cp1*ct*a5)*c1(j,jj);
ami(ir,ic)=ami(ir,ic)+s1*(s1*(a2+a9)+cp1*ct*(a13+a2))*c1(j,jj);
amm(ir,ic)=amm(ir,ic)+s1*a9*(s1*c2(j,jj)-st*cp1*c1(j,jj));
amn(ir,ic)=amn(ir,ic)+cp1*s1*(a13+a14)*2*k*c1(j,jj);
anm(ir,ic)=anm(ir,ic)-cp1*s1*(a14+a13)*2*k*c1(j,jj);
ann(ir,ic)=ann(ir,ic)+(s1*c2(j,jj)-st*cp1*c1(j,jj))*a14*s1;
end
end;end
%---------------------------------------------------------------------------------
% section boun - insert the boundary conditions
% band loading applied only on out surface r=b
%
% pressurized conditions on r=r1 line
%               and
% zero loading canditions on r=r2 line
%
ival=[1,m];
% loop over sampling points in the radial direction
% only for i=1 and i=m
for bloop=1:2
i=ival(bloop);
s=x(i); s1=1/s;
% loop over sampling points in the merid direction
for j=1:n
tee=tet(j);  ct=cos(tee);  st=sin(tee);  Ap=rarr+s*ct;
```

```
Ap1=1/Ap;
ir=(i-1)*n+j;
% terms which do not involve derivatives
aii(ir,ir)=a7*s1+Ap1*ct*a8;
aim(ir,ir)=-Ap1*st*a8;
ain(ir,ir)=Ap1*a8*k*2;
amm(ir,ir)=-s1;
ani(ir,ir)=-Ap1*k*2;
ann(ir,ir)=-ct*Ap1;
% loop for terms involving the r derivatives
for ii=1:m
ic=(ii-1)*n+j;
aii(ir,ic)=aii(ir,ic)+a1*b1(i,ii);
amm(ir,ic)=amm(ir,ic)+b1(i,ii);
ann(ir,ic)=ann(ir,ic)+b1(i,ii);
end
%loop for terms involving the phi derivatives
for jj=1:n
ic=(i-1)*n+jj;
aim(ir,ic)=aim(ir,ic)+a7*s1*c1(j,jj);
ami(ir,ic)=ami(ir,ic)+s1*c1(j,jj);
end
end; end
%---------------------------------------------------------------------------
% section loadv - set up the load vector for the right hand side
% pressure on out side only
rhss=zeros(3*m*n,1);
for ij=(m-1)*n+1:m*n
rhss(ij,1)=-2/(k*pi)*(1+(-1)^k)*sin(k*2*aphi);
end
%---------------------------------------------------------------------------
% section solu - combine submat - solv matrix eqn - find results
% find the vector of displacement components rslt
stiff=[aii aim ain; ami amm amn; ani anm ann];
clear aii;  clear aim; clear ain; clear ami;  clear amm; clear amn;
clear ani; clear anm; clear ann; clear pval; clear tval;
rslt=stiff\rhss; clear stiff; clear rhss;
%
% displacement components at the sampling points, u,v,w
for i=1:m; for j=1:n
iju=(i-1)*n+j;  ijv=m*n+(i-1)*n+j; ijw=2*m*n+(i-1)*n+j;
u(i,j)=rslt(iju);  v(i,j)=rslt(ijv); w(i,j)=rslt(ijw);
end;  end
%
for i=1:m; for j=1:n
ut(i,j)=ut(i,j)+u(i,j)*cos(2*k*theta);
vt(i,j)=vt(i,j)+v(i,j)*cos(2*k*theta);
wt(i,j)=wt(i,j)+w(i,j)*sin(2*k*theta);
wt1(i,j)=wt1(i,j)+w(i,j)*cos(2*k*theta)*2*k;
end; end
end
%
% stress components at the sampling points, srr, shh, stt, srh
% loop over samling points in the radial direction
for i=1:m
s=x(i); s1=1/s;
%
```

```
% loop over sampling points in the merid direction
for j=1:n
tee=tet(j);  ct=cos(tee);  st=sin(tee);
cp=rarr+s*ct;  cp1=1/cp;
% terms which do not involve derivatives
srr=(a9*s1+cp1*ct*a13)*ut(i,j)-cp1*a13*st*vt(i,j)+cp1*a13*wt1(i,j);
shh=(a13*s1+cp1*ct*a6)*ut(i,j)-cp1*a6*st*vt(i,j)+cp1*a6*wt1(i,j);
% loop for terms involving the r derivatives
for ii=1:m
srr=srr+a7*b1(i,ii)*ut(ii,j);
shh=shh+a8*b1(i,ii)*ut(ii,j);
end
% loop for terms involving the phi derivatives
for jj=1:n
srr=srr+s1*a9*c1(j,jj)*vt(i,jj);
shh=shh+s1*a13*c1(j,jj)*vt(i,jj);
end
sr(i,j)=srr; sh(i,j)=shh;
end;  end
%--------------------------------------------------------------------------------
%result output
sh
```

APPENDIX B - Listing of program Semistal.m

```
% Semistal.m last modification - May 28, 2001
% 3D elasticity theory using 3 gov. eqns Int. J. Pres. Ves. & Piping 51 (1992)
% statics of isotropic toroids using semi-analytical DQM - local loading, steel
%-------------------------------------------------------------------------------
% section input - input the problem parameters
% m=sampling points in radial direction; n1=trucation integer;
% n=sampling points in phi direction (even)
% pr=Poisson ratio;
% E=young's modulus (MPa); G=shear moduli (MPa);
% D=coefficient of material; p=pressure (MPa);
% rs=x-sect radius (mm); rarr=bend radius (mm);
% r1=outside radius (mm); r2=inside radius (mm);
% aphi=semi-angular length of an entire band;
% theta=the angle on circumferential; load=loading case;
% format long
m=8
n=24
n1=51
rarr=600; r1=115; r2=85; E=207000; pr=0.3; pr1=1/pr;
G=E*pr1/(2*(pr1+1)); F=2*G/(pr1-2);
D=[F+2*G, F, F, 0, 0, 0; F, F+2*G, F, 0, 0, 0; F, F, F+2*G, 0, 0, 0;
0, 0, 0, 2*G, 0, 0; 0, 0, 0, 0, 2*G, 0; 0, 0, 0, 0, 0, 2*G];
aphi=1/12*pi;
ipr=2*aphi/(pi/2); xdif=r1-r2; a1=D(1,1); a2=D(4,4)/2;
a3=D(5,5)/2; a4=D(1,3)-D(2,3); a5=D(1,2)-D(2,3);
a6=D(3,3); a7=D(1,2); a8=D(1,3); a9=D(2,2);
a10=D(1,2)-D(3,3); a11=D(1,3)-D(3,3);
a12=D(1,2)-D(1,3); a13=D(2,3); a14=D(6,6)/2; theta=0; load=3;
%
%-------------------------------------------------------------------------------
% section wcrd - find weighting coeffts for radial direction
% unequal spacing with one delta point at each end for radial direction
% output - b1(i,j),b2(i,j)- 1st to 2nd derivatives
%
for i=1:m; tt=pi*(i-1)/(m-1);y(i)=(1-cos(tt))/2; end
for i=1:m; x(i)=r2+y(i)*xdif; end
%
for i=1:m; pai(i)=1; for j=1:m
if i~=j; pai(i)=(x(i)-x(j))*pai(i); end
end;end
%
for i=1:m; for j=1:m
if i~=j; b1(i,j)=pai(i)/((x(i)-x(j))*pai(j)); end
end;end
%
for i=1:m; b1(i,i)=0; for j=1:m
if i~=j; b1(i,i)=b1(i,i)-b1(i,j); end
end;end
%
for i=1:m; for j=1:m
if i~=j; b2(i,j)=2*(b1(i,i)*b1(i,j)-b1(i,j)/(x(i)-x(j))); end
end;end
%
for i=1:m; b2(i,i)=0; for j=1:m
if i~=j; b2(i,i)=b2(i,i)-b2(i,j); end
```

```
end;end
%
%---------------------------------------------------------------------------
% section wcmd - find weighting coeffts for the meridional direction
% equal spacing with no delta points
% output - c1(i,j),c2(i,j) - 1st to 2nd derivatives
n2=n/2;
%
for i=1:n; tet(i)=2*pi*(i-1)/n; ib(i,1)=.5; ib(i,n2+1)=.5; end
%
for i=2:2:n; ib(i,n2+1)=-0.5; end
for i=2:n2; ib(1,i)=1; end
for i=n2+2:n; ib(1,i)=0; end
%
for i=2:n; for j=n2+2:n
ib(i,j)=sin(2*pi*(i-1)*(j-1-n2)/n);
end; end
%
for i=2:n; for j=2:n2
ib(i,j)=cos(2*pi*(i-1)*(j-1)/n);
end; end
%
ib=ib/n2;
%
for i=1:n; for j=1:n2+1
rr1(j)=(1-j)*sin((j-1)*tet(i));
rr2(j)=-1*((j-1)^2)*cos((j-1)*tet(i));
end
for j=n2+2:n
rr1(j)=(j-1-n2)*cos((j-1-n2)*tet(i));
rr2(j)=-1*((j-1-n2)^2)*sin((j-1-n2)*tet(i));
end
c1(i,:)=(ib*rr1')';
c2(i,:)=(ib*rr2')';
end
%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
% solution for axisymmetric part
%
% section domeqn - set up matrices enforcing the governing equations
%    aii  aim      u      0
%    ami  amm      v  =   0
% output - aii(i,j),...,ann(i,j) - coefft matrices for displacements
mn=m*n;
%
aii(:,:)=zeros(mn);aim(:,:)=zeros(mn);
ami(:,:)=zeros(mn);amm(:,:)=zeros(mn);
%
% loop over sampling points in the r direction
for i=2:m-1
s=x(i);  s1=1/s;  s2=(s1)*(s1);
%
% loop over sampling points in the phi direction
for j=1:n
ir=(i-1)*n+j;
tee=tet(j); ct=cos(tee); st=sin(tee);
cp=rarr+s*ct; cp1=1/cp; cp2=(cp1)*(cp1);
%
```

```
% terms which do not involve derivatives
aii(ir,ir)=-s2*a9+s1*ct*cp1*(a4+a5)-cp2*ct*ct*a6;
aim(ir,ir)=-cp1*st*(s1*a4-cp1*ct*a6-s1*a2);
ami(ir,ir)=-cp1*st*(s1*a9-cp1*ct*a6);
amm(ir,ir)=-a2*s2-cp1*s1*ct*(a13+a2)-st*st*cp2*a6;
%
% terms which involve the r derivatives
for ii=1:m
ic=(ii-1)*n+j;
aii(ir,ic)=aii(ir,ic)+(b2(i,ii)+(s1+cp1*ct)*b1(i,ii))*a1;
aim(ir,ic)=aim(ir,ic)-cp1*st*(a8+a2)*b1(i,ii);
ami(ir,ic)=ami(ir,ic)-cp1*st*a12*b1(i,ii);
amm(ir,ic)=amm(ir,ic)+a2*(b2(i,ii)+(s1+ct*cp1)*b1(i,ii));
%
% terms which involve mixed derivatives
for ij=1:n
ic=(ii-1)*n+ij;
aim(ir,ic)=aim(ir,ic)+s1*(a2+a7)*b1(i,ii)*c1(j,ij);
ami(ir,ic)=ami(ir,ic)+s1*(a2+a7)*b1(i,ii)*c1(j,ij);
end
end
% terms which involve the phi derivatives
for jj=1:n
ic=(i-1)*n+jj;
aii(ir,ic)=aii(ir,ic)+(s2*c2(j,jj)-cp1*s1*st*c1(j,jj))*a2;
aim(ir,ic)=aim(ir,ic)-s1*(a2*s1+s1*a9-cp1*ct*a5)*c1(j,jj);
ami(ir,ic)=ami(ir,ic)+s1*(s1*(a2+a9)+cp1*ct*(a13+a2))*c1(j,jj);
amm(ir,ic)=amm(ir,ic)+s1*a9*(s1*c2(j,jj)-st*cp1*c1(j,jj));
end
end;end
%---------------------------------------------------------------------------
% section boun - insert the boundary conditions
% pad loading applied only on out surface r=b
%
% pressurized conditions on r=r1 line
%                 and
% zero loading canditions on r=r2 line
%
ival=[1,m];
% loop over sampling points in the radial direction
% only for i=1 and i=m
for bloop=1:2
i=ival(bloop);
s=x(i); s1=1/s;
% loop over sampling points in the merid direction
for j=1:n
tee=tet(j); ct=cos(tee); st=sin(tee); Ap=rarr+s*ct;
Ap1=1/Ap;
ir=(i-1)*n+j;
% terms which do not involve derivatives
aii(ir,ir)=a7*s1+Ap1*ct*a8;
aim(ir,ir)=-Ap1*st*a8;
amm(ir,ir)=-s1;
% loop for terms involving the r derivatives
for ii=1:m
ic=(ii-1)*n+j;
aii(ir,ic)=aii(ir,ic)+a1*b1(i,ii);
```

```
amm(ir,ic)=amm(ir,ic)+b1(i,ii);
end
% loop for terms involving the phi derivatives
for jj=1:n
ic=(i-1)*n+jj;
aim(ir,ic)=aim(ir,ic)+a7*s1*c1(j,jj);
ami(ir,ic)=ami(ir,ic)+s1*c1(j,jj);
end
end; end
%------------------------------------------------------------------------------
% section loadv - set up the load vector for the right hand side
% pressure on out side only
rhss=zeros(2*m*n,1);
for j=1:n
pval=0;
tval=tet(j);
if load ==1
test1=1/6*pi;
test2=11/6*pi;
if tval > .999*test1, if tval < 1.001*test1, pval=0.5;   end; end;
if tval > .999*test2, if tval < 1.001*test2, pval=0.5;   end; end;
if tval > 1.001*test2,  pval=1.0;  end;
if tval < .999*test1, pval=1.0; end;
end
if load == 2
test1=5/12*pi;
test2=7/12*pi;
test3=17/12*pi;
test4=19/12*pi;
if tval > .999*test1, if tval < 1.001*test1, pval=0.5;   end; end;
if tval > .999*test2, if tval < 1.001*test2, pval=0.5;   end; end;
if tval > .999*test3, if tval < 1.001*test3, pval=0.5; end; end;
if tval > .999*test4, if tval < 1.001*test4, pval=0.5; end; end;
if tval > 1.001*test1, if tval < .999*test2, pval=1.0;  end; end;
if tval > 1.001*test3, if tval < .999*test4, pval=1.0; end; end;
end
if load == 3
test1=15/18*pi;
test2=21/18*pi;
if tval > .999*test1, if tval < 1.001*test1, pval=0.5;   end; end;
if tval > .999*test2, if tval < 1.001*test2, pval=0.5;   end; end;
if tval > 1.001*test1, if tval < .999*test2, pval=1.0;  end; end;
end
ij=(m-1)*n+j;
rhss(ij,1)=-ipr*pval;
end
%------------------------------------------------------------------------------
% section solu - combine submat - solv matrix eqn - find results
% find the vector of displacement components rslt
stiff=[aii aim; ami amm];
clear aii;  clear aim; clear ami;  clear amm; clear pval; clear tval; clear
test1;
clear test2;
rslt=stiff\rhss; clear stiff;  clear rhss;
%
% displacement components at the sampling points, u,v
for i=1:m; for j=1:n
```

```
 iju=(i-1)*n+j;  ijv=m*n+(i-1)*n+j;
u0(i,j)=rslt(iju);  v0(i,j)=rslt(ijv);
end;  end
ut=u0; vt=v0; wt=zeros(m,n); wt1=zeros(m,n);
%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
% solution for nonsymmetric part
%
for k=1:n1
%
% section domeqn - set up matrices enforcing the governing equations
%    aii  aim  ain      u       0
%    ami  amm  amn      v   =   0
%    ani  anm  ann      w       0
% output - aii(i,j),...,ann(i,j) - coefft matrices for displacements
mn=m*n;
%
aii(:,:)=zeros(mn);aim(:,:)=zeros(mn);ain(:,:)=zeros(mn);
ami(:,:)=zeros(mn);amm(:,:)=zeros(mn);amn(:,:)=zeros(mn);
ani(:,:)=zeros(mn);anm(:,:)=zeros(mn);ann(:,:)=zeros(mn);
%
% loop over sampling points in the r direction
for i=2:m-1
s=x(i);  s1=1/s;  s2=(s1)*(s1);
%
% loop over sampling points in the phi direction
for j=1:n
ir=(i-1)*n+j;
tee=tet(j); ct=cos(tee); st=sin(tee);
cp=rarr+s*ct; cp1=1/cp; cp2=(cp1)*(cp1);
%
% terms which do not involve derivatives
aii(ir,ir)=-(4*k*k*cp2)*a3-s2*a9+s1*ct*cp1*(a4+a5)-cp2*ct*ct*a6;
aim(ir,ir)=-cp1*st*(s1*a4-cp1*ct*a6-s1*a2);
ain(ir,ir)=2*k*cp1*(s1*a4-cp1*ct*a3-cp1*ct*a6);
ami(ir,ir)=-cp1*st*(s1*a9-cp1*ct*a6);
amm(ir,ir)=-4*k*k*cp2*a14-a2*s2-cp1*s1*ct*(a13+a2)-st*st*cp2*a6;
amn(ir,ir)=st*cp2*(a14+a6)*k*2;
ani(ir,ir)=-cp1*((ct*cp1+s1)*a3+s1*a13+ct*cp1*a6)*k*2;
anm(ir,ir)=st*cp2*(a14+a6)*k*2;
ann(ir,ir)=-cp2*(4*k*k*a6+st*st*a14)-ct*cp1*a3*(ct*cp1+s1);
%
% terms which involve the r derivatives
for ii=1:m
ic=(ii-1)*n+j;
aii(ir,ic)=aii(ir,ic)+(b2(i,ii)+(s1+cp1*ct)*b1(i,ii))*a1;
aim(ir,ic)=aim(ir,ic)-cp1*st*(a8+a2)*b1(i,ii);
ain(ir,ic)=ain(ir,ic)+2*k*cp1*(a8+a3)*b1(i,ii);
ami(ir,ic)=ami(ir,ic)-cp1*st*a12*b1(i,ii);
amm(ir,ic)=amm(ir,ic)+a2*(b2(i,ii)+(s1+ct*cp1)*b1(i,ii));
ani(ir,ic)=ani(ir,ic)-2*k*cp1*(a3+a8)*b1(i,ii);
ann(ir,ic)=ann(ir,ic)+(b2(i,ii)+(ct*cp1+s1)*b1(i,ii))*a3;
%
% terms which involve mixed derivatives
for ij=1:n
ic=(ii-1)*n+ij;
aim(ir,ic)=aim(ir,ic)+s1*(a2+a7)*b1(i,ii)*c1(j,ij);
ami(ir,ic)=ami(ir,ic)+s1*(a2+a7)*b1(i,ii)*c1(j,ij);
```

```
end
end
% terms which involve the phi derivatives
for jj=1:n
ic=(i-1)*n+jj;
aii(ir,ic)=aii(ir,ic)+(s2*c2(j,jj)-cp1*s1*st*c1(j,jj))*a2;
aim(ir,ic)=aim(ir,ic)-s1*(a2*s1+s1*a9-cp1*ct*a5)*c1(j,jj);
ami(ir,ic)=ami(ir,ic)+s1*(s1*(a2+a9)+cp1*ct*(a13+a2))*c1(j,jj);
amm(ir,ic)=amm(ir,ic)+s1*a9*(s1*c2(j,jj)-st*cp1*c1(j,jj));
amn(ir,ic)=amn(ir,ic)+cp1*s1*(a13+a14)*2*k*c1(j,jj);
anm(ir,ic)=anm(ir,ic)-cp1*s1*(a14+a13)*2*k*c1(j,jj);
ann(ir,ic)=ann(ir,ic)+(s1*c2(j,jj)-st*cp1*c1(j,jj))*a14*s1;
end
end;end
%-----------------------------------------------------------------------------
% section boun - insert the boundary conditions
% pad loading applied only on out surface r=b
%
% pressurized conditions on r=r1 line
%               and
% zero loading canditions on r=r2 line
%
ival=[1,m];
% loop over sampling points in the radial direction
% only for i=1 and i=m
for bloop=1:2
i=ival(bloop);
s=x(i); s1=1/s;
% loop over sampling points in the merid direction
for j=1:n
tee=tet(j);  ct=cos(tee);  st=sin(tee);  Ap=rarr+s*ct;
Ap1=1/Ap;
ir=(i-1)*n+j;
% terms which do not involve derivatives
aii(ir,ir)=a7*s1+Ap1*ct*a8;
aim(ir,ir)=-Ap1*st*a8;
ain(ir,ir)=Ap1*a8*k*2;
amm(ir,ir)=-s1;
ani(ir,ir)=-Ap1*k*2;
ann(ir,ir)=-ct*Ap1;
% loop for terms involving the r derivatives
for ii=1:m
ic=(ii-1)*n+j;
aii(ir,ic)=aii(ir,ic)+a1*b1(i,ii);
amm(ir,ic)=amm(ir,ic)+b1(i,ii);
ann(ir,ic)=ann(ir,ic)+b1(i,ii);
end
% loop for terms involving the phi derivatives
for jj=1:n
ic=(i-1)*n+jj;
aim(ir,ic)=aim(ir,ic)+a7*s1*c1(j,jj);
ami(ir,ic)=ami(ir,ic)+s1*c1(j,jj);
end
end; end
%-----------------------------------------------------------------------------
% section loadv - set up the load vector for the right hand side
% pressure on out side only
```

```
rhss=zeros(3*m*n,1);
for j=1:n
pval=0;
tval=tet(j);
if load==1
test1=1/6*pi;
test2=11/6*pi;
if tval > .999*test1, if tval < 1.001*test1, pval=0.5;   end; end;
if tval > .999*test2, if tval < 1.001*test2, pval=0.5;   end; end;
if tval > 1.001*test2,  pval=1.0;  end;
if tval < .999*test1, pval=1.0; end;
end
if load==2
test1=5/12*pi;
test2=7/12*pi;
test3=17/12*pi;
test4=19/12*pi;
if tval > .999*test1, if tval < 1.001*test1, pval=0.5;   end; end;
if tval > .999*test2, if tval < 1.001*test2, pval=0.5;   end; end;
if tval > .999*test3, if tval < 1.001*test3, pval=0.5; end; end;
if tval > .999*test4, if tval < 1.001*test4, pval=0.5; end; end;
if tval > 1.001*test1, if tval < .999*test2, pval=1.0;  end; end;
if tval > 1.001*test3, if tval < .999*test4, pval=1.0; end; end;
end
if load==3
test1=15/18*pi;
test2=21/18*pi;
if tval > .999*test1, if tval < 1.001*test1, pval=1.0;   end; end;
if tval > .999*test2, if tval < 1.001*test2, pval=1.0;   end; end;
if tval > 1.001*test1, if tval < .999*test2, pval=1.0;  end; end;
end
ij=(m-1)*n+j;
rhss(ij,1)=-2*pval/(k*pi)*(1+(-1)^k)*sin(k*2*aphi);
end
%-------------------------------------------------------------------------
% section solu - combine submat - solv matrix eqn - find results
% find the vector of displacement components rslt
stiff=[aii aim ain; ami amm amn; ani anm ann];
clear aii;  clear aim; clear ain; clear ami;  clear amm; clear amn;
clear ani; clear anm; clear ann; clear pval; clear tval;
rslt=stiff\rhss; clear stiff; clear rhss;
%
% displacement components at the sampling points, u,v,w
for i=1:m; for j=1:n
iju=(i-1)*n+j;  ijv=m*n+(i-1)*n+j; ijw=2*m*n+(i-1)*n+j;
u(i,j)=rslt(iju);  v(i,j)=rslt(ijv); w(i,j)=rslt(ijw);
end;  end
%
for i=1:m; for j=1:n
ut(i,j)=ut(i,j)+u(i,j)*cos(2*k*theta);
vt(i,j)=vt(i,j)+v(i,j)*cos(2*k*theta);
wt(i,j)=wt(i,j)+w(i,j)*sin(2*k*theta);
wt1(i,j)=wt1(i,j)+w(i,j)*cos(2*k*theta)*2*k;
end; end
end
%
% stress components at the sampling points, srr, shh, stt, srh
```

```
% loop over samling points in the radial direction
for i=1:m
s=x(i); s1=1/s;
%
% loop over sampling points in the merid direction
for j=1:n
tee=tet(j);  ct=cos(tee);  st=sin(tee);
cp=rarr+s*ct;  cp1=1/cp;
% terms which do not involve derivatives
srr=(a9*s1+cp1*ct*a13)*ut(i,j)-cp1*a13*st*vt(i,j)+cp1*a13*wt1(i,j);
shh=(a13*s1+cp1*ct*a6)*ut(i,j)-cp1*a6*st*vt(i,j)+cp1*a6*wt1(i,j);
% loop for terms involving the r derivatives
for ii=1:m
srr=srr+a7*b1(i,ii)*ut(ii,j);
shh=shh+a8*b1(i,ii)*ut(ii,j);
end
% loop for terms involving the phi derivatives
for jj=1:n
srr=srr+s1*a9*c1(j,jj)*vt(i,jj);
shh=shh+s1*a13*c1(j,jj)*vt(i,jj);
end
sr(i,j)=srr; sh(i,j)=shh;
end;  end
%---------------------------------------------------------------------------
% result output
sr
```

APPENDIX B - Listing of program Semista2.m

```
% Semista2.m last modification - May 28, 2001
% Statics of toroids using semi-analytical DQM local loading, M2 material
% 3D elasticity theory using 3 gov. eqns. Int. J. Pres. Ves. & Piping 51 (1992)
%-----------------------------------------------------------------------------
% section input - input the problem parameters
% m=sampling points in radial direction; n1=trucation integer;
% n=sampling points in phi direction (even)
% pr, pr1=Poisson ratio;
% E1, E=young's modulus (MPa); G, G1=shear moduli (MPa);
% D=coefficient of material; p=pressure (MPa);
% rs=x-sect radius (mm); rarr=bend radius (mm);
% r1=outside radius (mm); r2=inside radius (mm);
% aphi=semi-angular length of an entire band;
% theta=the angle on circumferential; load=loading case;
% format long
m=8
n=28
n1=51
rarr=0.6; r1=0.115; r2=0.085; E=6.9e9; E1=172.38e9; G1=3.45e9; pr=0.25;
pr1=0.25; delta=(E1*(1-pr1*pr1)-2*E*pr1*pr1*(1+pr))/(E*E*E1*E1);
G=E/(2*(1+pr)); d11=(E1-E*pr1*pr1)/(E*E1*E1*delta);
d22=(E1-E*pr1*pr1)/(E*E1*E1*delta);  d33=(1-pr*pr)/(E*E*delta);
d12=(pr*E1+E*pr1*pr1)/(E*E1*E1*delta);
d13=(pr1*(1+pr))/(E*E1*delta); d23=(pr1*(1+pr))/(E*E1*delta);
d44=2*G; d55=2*G1; d66=2*G1;
D=[d11,d12,d13,0,0,0;d12,d22,d23,0,0,0;d13,d23,d33,0,0,0;0,0,0,d44,0,0;
0,0,0,0,d55,0;0,0,0,0,0,d66];
xdif=r1-r2; a1=D(1,1); a2=D(4,4)/2; a3=D(5,5)/2; a4=D(1,3)-D(2,3);
a5=D(1,2)-D(2,3); a6=D(3,3); a7=D(1,2); a8=D(1,3); a9=D(2,2);
a10=D(1,2)-D(3,3); a11=D(1,3)-D(3,3);
a12=D(1,2)-D(1,3); a13=D(2,3); a14=D(6,6)/2;
aphi=1/12*pi;
ipr=2*aphi/(pi/2); theta=0; load=2;
%
%-----------------------------------------------------------------------------
% section wcrd - find weighting coeffts for radial direction
% unequal spacing with one delta point at each end for radial direction
% output - b1(i,j),b2(i,j)- 1st to 2nd derivatives
%
for i=1:m; tt=pi*(i-1)/(m-1);y(i)=(1-cos(tt))/2; end
for i=1:m; x(i)=r2+y(i)*xdif; end
%
for i=1:m; pai(i)=1; for j=1:m
if i~=j; pai(i)=(x(i)-x(j))*pai(i); end
end;end
%
for i=1:m; for j=1:m
if i~=j; b1(i,j)=pai(i)/((x(i)-x(j))*pai(j)); end
end;end
% .
for i=1:m; b1(i,i)=0; for j=1:m
if i~=j; b1(i,i)=b1(i,i)-b1(i,j); end
end;end
%
for i=1:m; for j=1:m
```

```
if i~=j; b2(i,j)=2*(b1(i,i)*b1(i,j)-b1(i,j)/(x(i)-x(j))); end
end;end
%
for i=1:m; b2(i,i)=0; for j=1:m
if i~=j; b2(i,i)=b2(i,i)-b2(i,j); end
end;end
%
%----------------------------------------------------------------
% section wcmd - find weighting coeffts for the meridional direction
% equal spacing with no delta points
% output - c1(i,j),c2(i,j) - 1st to 2nd derivatives
n2=n/2;
%
for i=1:n; tet(i)=2*pi*(i-1)/n; ib(i,1)=.5; ib(i,n2+1)=.5; end
%
for i=2:2:n; ib(i,n2+1)=-0.5; end
for i=2:n2; ib(1,i)=1; end
for i=n2+2:n; ib(1,i)=0; end
%
for i=2:n; for j=n2+2:n
ib(i,j)=sin(2*pi*(i-1)*(j-1-n2)/n);
end; end
%
for i=2:n; for j=2:n2
ib(i,j)=cos(2*pi*(i-1)*(j-1)/n);
end; end
%
ib=ib/n2;
%
for i=1:n; for j=1:n2+1
rr1(j)=(1-j)*sin((j-1)*tet(i));
rr2(j)=-1*((j-1)^2)*cos((j-1)*tet(i));
end
for j=n2+2:n
rr1(j)=(j-1-n2)*cos((j-1-n2)*tet(i));
rr2(j)=-1*((j-1-n2)^2)*sin((j-1-n2)*tet(i));
end
c1(i,:)=(ib*rr1')';
c2(i,:)=(ib*rr2')';
end
%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
% solution for axisymmetric part
%
% section domeqn - set up matrices enforcing the governing equations
%    aii  aim      u       0
%    ami  amm      v   =   0
% output - aii(i,j),...,ann(i,j) - coefft matrices for displacements
mn=m*n;
%
aii(:,:)=zeros(mn);aim(:,:)=zeros(mn);
ami(:,:)=zeros(mn);amm(:,:)=zeros(mn);
%
% loop over sampling points in the r direction
for i=2:m-1
s=x(i);  s1=1/s;  s2=(s1)*(s1);
%
% loop over sampling points in the phi direction
```

```
for j=1:n
ir=(i-1)*n+j;
tee=tet(j); ct=cos(tee); st=sin(tee);
cp=rarr+s*ct; cp1=1/cp; cp2=(cp1)*(cp1);
%
% terms which do not involve derivatives
aii(ir,ir)=-s2*a9+s1*ct*cp1*(a4+a5)-cp2*ct*ct*a6;
aim(ir,ir)=-cp1*st*(s1*a4-cp1*ct*a6-s1*a2);
ami(ir,ir)=-cp1*st*(s1*a9-cp1*ct*a6);
amm(ir,ir)=-a2*s2-cp1*s1*ct*(a13+a2)-st*st*cp2*a6;
%
% terms which involve the r derivatives
for ii=1:m
ic=(ii-1)*n+j;
aii(ir,ic)=aii(ir,ic)+(b2(i,ii)+(s1+cp1*ct)*b1(i,ii))*a1;
aim(ir,ic)=aim(ir,ic)-cp1*st*(a8+a2)*b1(i,ii);
ami(ir,ic)=ami(ir,ic)-cp1*st*a12*b1(i,ii);
amm(ir,ic)=amm(ir,ic)+a2*(b2(i,ii)+(s1+ct*cp1)*b1(i,ii));
%
% terms which involve mixed derivatives
for ij=1:n
ic=(ii-1)*n+ij;
aim(ir,ic)=aim(ir,ic)+s1*(a2+a7)*b1(i,ii)*c1(j,ij);
ami(ir,ic)=ami(ir,ic)+s1*(a2+a7)*b1(i,ii)*c1(j,ij);
end
end
% terms which involve the phi derivatives
for jj=1:n
ic=(i-1)*n+jj;
aii(ir,ic)=aii(ir,ic)+(s2*c2(j,jj)-cp1*s1*st*c1(j,jj))*a2;
aim(ir,ic)=aim(ir,ic)-s1*(a2*s1+s1*a9-cp1*ct*a5)*c1(j,jj);
ami(ir,ic)=ami(ir,ic)+s1*(s1*(a2+a9)+cp1*ct*(a13+a2))*c1(j,jj);
amm(ir,ic)=amm(ir,ic)+s1*a9*(s1*c2(j,jj)-st*cp1*c1(j,jj));
end
end;end
%------------------------------------------------------------------------------
% section boun - insert the boundary conditions
% pad loading applied only on out surface r=b
%
% pressurized conditions on r=r1 line
%              and
% zero loading canditions on r=r2 line
%
ival=[1,m];
% loop over sampling points in the radial direction
% only for i=1 and i=m
for bloop=1:2
i=ival(bloop);
s=x(i); s1=1/s;
% loop over sampling points in the merid direction
for j=1:n
tee=tet(j);  ct=cos(tee);  st=sin(tee);  Ap=rarr+s*ct;
Ap1=1/Ap;
ir=(i-1)*n+j;
% terms which do not involve derivatives
aii(ir,ir)=a7*s1+Ap1*ct*a8;
aim(ir,ir)=-Ap1*st*a8;
```

```
amm(ir,ir)=-s1;
% loop for terms involving the r derivatives
for ii=1:m
ic=(ii-1)*n+j;
aii(ir,ic)=aii(ir,ic)+a1*b1(i,ii);
amm(ir,ic)=amm(ir,ic)+b1(i,ii);
end
%loop for terms involving the phi derivatives
for jj=1:n
ic=(i-1)*n+jj;
aim(ir,ic)=aim(ir,ic)+a7*s1*c1(j,jj);
ami(ir,ic)=ami(ir,ic)+s1*c1(j,jj);
end
end; end
%---------------------------------------------------------------------------
% section loadv - set up the load vector for the right hand side
% pressure on out side only
rhss=zeros(2*m*n,1);
for j=1:n
pval=0;
tval=tet(j);
if load==1
test1=1/6*pi;
test2=11/6*pi;
if tval > .999*test1, if tval < 1.001*test1, pval=0.5;   end; end;
if tval > .999*test2, if tval < 1.001*test2, pval=0.5;   end; end;
if tval > 1.001*test2,  pval=1.0;  end;
if tval < .999*test1, pval=1.0; end;
end
if load==2
test1=5/12*pi;
test2=7/12*pi;
test3=17/12*pi;
test4=19/12*pi;
if tval > 1.001*test1, if tval < .999*test2, pval=1.0;  end; end;
if tval > 1.001*test3, if tval < .999*test4, pval=1.0; end; end;
end
if load==3
test1=15/18*pi;
test2=21/18*pi;
if tval > .999*test1, if tval < 1.001*test1, pval=0.5;   end; end;
if tval > .999*test2, if tval < 1.001*test2, pval=0.5;   end; end;
if tval > 1.001*test1, if tval < .999*test2, pval=1.0;  end; end;
end
ij=(m-1)*n+j;
rhss(ij,1)=-ipr*pval;
end
%---------------------------------------------------------------------------
% section solu - combine submat - solv matrix eqn - find results
% find the vector of displacement components rslt
stiff=[aii aim; ami amm];
clear aii;  clear aim; clear ami;  clear amm; clear pval; clear tval; clear
test1;
clear test2;
rslt=stiff\rhss; clear stiff;  clear rhss;
%
% displacement components at the sampling points, u,v
```

```
for i=1:m; for j=1:n
 iju=(i-1)*n+j;  ijv=m*n+(i-1)*n+j;
u0(i,j)=rslt(iju);  v0(i,j)=rslt(ijv);
end;  end
ut=u0; vt=v0; wt=zeros(m,n); wt1=zeros(m,n);
%++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
% solution for nonsymmetric part
%
for k=1:n1
%
% section domeqn - set up matrices enforcing the governing equations
%    aii  aim  ain     u      0
%    ami  amm  amn     v  =   0
%    ani  anm  ann     w      0
% output - aii(i,j),...,ann(i,j) - coefft matrices for displacements
mn=m*n;
%
aii(:,:)=zeros(mn);aim(:,:)=zeros(mn);ain(:,:)=zeros(mn);
ami(:,:)=zeros(mn);amm(:,:)=zeros(mn);amn(:,:)=zeros(mn);
ani(:,:)=zeros(mn);anm(:,:)=zeros(mn);ann(:,:)=zeros(mn);
%
% loop over sampling points in the r direction
for i=2:m-1
s=x(i);  s1=1/s;  s2=(s1)*(s1);
%
% loop over sampling points in the phi direction
for j=1:n
ir=(i-1)*n+j;
tee=tet(j); ct=cos(tee); st=sin(tee);
cp=rarr+s*ct; cp1=1/cp; cp2=(cp1)*(cp1);
%
% terms which do not involve derivatives
aii(ir,ir)=-(4*k*k*cp2)*a3-s2*a9+s1*ct*cp1*(a4+a5)-cp2*ct*ct*a6;
aim(ir,ir)=-cp1*st*(s1*a4-cp1*ct*a6-s1*a2);
ain(ir,ir)=2*k*cp1*(s1*a4-cp1*ct*a3-cp1*ct*a6);
ami(ir,ir)=-cp1*st*(s1*a9-cp1*ct*a6);
amm(ir,ir)=-4*k*k*cp2*a14-a2*s2-cp1*s1*ct*(a13+a2)-st*st*cp2*a6;
amn(ir,ir)=st*cp2*(a14+a6)*k*2;
ani(ir,ir)=-cp1*((ct*cp1+s1)*a3+s1*a13+ct*cp1*a6)*k*2;
anm(ir,ir)=st*cp2*(a14+a6)*k*2;
ann(ir,ir)=-cp2*(4*k*k*a6+st*st*a14)-ct*cp1*a3*(ct*cp1+s1);
%
% terms which involve the r derivatives
for ii=1:m
ic=(ii-1)*n+j;
aii(ir,ic)=aii(ir,ic)+(b2(i,ii)+(s1+cp1*ct)*b1(i,ii))*a1;
aim(ir,ic)=aim(ir,ic)-cp1*st*(a8+a2)*b1(i,ii);
ain(ir,ic)=ain(ir,ic)+2*k*cp1*(a8+a3)*b1(i,ii);
ami(ir,ic)=ami(ir,ic)-cp1*st*a12*b1(i,ii);
amm(ir,ic)=amm(ir,ic)+a2*(b2(i,ii)+(s1+ct*cp1)*b1(i,ii));
ani(ir,ic)=ani(ir,ic)-2*k*cp1*(a3+a8)*b1(i,ii);
ann(ir,ic)=ann(ir,ic)+(b2(i,ii)+(ct*cp1+s1)*b1(i,ii))*a3;
%
% terms which involve mixed derivatives
for ij=1:n
ic=(ii-1)*n+ij;
aim(ir,ic)=aim(ir,ic)+s1*(a2+a7)*b1(i,ii)*c1(j,ij);
```

```
ami(ir,ic)=ami(ir,ic)+s1*(a2+a7)*b1(i,ii)*c1(j,ij);
end
end
% terms which involve the phi derivatives
for jj=1:n
ic=(i-1)*n+jj;
aii(ir,ic)=aii(ir,ic)+(s2*c2(j,jj)-cp1*s1*st*c1(j,jj))*a2;
aim(ir,ic)=aim(ir,ic)-s1*(a2*s1+s1*a9-cp1*ct*a5)*c1(j,jj);
ami(ir,ic)=ami(ir,ic)+s1*(s1*(a2+a9)+cp1*ct*(a13+a2))*c1(j,jj);
amm(ir,ic)=amm(ir,ic)+s1*a9*(s1*c2(j,jj)-st*cp1*c1(j,jj));
amn(ir,ic)=amn(ir,ic)+cp1*s1*(a13+a14)*2*k*c1(j,jj);
anm(ir,ic)=anm(ir,ic)-cp1*s1*(a14+a13)*2*k*c1(j,jj);
ann(ir,ic)=ann(ir,ic)+(s1*c2(j,jj)-st*cp1*c1(j,jj))*a14*s1;
end
end;end
%-----------------------------------------------------------------------------
% section boun - insert the boundary conditions
% pad loading applied only on out surface r=b
%
% pressurized conditions on r=r1 line
%              and
% zero loading canditions on r=r2 line
%
ival=[1,m];
% loop over sampling points in the radial direction
% only for i=1 and i=m
for bloop=1:2
i=ival(bloop);
s=x(i); s1=1/s;
% loop over sampling points in the merid direction
for j=1:n
tee=tet(j);  ct=cos(tee);  st=sin(tee);  Ap=rarr+s*ct;
Ap1=1/Ap;
ir=(i-1)*n+j;
% terms which do not involve derivatives
aii(ir,ir)=a7*s1+Ap1*ct*a8;
aim(ir,ir)=-Ap1*st*a8;
ain(ir,ir)=Ap1*a8*k*2;
amm(ir,ir)=-s1;
ani(ir,ir)=-Ap1*k*2;
ann(ir,ir)=-ct*Ap1;
% loop for terms involving the r derivatives
for ii=1:m
ic=(ii-1)*n+j;
aii(ir,ic)=aii(ir,ic)+a1*b1(i,ii);
amm(ir,ic)=amm(ir,ic)+b1(i,ii);
ann(ir,ic)=ann(ir,ic)+b1(i,ii);
end
% loop for terms involving the phi derivatives
for jj=1:n
ic=(i-1)*n+jj;
aim(ir,ic)=aim(ir,ic)+a7*s1*c1(j,jj);
ami(ir,ic)=ami(ir,ic)+s1*c1(j,jj);
end
end; end
%-----------------------------------------------------------------------------
% section loadv - set up the load vector for the right hand side
```

```
% pressure on out side only
rhss=zeros(3*m*n,1);
for j=1:n
pval=0;
tval=tet(j);
if load==1
test1=1/6*pi;
test2=11/6*pi;
if tval > .999*test1, if tval < 1.001*test1, pval=0.5;   end; end;
if tval > .999*test2, if tval < 1.001*test2, pval=0.5;   end; end;
if tval > 1.001*test2,  pval=1.0;  end;
if tval < .999*test1, pval=1.0; end;
end
if load==2
test1=5/12*pi;
test2=7/12*pi;
test3=17/12*pi;
test4=19/12*pi;
if tval > 1.001*test1, if tval < .999*test2, pval=1.0;  end; end;
if tval > 1.001*test3, if tval < .999*test4, pval=1.0; end; end;
end
if load==3
test1=15/18*pi;
test2=21/18*pi;
if tval > .999*test1, if tval < 1.001*test1, pval=1.0;   end; end;
if tval > .999*test2, if tval < 1.001*test2, pval=1.0;   end; end;
if tval > 1.001*test1, if tval < .999*test2, pval=1.0;  end; end;
end
ij=(m-1)*n+j;
rhss(ij,1)=-2*pval/(k*pi)*(1+(-1)^k)*sin(k*2*aphi);
end
%---------------------------------------------------------------------------
% section solu - combine submat - solv matrix eqn - find results
% find the vector of displacement components rslt
stiff=[aii aim ain; ami amm amn; ani anm ann];
clear aii;  clear aim; clear ain; clear ami;  clear amm; clear amn;
clear ani; clear anm; clear ann; clear pval; clear tval;
rslt=stiff\rhss; clear stiff; clear rhss;
%
% displacement components at the sampling points, u,v,w
for i=1:m; for j=1:n
iju=(i-1)*n+j;  ijv=m*n+(i-1)*n+j; ijw=2*m*n+(i-1)*n+j;
u(i,j)=rslt(iju);  v(i,j)=rslt(ijv); w(i,j)=rslt(ijw);
end;  end
%---------------------------------------------------------------------------
for i=1:m; for j=1:n
ut(i,j)=ut(i,j)+u(i,j)*cos(2*k*theta);
vt(i,j)=vt(i,j)+v(i,j)*cos(2*k*theta);
wt(i,j)=wt(i,j)+w(i,j)*sin(2*k*theta);
wt1(i,j)=wt1(i,j)+w(i,j)*cos(2*k*theta)*2*k;
end; end
end
%---------------------------------------------------------------------------
% stress components at the sampling points, srr, shh, stt, srh
% loop over samling points in the radial direction
for i=1:m
s=x(i); s1=1/s;
```

```
%
% loop over sampling points in the merid direction
for j=1:n
tee=tet(j);  ct=cos(tee);  st=sin(tee);
cp=rarr+s*ct;  cp1=1/cp;
% terms which do not involve derivatives
srr=(a9*s1+cp1*ct*a13)*ut(i,j)-cp1*a13*st*vt(i,j)+cp1*a13*wt1(i,j);
shh=(a13*s1+cp1*ct*a6)*ut(i,j)-cp1*a6*st*vt(i,j)+cp1*a6*wt1(i,j);
% loop for terms involving the r derivatives
for ii=1:m
srr=srr+a7*b1(i,ii)*ut(ii,j);
shh=shh+a8*b1(i,ii)*ut(ii,j);
end
% loop for terms involving the phi derivatives
for jj=1:n
srr=srr+s1*a9*c1(j,jj)*vt(i,jj);
shh=shh+s1*a13*c1(j,jj)*vt(i,jj);
end
sr(i,j)=srr; sh(i,j)=shh;
end;  end
%---------------------------------------------------------------------------------
% result output
thetao=linspace(0,2*pi,200);
rl=115+0*sin(thetao);
polar(thetao,rl);
hold on
rh=85+0*sin(thetao);
polar(thetao,rh);
hold on
[theta1,rho]=meshgrid((0:360/28:360)*pi/180, 85:30/7:115);
[X, Y]=pol2cart(theta1,rho);
sr1=zeros(8,29);
for i=1:8
for j=1:28
sr1(i,j)=sh(i,j);
end
end
sr1(:,29)=sh(:,1);
polar([0 2*pi],[85 115]);
hold on
contour(X,Y,sr1)
```

APPENDIX B - Listing of program Semioml.m

```
% Semioml.m last modification - June 2, 2001
% vibration of isotropic toroids using semi-analytical DQM - steel
% 3D elasticity theory using 3 gov. eqns Int. J. Pres. Ves. & Piping 51 (1992)
%--------------------------------------------------------------------------------
% section input - input the problem parameters
% m=sampling points in radial direction; n1=trucation integer;
% n=sampling points in phi direction (even)
% pr=Poisson ratio;
% E=young's modulus (MPa); G=shear moduli (MPa);
% D=coefficient of material; p=pressure (MPa);
% rs=x-sect radius (mm); rarr=bend radius (mm);
% r1=outside radius (mm); r2=inside radius (mm);
% aphi=semi-angular length of an entire band;
% theta=the angle on circumferential; load=loading case;
% format long
m=12
n=48
k=1
rarr=600; r1=115; r2=85; E=207000; pr=0.3; pr1=1/pr;
G=E*pr1/(2*(pr1+1)); F=2*G/(pr1-2);
D=[F+2*G, F, F, 0, 0, 0; F, F+2*G, F, 0, 0, 0; F, F, F+2*G, 0, 0, 0;
 0, 0, 0, 2*G, 0, 0; 0, 0, 0, 0, 2*G, 0; 0, 0, 0, 0, 0, 2*G];
xdif=r1-r2; a1=D(1,1); a2=D(4,4)/2; a3=D(5,5)/2; a4=D(1,3)-D(2,3);
a5=D(1,2)-D(2,3); a6=D(3,3); a7=D(1,2); a8=D(1,3); a9=D(2,2);
 a10=D(1,2)-D(3,3); a11=D(1,3)-D(3,3);
a12=D(1,2)-D(1,3); a13=D(2,3); a14=D(6,6)/2;
%
%--------------------------------------------------------------------------------
% section wcrd - find weighting coeffts for radial direction
% unequal spacing with one delta point at each end for radial direction
% output - b1(i,j),b2(i,j)- 1st to 2nd derivatives
%
for i=1:m; tt=pi*(i-1)/(m-1);y(i)=(1-cos(tt))/2; end
for i=1:m; x(i)=r2+y(i)*xdif; end
%
for i=1:m; pai(i)=1; for j=1:m
if i~=j; pai(i)=(x(i)-x(j))*pai(i); end
end;end
%
for i=1:m; for j=1:m
if i~=j; b1(i,j)=pai(i)/((x(i)-x(j))*pai(j)); end
end;end
%
for i=1:m; b1(i,i)=0; for j=1:m
if i~=j; b1(i,i)=b1(i,i)-b1(i,j); end
end;end
%
for i=1:m; for j=1:m
if i~=j; b2(i,j)=2*(b1(i,i)*b1(i,j)-b1(i,j)/(x(i)-x(j))); end
end;end
%
for i=1:m; b2(i,i)=0; for j=1:m
if i~=j; b2(i,i)=b2(i,i)-b2(i,j); end
end;end
%
```

```
%--------------------------------------------------------------------------
% section wcmd - find weighting coeffts for the meridional direction
% equal spacing with no delta points
% output - c1(i,j),c2(i,j) - 1st to 2nd derivatives
n2=n/2;
%
for i=1:n; tet(i)=2*pi*(i-1)/n; ib(i,1)=.5; ib(i,n2+1)=.5; end
%
for i=2:2:n; ib(i,n2+1)=-0.5; end
for i=2:n2; ib(1,i)=1; end
for i=n2+2:n; ib(1,i)=0; end
%
for i=2:n; for j=n2+2:n
ib(i,j)=sin(2*pi*(i-1)*(j-1-n2)/n);
end; end
%
for i=2:n; for j=2:n2
ib(i,j)=cos(2*pi*(i-1)*(j-1)/n);
end; end
%
ib=ib/n2;
%
for i=1:n; for j=1:n2+1
rr1(j)=(1-j)*sin((j-1)*tet(i));
rr2(j)=-1*((j-1)^2)*cos((j-1)*tet(i));
end
for j=n2+2:n
rr1(j)=(j-1-n2)*cos((j-1-n2)*tet(i));
rr2(j)=-1*((j-1-n2)^2)*sin((j-1-n2)*tet(i));
end
c1(i,:)=(ib*rr1')';
c2(i,:)=(ib*rr2')';
end
%--------------------------------------------------------------------------
% section domeqn - set up matrices enforcing the governing equations
%    aii  aim  ain     u      0
%    ami  amm  amn     v  =   0
%    ani  anm  ann     w      0
% output - aii(i,j),...,ann(i,j) - coefft matrices for displacements
mn=m*n;
%
aii(:,:)=zeros(mn);aim(:,:)=zeros(mn);ain(:,:)=zeros(mn);
ami(:,:)=zeros(mn);amm(:,:)=zeros(mn);amn(:,:)=zeros(mn);
ani(:,:)=zeros(mn);anm(:,:)=zeros(mn);ann(:,:)=zeros(mn);
%
% loop over sampling points in the r direction
for i=2:m-1
s=x(i);  s1=1/s;  s2=(s1)*(s1);
%
% loop over sampling points in the phi direction
for j=1:n
ir=(i-1)*n+j;
tee=tet(j); ct=cos(tee); st=sin(tee);
cp=rarr+s*ct; cp1=1/cp; cp2=(cp1)*(cp1);
%
% terms which do not involve derivatives
aii(ir,ir)=-(4*k*k*cp2)*a3-s2*a9+s1*ct*cp1*(a4+a5)-cp2*ct*ct*a6;
```

```
aim(ir,ir)=-cp1*st*(s1*a4-cp1*ct*a6-s1*a2);
ain(ir,ir)=2*k*cp1*(s1*a4-cp1*ct*a3-cp1*ct*a6);
ami(ir,ir)=-cp1*st*(s1*a9-cp1*ct*a6);
amm(ir,ir)=-4*k*k*cp2*a14-a2*s2-cp1*s1*ct*(a13+a2)-st*st*cp2*a6;
amn(ir,ir)=st*cp2*(a14+a6)*k*2;
ani(ir,ir)=-cp1*((ct*cp1+s1)*a3+s1*a13+ct*cp1*a6)*k*2;
anm(ir,ir)=st*cp2*(a14+a6)*k*2;
ann(ir,ir)=-cp2*(4*k*k*a6+st*st*a14)-ct*cp1*a3*(ct*cp1+s1);
%
% terms which involve the r derivatives
for ii=1:m
ic=(ii-1)*n+j;
aii(ir,ic)=aii(ir,ic)+(b2(i,ii)+(s1+cp1*ct)*b1(i,ii))*a1;
aim(ir,ic)=aim(ir,ic)-cp1*st*(a8+a2)*b1(i,ii);
ain(ir,ic)=ain(ir,ic)+2*k*cp1*(a8+a3)*b1(i,ii);
ami(ir,ic)=ami(ir,ic)-cp1*st*a12*b1(i,ii);
amm(ir,ic)=amm(ir,ic)+a2*(b2(i,ii)+(s1+ct*cp1)*b1(i,ii));
ani(ir,ic)=ani(ir,ic)-2*k*cp1*(a3+a8)*b1(i,ii);
ann(ir,ic)=ann(ir,ic)+(b2(i,ii)+(ct*cp1+s1)*b1(i,ii))*a3;
%
% terms which involve mixed derivatives
for ij=1:n
ic=(ii-1)*n+ij;
aim(ir,ic)=aim(ir,ic)+s1*(a2+a7)*b1(i,ii)*c1(j,ij);
ami(ir,ic)=ami(ir,ic)+s1*(a2+a7)*b1(i,ii)*c1(j,ij);
end
end
% terms which involve the phi derivatives
for jj=1:n
ic=(i-1)*n+jj;
aii(ir,ic)=aii(ir,ic)+(s2*c2(j,jj)-cp1*s1*st*c1(j,jj))*a2;
aim(ir,ic)=aim(ir,ic)-s1*(a2*s1+s1*a9-cp1*ct*a5)*c1(j,jj);
ami(ir,ic)=ami(ir,ic)+s1*(s1*(a2+a9)+cp1*ct*(a13+a2))*c1(j,jj);
amm(ir,ic)=amm(ir,ic)+s1*a9*(s1*c2(j,jj)-st*cp1*c1(j,jj));
amn(ir,ic)=amn(ir,ic)+2*cp1*s1*(a13+a14)*k*c1(j,jj);
anm(ir,ic)=anm(ir,ic)-2*cp1*s1*(a14+a13)*k*c1(j,jj);
ann(ir,ic)=ann(ir,ic)+(s1*c2(j,jj)-st*cp1*c1(j,jj))*a14*s1;
end
end;end
%-----------------------------------------------------------------------------
% section boun - insert the boundary conditions
% pad loading applied only on out surface r=b
%
% pressurized conditions on r=r1 line
%              and
% zero loading canditions on r=r2 line
%
ival=[1,m];
% loop over sampling points in the radial direction
% only for i=1 and i=m
for bloop=1:2
i=ival(bloop);
s=x(i); s1=1/s;
% loop over sampling points in the merid direction
for j=1:n
tee=tet(j);  ct=cos(tee);  st=sin(tee);  Ap=rarr+s*ct;
Ap1=1/Ap;
```

```
ir=(i-1)*n+j;
% terms which do not involve derivatives
aii(ir,ir)=a7*s1+Ap1*ct*a8;
aim(ir,ir)=-Ap1*st*a8;
ain(ir,ir)=2*Ap1*a8*k;
amm(ir,ir)=-s1*a2;
ani(ir,ir)=-2*Ap1*k*a3;
ann(ir,ir)=-ct*Ap1*a3;
% loop for terms involving the r derivatives
for ii=1:m
ic=(ii-1)*n+j;
aii(ir,ic)=aii(ir,ic)+a1*b1(i,ii);
amm(ir,ic)=amm(ir,ic)+b1(i,ii)*a2;
ann(ir,ic)=ann(ir,ic)+b1(i,ii)*a3;
end
% loop for terms involving the phi derivatives
for jj=1:n
ic=(i-1)*n+jj;
aim(ir,ic)=aim(ir,ic)+a7*s1*c1(j,jj);
ami(ir,ic)=ami(ir,ic)+s1*c1(j,jj)*a2;
end
end; end
%---------------------------------------------------------------------------
% section partition - partition matrices for eigenvalue extraction
% this step necessary for freq analysis to form principle ev problem
% form bb matrix - bdy eqns involving bdy unknows
ar=1:n; br=(m-1)*n+1:m*n;
bb=[aii(ar,ar) aii(ar,br) aim(ar,ar) aim(ar,br) ain(ar,ar) ain(ar,br);
    aii(br,ar) aii(br,br) aim(br,ar) aim(br,br) ain(br,ar) ain(br,br);
    ami(ar,ar) ami(ar,br) amm(ar,ar) amm(ar,br) amn(ar,ar) amn(ar,br);
    ami(br,ar) ami(br,br) amm(br,ar) amm(br,br) amn(br,ar) amn(br,br);
    ani(ar,ar) ani(ar,br) anm(ar,ar) anm(ar,br) ann(ar,ar) ann(ar,br);
    ani(br,ar) ani(br,br) anm(br,ar) anm(br,br) ann(br,ar) ann(br,br)];
% form bd matrix - bdy eqns involving domain unknows
er=n+1:(m-1)*n;
bd=[aii(ar,er) aim(ar,er) ain(ar,er);
    aii(br,er) aim(br,er) ain(br,er);
    ami(ar,er) amm(ar,er) amn(ar,er);
    ami(br,er) amm(br,er) amn(br,er);
    ani(ar,er) anm(ar,er) ann(ar,er);
    ani(br,er) anm(br,er) ann(br,er)];
% form db matrix - domain eqns involving bdy unknows
db=[aii(er,ar) aii(er,br) aim(er,ar) aim(er,br) ain(er,ar) ain(er,br);
    ami(er,ar) ami(er,br) amm(er,ar) amm(er,br) amn(er,ar) amn(er,br);
    ani(er,ar) ani(er,br) anm(er,ar) anm(er,br) ann(er,ar) ann(er,br)];
% form dd matrix - domain eqns involving domain unknows
ar=n+1:(m-1)*n;
dd=[aii(ar,ar) aim(ar,ar) ain(ar,ar);
    ami(ar,ar) amm(ar,ar) amn(ar,ar);
    ani(ar,ar) anm(ar,ar) ann(ar,ar)];
clear aii aim ain ami amm amn ani anm ann;
%---------------------------------------------------------------------------
% section eign
prod1=-inv(bb)*bd; clear bb bd;
prod2=db*prod1; clear db prod1;
stiff=-(dd+prod2); clear dd prod2;
%Find eigenvalues
```

```
e=eig(stiff); clear stiff;
tmn=3*m*n-6*n;
kk=0;
for i=1:tmn
if imag(e(i)) == 0
if e(i) > 0; kk=kk+1; ezz(kk)=e(i); end
end
end
ezzz=ezz(1:kk);
% sort the eigenvalues and output the five lowest
sval=sort(ezzz);
for j=1:10; result(j)=sqrt(sval(j)/7770); end
%
end
result
```

APPENDIX B - Listing of program Semiom2.m

```
% Semiom2.m last modification - June 2, 2001
% vibration of isotropic Toroids using Semi-Analytical DQM - M2 material
% 3D elasticity theory using 3 gov. eqns. Int. J. Pres. Ves. & Piping 51 (1992)
%------------------------------------------------------------------------------
% section input - input the problem parameters
% m=sampling points in radial direction; n1=trucation integer;
% n=sampling points in phi direction (even)
% pr, pr1=Poisson ratio;
% E1, E=young's modulus (MPa); G, G1=shear moduli (MPa);
% D=coefficient of material; p=pressure (MPa);
% rs=x-sect radius (mm); rarr=bend radius (mm);
% r1=outside radius (mm); r2=inside radius (mm);
% aphi=semi-angular length of an entire band;
% theta=the angle on circumferential; load=loading case;
% format long
m=8
n=32
k=1
rarr=600; r1=115; r2=85; E=6.9e3; E1=172.38e3; G1=3.45e3; pr=0.25;
pr1=0.25; delta=(E1*(1-pr1*pr1)-2*E*pr1*pr1*(1+pr))/(E*E*E1*E1);
G=E/(2*(1+pr)); d11=(E1-E*pr1*pr1)/(E*E1*E1*delta);
d22=(E1-E*pr1*pr1)/(E*E1*E1*delta);  d33=(1-pr*pr)/(E*E*delta);
d12=(pr*E1+E*pr1*pr1)/(E*E1*E1*delta);
d13=(pr1*(1+pr))/(E*E1*delta); d23=(pr1*(1+pr))/(E*E1*delta);
d44=2*G; d55=2*G1; d66=2*G1;
D=[d11,d12,d13,0,0,0;d12,d22,d23,0,0,0;d13,d23,d33,0,0,0;0,0,0,d44,0,0;
0,0,0,0,d55,0;0,0,0,0,0,d66];
xdif=r1-r2; a1=D(1,1); a2=D(4,4)/2; a3=D(5,5)/2; a4=D(1,3)-D(2,3);
a5=D(1,2)-D(2,3);  a6=D(3,3); a7=D(1,2); a8=D(1,3); a9=D(2,2);
a10=D(1,2)-D(3,3); a11=D(1,3)-D(3,3);
a12=D(1,2)-D(1,3); a13=D(2,3); a14=D(6,6)/2;
%
%------------------------------------------------------------------------------
% section wcrd - find weighting coeffts for radial direction
% unequal spacing with one delta point at each end for radial direction
% output - b1(i,j),b2(i,j)- 1st to 2nd derivatives
%
for i=1:m; tt=pi*(i-1)/(m-1);y(i)=(1-cos(tt))/2; end
for i=1:m; x(i)=r2+y(i)*xdif; end
%
for i=1:m; pai(i)=1; for j=1:m
if i~=j; pai(i)=(x(i)-x(j))*pai(i); end
end;end
%
for i=1:m; for j=1:m
if i~=j; b1(i,j)=pai(i)/((x(i)-x(j))*pai(j)); end
end;end
%
for i=1:m; b1(i,i)=0; for j=1:m
if i~=j; b1(i,i)=b1(i,i)-b1(i,j); end
end;end
%
for i=1:m; for j=1:m
if i~=j; b2(i,j)=2*(b1(i,i)*b1(i,j)-b1(i,j)/(x(i)-x(j))); end
end;end
```

```
%
for i=1:m; b2(i,i)=0; for j=1:m
if i~=j; b2(i,i)=b2(i,i)-b2(i,j); end
end;end
%
%---------------------------------------------------------------------------
% section wcmd - find weighting coeffts for the meridional direction
% equal spacing with no delta points
% output - c1(i,j),c2(i,j) - 1st to 2nd derivatives
n2=n/2;
%
for i=1:n; tet(i)=2*pi*(i-1)/n; ib(i,1)=.5; ib(i,n2+1)=.5; end
%
for i=2:2:n; ib(i,n2+1)=-0.5; end
for i=2:n2; ib(1,i)=1; end
for i=n2+2:n; ib(1,i)=0; end
%
for i=2:n; for j=n2+2:n
ib(i,j)=sin(2*pi*(i-1)*(j-1-n2)/n);
end; end
%
for i=2:n; for j=2:n2
ib(i,j)=cos(2*pi*(i-1)*(j-1)/n);
end; end
%
ib=ib/n2;
%
for i=1:n; for j=1:n2+1
rr1(j)=(1-j)*sin((j-1)*tet(i));
rr2(j)=-1*((j-1)^2)*cos((j-1)*tet(i));
end
for j=n2+2:n
rr1(j)=(j-1-n2)*cos((j-1-n2)*tet(i));
rr2(j)=-1*((j-1-n2)^2)*sin((j-1-n2)*tet(i));
end
c1(i,:)=(ib*rr1')';
c2(i,:)=(ib*rr2')';
end
%---------------------------------------------------------------------------
% section domeqn - set up matrices enforcing the governing equations
%    aii  aim  ain     u      0
%    ami  amm  amn     v  =   0
%    ani  anm  ann     w      0
% output - aii(i,j),...,ann(i,j) - coefft matrices for displacements
mn=m*n;
%
aii(:,:)=zeros(mn);aim(:,:)=zeros(mn);ain(:,:)=zeros(mn);
ami(:,:)=zeros(mn);amm(:,:)=zeros(mn);amn(:,:)=zeros(mn);
ani(:,:)=zeros(mn);anm(:,:)=zeros(mn);ann(:,:)=zeros(mn);
%
% loop over sampling points in the r direction
for i=2:m-1
s=x(i);  s1=1/s;  s2=(s1)*(s1);
%
% loop over sampling points in the phi direction
for j=1:n
ir=(i-1)*n+j;
```

```
tee=tet(j); ct=cos(tee); st=sin(tee);
cp=rarr+s*ct; cp1=1/cp; cp2=(cp1)*(cp1);
%
% terms which do not involve derivatives
aii(ir,ir)=-(4*k*k*cp2)*a3-s2*a9+s1*ct*cp1*(a4+a5)-cp2*ct*ct*a6;
aim(ir,ir)=-cp1*st*(s1*a4-cp1*ct*a6-s1*a2);
ain(ir,ir)=2*k*cp1*(s1*a4-cp1*ct*a3-cp1*ct*a6);
ami(ir,ir)=-cp1*st*(s1*a9-cp1*ct*a6);
amm(ir,ir)=-4*k*k*cp2*a14-a2*s2-cp1*s1*ct*(a13+a2)-st*st*cp2*a6;
amn(ir,ir)=st*cp2*(a14+a6)*k*2;
ani(ir,ir)=-cp1*((ct*cp1+s1)*a3+s1*a13+ct*cp1*a6)*k*2;
anm(ir,ir)=st*cp2*(a14+a6)*k*2;
ann(ir,ir)=-cp2*(4*k*k*a6+st*st*a14)-ct*cp1*a3*(ct*cp1+s1);
%
% terms which involve the r derivatives
for ii=1:m
ic=(ii-1)*n+j;
aii(ir,ic)=aii(ir,ic)+(b2(i,ii)+(s1+cp1*ct)*b1(i,ii))*a1;
aim(ir,ic)=aim(ir,ic)-cp1*st*(a8+a2)*b1(i,ii);
ain(ir,ic)=ain(ir,ic)+2*k*cp1*(a8+a3)*b1(i,ii);
ami(ir,ic)=ami(ir,ic)-cp1*st*a12*b1(i,ii);
amm(ir,ic)=amm(ir,ic)+a2*(b2(i,ii)+(s1+ct*cp1)*b1(i,ii));
ani(ir,ic)=ani(ir,ic)-2*k*cp1*(a3+a8)*b1(i,ii);
ann(ir,ic)=ann(ir,ic)+(b2(i,ii)+(ct*cp1+s1)*b1(i,ii))*a3;
%
% terms which involve mixed derivatives
for ij=1:n
ic=(ii-1)*n+ij;
aim(ir,ic)=aim(ir,ic)+s1*(a2+a7)*b1(i,ii)*c1(j,ij);
ami(ir,ic)=ami(ir,ic)+s1*(a2+a7)*b1(i,ii)*c1(j,ij);
end
end
% terms which involve the phi derivatives
for jj=1:n
ic=(i-1)*n+jj;
aii(ir,ic)=aii(ir,ic)+(s2*c2(j,jj)-cp1*s1*st*c1(j,jj))*a2;
aim(ir,ic)=aim(ir,ic)-s1*(a2*s1+s1*a9-cp1*ct*a5)*c1(j,jj);
ami(ir,ic)=ami(ir,ic)+s1*(s1*(a2+a9)+cp1*ct*(a13+a2))*c1(j,jj);
amm(ir,ic)=amm(ir,ic)+s1*a9*(s1*c2(j,jj)-st*cp1*c1(j,jj));
amn(ir,ic)=amn(ir,ic)+2*cp1*s1*(a13+a14)*k*c1(j,jj);
anm(ir,ic)=anm(ir,ic)-2*cp1*s1*(a14+a13)*k*c1(j,jj);
ann(ir,ic)=ann(ir,ic)+(s1*c2(j,jj)-st*cp1*c1(j,jj))*a14*s1;
end
end;end
%---------------------------------------------------------------------------
% section boun - insert the boundary conditions
% zero loading conditions on r=r1 line
%              and
% zero loading conditions on r=r2 line
%
ival=[1,m];
% loop over sampling points in the radial direction
% only for i=1 and i=m
for bloop=1:2
i=ival(bloop);
s=x(i); s1=1/s;
% loop over sampling points in the merid direction
```

```
for j=1:n
tee=tet(j);  ct=cos(tee);  st=sin(tee);  Ap=rarr+s*ct;
Ap1=1/Ap;
ir=(i-1)*n+j;
% terms which do not involve derivatives
aii(ir,ir)=a7*s1+Ap1*ct*a8;
aim(ir,ir)=-Ap1*st*a8;
ain(ir,ir)=2*Ap1*a8*k;
amm(ir,ir)=-s1*a2;
ani(ir,ir)=-2*Ap1*k*a3;
ann(ir,ir)=-ct*Ap1*a3;
% loop for terms involving the r derivatives
for ii=1:m
ic=(ii-1)*n+j;
aii(ir,ic)=aii(ir,ic)+a1*b1(i,ii);
amm(ir,ic)=amm(ir,ic)+b1(i,ii)*a2;
ann(ir,ic)=ann(ir,ic)+b1(i,ii)*a3;
end
% loop for terms involving the phi derivatives
for jj=1:n
ic=(i-1)*n+jj;
aim(ir,ic)=aim(ir,ic)+a7*s1*c1(j,jj);
ami(ir,ic)=ami(ir,ic)+s1*c1(j,jj)*a2;
end
end; end
%-----------------------------------------------------------------------------
% section partition - partition matrices for eigenvalue extraction
% this step necessary for freq analysis to form principle ev problem
% form bb matrix - bdy eqns involving bdy unknows
ar=1:n; br=(m-1)*n+1:m*n;
bb=[aii(ar,ar) aii(ar,br) aim(ar,ar) aim(ar,br) ain(ar,ar) ain(ar,br);
    aii(br,ar) aii(br,br) aim(br,ar) aim(br,br) ain(br,ar) ain(br,br);
    ami(ar,ar) ami(ar,br) amm(ar,ar) amm(ar,br) amn(ar,ar) amn(ar,br);
    ami(br,ar) ami(br,br) amm(br,ar) amm(br,br) amn(br,ar) amn(br,br);
    ani(ar,ar) ani(ar,br) anm(ar,ar) anm(ar,br) ann(ar,ar) ann(ar,br);
    ani(br,ar) ani(br,br) anm(br,ar) anm(br,br) ann(br,ar) ann(br,br)];
% form bd matrix - bdy eqns involving domain unknows
er=n+1:(m-1)*n;
bd=[aii(ar,er) aim(ar,er) ain(ar,er);
    aii(br,er) aim(br,er) ain(br,er);
    ami(ar,er) amm(ar,er) amn(ar,er);
    ami(br,er) amm(br,er) amn(br,er);
    ani(ar,er) anm(ar,er) ann(ar,er);
    ani(br,er) anm(br,er) ann(br,er)];
% form db matrix - domain eqns involving bdy unknows
db=[aii(er,ar) aii(er,br) aim(er,ar) aim(er,br) ain(er,ar) ain(er,br);
    ami(er,ar) ami(er,br) amm(er,ar) amm(er,br) amn(er,ar) amn(er,br);
    ani(er,ar) ani(er,br) anm(er,ar) anm(er,br) ann(er,ar) ann(er,br)];
% form dd matrix - domain eqns involving domain unknows
ar=n+1:(m-1)*n;
dd=[aii(ar,ar) aim(ar,ar) ain(ar,ar);
    ami(ar,ar) amm(ar,ar) amn(ar,ar);
    ani(ar,ar) anm(ar,ar) ann(ar,ar)];
clear aii aim ain ami amm amn ani anm ann;
%-----------------------------------------------------------------------------
% section eign
prod1=-inv(bb)*bd; clear bb bd;
```

```
prod2=db*prod1; clear db prod1;
stiff=-(dd+prod2); clear dd prod2;
%Find eigenvalues
e=eig(stiff); clear stiff;
tmn=3*m*n-6*n;
kk=0;
for i=1:tmn
if imag(e(i)) == 0
if e(i) > 0; kk=kk+1; ezz(kk)=e(i); end
end
end
ezzz=ezz(1:kk);
% sort the eigenvalues and output the five lowest
sval=sort(ezzz);
for j=1:10; result(j)=sval(j); end
%
end
result
```

APPENDIX C – Listing of coefficients $f_1 - f_{41}$

$$f_1 = -(\frac{E_1 h\varsigma}{r(1-\nu_{12}\nu_{21})} + \frac{1}{12}\frac{E_1 h^3\varsigma}{r^3(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}hr\varsigma}$$

$$f_2 = -((\frac{E_1\varsigma}{r(1-\nu_{12}\nu_{21})} + \frac{1}{4}\frac{E_1\varsigma h^2}{r^3(1-\nu_{12}\nu_{21})})\frac{\partial h}{\partial\phi} - \frac{E_1 h\sin\phi}{R(1-\nu_{12}\nu_{21})}$$
$$-\frac{1}{12}\frac{E_1\sin\phi h^3}{Rr^2(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}hr\varsigma}$$

$$f_3 = -(\frac{G_{12}h}{r\varsigma} + \frac{1}{24}\frac{G_{12}h^3}{r^3\varsigma^2} + \frac{1}{24}\frac{G_{12}h^3}{r^3\varsigma} + \frac{1}{96}\frac{G_{12}h^3}{r^3\varsigma^3})\frac{1}{\hat{\rho}hr\varsigma}$$

$$f_4 = -((-\frac{\nu_{21}E_1\sin\phi}{R(1-\nu_{12}\nu_{21})} - \frac{1}{4}\frac{\nu_{21}E_1\sin\phi h^2}{Rr^2})\frac{\partial h}{\partial\phi} - \frac{\nu_{21}E_1 h\sin^2\phi}{R^2\varsigma\nu_{12}(1-\nu_{12}\nu_{21})}$$
$$-\frac{\nu_{21}E_1 h\cos\phi}{R(1-\nu_{12}\nu_{21})} - \frac{1}{12}\frac{\nu_{21}E_1 h^3\cos\phi}{r^2R(1-\nu_{12}\nu_{21})}(1+\frac{1}{\nu_{12}}))\frac{1}{\hat{\rho}hr\varsigma}$$

$$f_5 = -(\frac{G_{12}h}{r} + \frac{1}{24}\frac{G_{12}h^3}{r^3}(1-\frac{1}{\varsigma}) - \frac{1}{32}\frac{G_{12}h^3}{r^3\varsigma^2}$$
$$+\frac{\nu_{21}E_1 h}{r(1-\nu_{12}\nu_{21})} + \frac{1}{12}\frac{\nu_{21}E_1 h^3\cos\phi}{Rr^2\varsigma(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}hr\varsigma}$$

$$f_6 = -(\frac{\nu_{21}E_1}{r(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi} + \frac{\nu_{21}E_1 h\sin\phi}{\nu_{12}R\varsigma(1-\nu_{12}\nu_{21})} + \frac{G_{12}h\sin\phi}{R\varsigma} - \frac{1}{24}\frac{G_{12}h^3\sin\phi}{Rr^2\varsigma}$$
$$+\frac{1}{96}\frac{G_{12}h^3\sin\phi}{Rr^2\varsigma^3} - \frac{1}{12}\frac{\nu_{21}E_1 h^3\sin\phi}{Rr^2\varsigma(1-\nu_{12}\nu_{21})} + \frac{1}{24}\frac{\nu_{21}E_1 h^3\sin 2\phi}{R^2r\varsigma^2(1-\nu_{12}\nu_{21})}(1+\frac{1}{\nu_{12}})$$
$$+\frac{1}{48}\frac{G_{12}h^3\sin 2\phi}{R^2r\varsigma^3} + \frac{1}{24}\frac{G_{12}h^3\sin 2\phi}{R^2r\varsigma^2} + \frac{1}{4}\frac{\nu_{21}E_1 h^2\cos\phi}{Rr^2\varsigma(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}hr\varsigma}$$

$$f_7 = \frac{1}{12}\frac{E_1 h^3\varsigma}{r^3(1-\nu_{12}\nu_{21})}\frac{1}{\hat{\rho}hr\varsigma}$$

$$f_8 = -(-\frac{1}{4}\frac{E_1\varsigma h^2}{r^3(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi} + \frac{1}{12}\frac{E_1 h^3\sin\phi}{Rr^2(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}hr\varsigma}$$

$$f_9 = -(\frac{\nu_{21}E_1 h\cos\phi}{R(1-\nu_{12}\nu_{21})} + \frac{E_1 h\varsigma}{r(1-\nu_{12}\nu_{21})} + \frac{1}{4}\frac{\nu_{21}E_1 \sin\phi h^2}{Rr^2(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}$$
$$+\frac{1}{12}\frac{\nu_{21}E_1 h^3\cos\phi}{Rr^2(1-\nu_{12}\nu_{21})} + \frac{1}{12}\frac{\nu_{21}E_1 h^3\sin^2\phi}{R^3 r\varsigma\nu_{12}(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}hr\varsigma}$$

$$f_{10} = -(-\frac{1}{12}\frac{\nu_{21}E_1 h^3}{r^3\varsigma(1-\nu_{12}\nu_{21})} - \frac{1}{12}\frac{G_{12}h^3}{r^3\varsigma} - \frac{1}{24}\frac{G_{12}h^3}{r^3\varsigma^2})\frac{1}{\hat{\rho}hr\varsigma}$$

$$f_{11} = -(-\frac{1}{4}\frac{\nu_{21}E_1 h^2}{r^3\varsigma(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi} - \frac{1}{12}\frac{\nu_{21}E_1 h^3\sin\phi}{Rr^2\varsigma^2(1-\nu_{12}\nu_{21})}$$
$$-\frac{1}{12}\frac{G_{12}h^3}{Rr^2\varsigma^2} - \frac{1}{24}\frac{G_{12}h^3}{Rr^2\varsigma^3} - \frac{1}{12}\frac{\nu_{21}E_1 h^3\sin\phi}{\nu_{12}Rr^2\varsigma^2(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}hr\varsigma}$$

$$f_{12} = -(\frac{\nu_{21}E_1\cos\phi}{R(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi} + \frac{E_1\varsigma}{r(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}$$
$$-\frac{E_1 h\sin\phi}{R(1-\nu_{12}\nu_{21})} + \frac{\nu_{21}E_1 rh\sin\phi\cos\phi}{R^2\varsigma\nu_{12}(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}hr\varsigma}$$

$$f_{13} = (\frac{1}{12}\frac{\nu_{21}E_1 h^3\cos\phi}{Rr^2\varsigma(1-\nu_{12}\nu_{21})} + \frac{\nu_{21}E_1 h}{r(1-\nu_{12}\nu_{21})}$$
$$+\frac{1}{24}\frac{G_{12}h^3}{r^3} - \frac{1}{24}\frac{G_{12}h^3}{r^3\varsigma} - \frac{1}{32}\frac{G_{12}h^3}{r^3\varsigma^2} + \frac{G_{12}h}{r})\frac{1}{\hat{\rho}hr\varsigma}$$

$$f_{14} = -(-\frac{E_1\nu_{21}h\sin\phi}{R\varsigma\nu_{12}(1-\nu_{12}\nu_{21})} - \frac{G_{12}h\sin\phi}{R\varsigma} - \frac{1}{24}\frac{E_1\nu_{21}\sin 2\phi h^3}{R^2 r\varsigma^2\nu_{12}(1-\nu_{12}\nu_{21})}$$
$$+\frac{G_{12}}{r}\frac{\partial h}{\partial\phi} + \frac{1}{8}\frac{G_{12}h^2}{r^3}\frac{\partial h}{\partial\phi} - \frac{1}{8}\frac{G_{12}h^2}{r^3\varsigma}\frac{\partial h}{\partial\phi} - \frac{3}{32}\frac{G_{12}h^2}{r^3\varsigma^2}\frac{\partial h}{\partial\phi} + \frac{1}{24}\frac{G_{12}h^3\sin\phi}{Rr^2\varsigma}$$
$$-\frac{7}{96}\frac{G_{12}h^3\sin\phi}{Rr^2\varsigma^3} - \frac{1}{48}\frac{G_{12}h^3\sin 2\phi}{R^2 r\varsigma^3} - \frac{1}{24}\frac{G_{12}h^3\sin 2\phi}{R^2 r^2\varsigma^2} - \frac{1}{24}\frac{G_{12}h^3\sin\phi}{Rr^2\varsigma^2})\frac{1}{\hat{\rho}hr\varsigma}$$

$$f_{15} = -(\frac{G_{12}h\varsigma}{r} + \frac{1}{24}\frac{G_{12}h^3\varsigma}{r^3} - \frac{1}{8}\frac{G_{12}h^3}{r^3} + \frac{3}{32}\frac{G_{12}h^3}{r^3\varsigma})\frac{1}{\hat{\rho}hr\varsigma}$$

$$f_{16} = -(-\frac{G_{12}h\sin\phi}{R} + \frac{1}{8}\frac{G_{12}\varsigma h^2}{r^3}\frac{\partial h}{\partial\phi} - \frac{3}{8}\frac{G_{12}h^2}{r^3}\frac{\partial h}{\partial\phi} + \frac{9}{32}\frac{G_{12}h^2}{r^3\varsigma}\frac{\partial h}{\partial\phi}$$
$$+\frac{3}{32}\frac{G_{12}h^3\sin\phi}{Rr^2\varsigma^2} - \frac{1}{24}\frac{G_{12}h^3\sin\phi}{Rr^2} + \frac{G_{12}\varsigma}{r}\frac{\partial h}{\partial\phi})\frac{1}{\hat{\rho}r\varsigma h}$$

$$f_{17} = -(\frac{1}{12}\frac{\nu_{21}E_1 h^3\cos^2\phi}{R^2 r\varsigma^3\nu_{12}(1-\nu_{12}\nu_{21})} + \frac{\nu_{21}E_1 h}{r\varsigma\nu_{12}(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}hr\varsigma}$$

$$f_{18} = -(-\frac{G_{12}rh\sin^2\phi}{R^2\varsigma} + \frac{G_{12}h\cos\phi}{R} + \frac{1}{8}\frac{G_{12}h^2\sin 2\phi}{R^2 r\varsigma}\frac{\partial h}{\partial\phi}$$

$$-\frac{3}{16}\frac{G_{12}h^2\sin 2\phi}{Rr^2\varsigma^2}\frac{\partial h}{\partial\phi} - \frac{1}{8}\frac{G_{12}h^2\sin\phi}{Rr^2}\frac{\partial h}{\partial\phi} + \frac{1}{4}\frac{G_{12}h^2\sin\phi}{Rr^2\varsigma}\frac{\partial h}{\partial\phi}$$

$$-\frac{3}{32}\frac{G_{12}h^2\cos\phi}{Rr^2\varsigma^2}\frac{\partial h}{\partial\phi} + \frac{1}{12}\frac{G_{12}h^3\cos 2\phi}{R^2 r\varsigma} - \frac{1}{8}\frac{G_{12}h^3\cos 2\phi}{R^2 r\varsigma^2}$$

$$-\frac{1}{24}\frac{G_{12}h^3\cos\phi}{Rr^2} + \frac{1}{12}\frac{G_{12}h^3\cos\phi}{Rr^2\varsigma} - \frac{1}{32}\frac{G_{12}h^3\cos\phi}{Rr^2\varsigma^2}$$

$$-\frac{1}{24}\frac{G_{12}h^3\sin^2\phi}{R^2 r\varsigma} + \frac{G_{12}\sin\phi}{R}\frac{\partial h}{\partial\phi} - \frac{1}{24}\frac{G_{12}rh^3\sin^2 2\phi}{R^4\varsigma^3} + \frac{1}{8}\frac{G_{12}h^3\sin^2\phi}{R^2 r\varsigma^2}$$

$$-\frac{7}{96}\frac{G_{12}h^3\sin^2\phi}{R^2 r\varsigma^3} + \frac{1}{8}\frac{G_{12}h^3\sin\phi\sin 2\phi}{R^3\varsigma^2} - \frac{1}{6}\frac{G_{12}h^3\sin\phi\sin 2\phi}{R^3\varsigma^3})\frac{1}{\hat{\rho}hr\varsigma}$$

$$f_{19} = -(-\frac{1}{12}\frac{\nu_{21}E_1h^3\cos\phi}{Rr^2\varsigma(1-\nu_{12}\nu_{21})} - \frac{1}{12}\frac{G_{12}h^3\cos\phi}{Rr^2\varsigma} + \frac{1}{24}\frac{G_{12}h^3}{r^3\varsigma})\frac{1}{\hat{\rho}hr\varsigma}$$

$$f_{20} = -(\frac{1}{12}\frac{\nu_{21}E_1h^3\sin\phi\cos\phi}{\nu_{12}R^2 r\varsigma^2(1-\nu_{12}\nu_{21})} - \frac{1}{4}\frac{G_{12}h^2}{r^3}\frac{\partial h}{\partial\phi} + \frac{3}{8}\frac{G_{12}h^2}{r^3\varsigma}\frac{\partial h}{\partial\phi}$$

$$-\frac{1}{12}\frac{G_{12}h^3\sin\phi}{Rr^2\varsigma} - \frac{1}{12}\frac{G_{12}h^3}{Rr^2\varsigma} + \frac{1}{8}\frac{G_{12}h^3}{Rr^2\varsigma^2} + \frac{1}{12}\frac{G_{12}h^3\sin 2\phi}{R^2 r\varsigma^2} + \frac{1}{6}\frac{G_{12}h^3\sin\phi}{Rr^2\varsigma^2})\frac{1}{\hat{\rho}hr\varsigma}$$

$$f_{21} = \frac{1}{12}\frac{\nu_{21}E_1h^3\cos\phi}{Rr^2\varsigma^3\nu_{12}(1-\nu_{12}\nu_{21})}\frac{1}{\hat{\rho}hr\varsigma}$$

$$f_{22} = -(\frac{\nu_{21}E_1h}{r(1-\nu_{12}\nu_{21})} - \frac{1}{4}\frac{G_{12}h^2}{Rr^2\varsigma}\frac{\partial h}{\partial\phi} + \frac{3}{8}\frac{G_{12}h^2}{Rr^2\varsigma^2}\frac{\partial h}{\partial\phi} - \frac{1}{6}\frac{G_{12}h^3\sin\phi}{R^2 r\varsigma^2}$$

$$+\frac{\nu_{21}E_1h\cos\phi}{R\varsigma\nu_{12}(1-\nu_{12}\nu_{21})} + \frac{1}{12}\frac{G_{12}h^3\sin 2\phi}{R^3\varsigma^3} + \frac{7}{24}\frac{G_{12}h^3\sin\phi}{R^2 r}\varsigma^3)\frac{1}{\hat{\rho}hr\varsigma}$$

$$f_{23} = \frac{1}{12}\frac{E_1h^3}{r^4(1-\nu_{12}\nu_{21})}\frac{1}{\hat{\rho}h}$$

$$f_{24} = -(\frac{1}{12}\frac{\nu_{21}E_1h^3\sin\phi}{Rr^3\varsigma(1-\nu_{12}\nu_{21})} - \frac{1}{2}\frac{E_1h^2}{r^4(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}$$

$$-\frac{1}{12}\frac{\nu_{21}E_1h^3\sin\phi}{Rr^3\varsigma(1-\nu_{12}\nu_{21})} + \frac{1}{6}\frac{E_1h^3\sin\phi}{Rr^3\varsigma(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}h}$$

$$f_{25}=-(\frac{E_1h}{r^2(1-\nu_{12}\nu_{21})}+\frac{\nu_{21}E_1h\cos\phi}{Rr\varsigma(1-\nu_{12}\nu_{21})}+\frac{1}{12}\frac{\nu_{21}E_1h^3\cos\phi}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}$$

$$+\frac{1}{4}\frac{\nu_{21}E_1\sin\phi h^2}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}-\frac{1}{4}\frac{E_1h^2}{r^4(1-\nu_{12}\nu_{21})}\frac{\partial^2h}{\partial\phi^2}-\frac{1}{2}\frac{E_1h}{r^4(1-\nu_{12}\nu_{21})}(\frac{\partial h}{\partial\phi})^2$$

$$+\frac{1}{12}\frac{\nu_{21}E_1h^3\sin^2\phi}{R^2r^2\varsigma^2\nu_{12}(1-\nu_{12}\nu_{21})}+\frac{1}{2}\frac{E_1\sin\phi h^2}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}+\frac{1}{12}\frac{E_1h^3\cos\phi}{Rr^3\varsigma(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}h}$$

$$f_{26}=(\frac{1}{12}\frac{G_{12}h^3}{r^4\varsigma^2}+\frac{1}{24}\frac{G_{12}h^3}{r^4\varsigma^3}+\frac{1}{12}\frac{\nu_{21}E_1h^3}{r^4\varsigma^2(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}h}$$

$$f_{27}=(\frac{1}{4}\frac{G_{12}h^2}{r^4\varsigma^2}\frac{\partial h}{\partial\phi}+\frac{1}{8}\frac{G_{12}h^3}{r^4\varsigma^3}\frac{\partial h}{\partial\phi}+\frac{1}{24}\frac{G_{12}h^3\sin\phi}{Rr^3\varsigma^4}+\frac{1}{12}\frac{\nu_{21}E_1h^3\sin\phi}{Rr^3\varsigma^3\nu_{12}(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}h}$$

$$f_{28}=(\frac{\nu_{21}E_1h\sin\phi}{Rr\varsigma(1-\nu_{12}\nu_{21})}+\frac{\nu_{21}E_1h\sin\phi\cos\phi}{R^2\varsigma^2\nu_{12}(1-\nu_{12}\nu_{21})}-\frac{1}{2}\frac{\nu_{21}E_1h\sin\phi}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}(\frac{\partial h}{\partial\phi})^2$$

$$-\frac{1}{4}\frac{\nu_{21}E_1h^2\sin\phi}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}\frac{\partial^2h}{\partial\phi^2}-\frac{1}{2}\frac{\nu_{21}E_1h^2}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}+\frac{1}{12}\frac{\nu_{21}E_1h^3\sin\phi}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}$$

$$-\frac{1}{4}\frac{\nu_{21}E_1h^2\sin^2\phi}{R^2r^2\varsigma^2\nu_{12}(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}-\frac{1}{6}\frac{\nu_{21}E_1h^3\sin\phi\cos\phi}{R^2r^2\varsigma^2\nu_{12}(1-\nu_{12}\nu_{21})}-\frac{1}{12}\frac{\nu_{21}E_1h^3\sin^3\phi}{R^3r\varsigma^3\nu_{12}(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}h}$$

$$f_{29}=(\frac{1}{12}\frac{G_{12}h^3}{r^4\varsigma}-\frac{1}{8}\frac{G_{12}h^3}{r^4\varsigma^2}+\frac{1}{12}\frac{\nu_{21}E_1h^3\cos\phi}{Rr^3\varsigma^2(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}h}$$

$$f_{30}=(-\frac{1}{6}\frac{G_{12}h^3\sin\phi}{Rr^3\varsigma}+\frac{1}{24}\frac{G_{12}h^3\sin\phi}{Rr^3\varsigma^3}+\frac{1}{4}\frac{G_{12}h^2}{r^4\varsigma}\frac{\partial h}{\partial\phi}-\frac{3}{8}\frac{G_{12}h^2}{r^4\varsigma^2}\frac{\partial h}{\partial\phi}$$

$$+\frac{1}{12}\frac{G_{12}h^3\sin2\phi}{R^2r^2\varsigma^3}+\frac{1}{2}\frac{\nu_{21}E_1h^2\cos\phi}{Rr^3\varsigma^2}\frac{\partial h}{\partial\phi}-\frac{1}{6}\frac{\nu_{21}E_1h^3\sin\phi}{Rr^3\varsigma^2(1-\nu_{12}\nu_{21})}$$

$$+\frac{1}{24}\frac{\nu_{21}E_1h^3\sin2\phi}{\nu_{12}R^2r^2\varsigma^3(1-\nu_{12}\nu_{21})}+\frac{1}{12}\frac{\nu_{21}E_1h^3\sin2\phi}{R^2r^2\varsigma^3(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}h}$$

$$f_{31}=\frac{1}{12}\frac{\nu_{21}E_1h^3\cos\phi}{Rr^3\varsigma^4\nu_{12}(1-\nu_{12}\nu_{21})}\frac{1}{\hat{\rho}h}$$

$$f_{33}=-\frac{1}{12}\frac{E_1h^3}{r^4(1-\nu_{12}\nu_{21})}\frac{1}{\hat{\rho}h}$$

$$f_{34}=(-\frac{1}{2}\frac{E_1h^2}{r^4(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}+\frac{1}{6}\frac{E_1\sin\phi h^3\sin\phi}{Rr^3\varsigma(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}h}$$

$$f_{35}=-(-\frac{1}{12}\frac{\nu_{21}E_1h^3\cos\phi}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}-\frac{1}{4}\frac{\nu_{21}E_1\sin\phi h^2}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}+\frac{1}{4}\frac{E_1h^2}{r^4(1-\nu_{12}\nu_{21})}\frac{\partial^2 h}{\partial\phi^2}$$
$$+\frac{1}{2}\frac{E_1h}{r_4(1-\nu_{12}\nu_{21})}(\frac{\partial h}{\partial\phi})^2-\frac{1}{12}\frac{\nu_{21}E_1h^3\sin^2\phi}{R^2r^2\varsigma^2\nu_{12}(1-\nu_{12}\nu_{21})}$$
$$-\frac{1}{2}\frac{E_1\sin\phi h^2}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}-\frac{1}{12}\frac{E_1h^3\cos\phi}{Rr^3\varsigma(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}h}$$

$$f_{36}=(\frac{1}{12}\frac{\nu_{21}E_1h^3\sin\phi}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}-\frac{1}{2}\frac{\nu_{21}E_1\sin\phi t}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}(\frac{\partial h}{\partial\phi})^2-\frac{1}{2}\frac{\nu_{21}E_1\cos\phi h^2}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}$$
$$-\frac{1}{4}\frac{\nu_{21}E_1\sin\phi h^2}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}-\frac{1}{12}\frac{\nu_{21}E_1h^3\sin^3\phi}{R^3r\varsigma^3\nu_{12}(1-\nu_{12}\nu_{21})}$$
$$-\frac{1}{4}\frac{\nu_{21}E_1\sin^2\phi h^2}{R^2r^2\varsigma^2\nu_{12}(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}-\frac{1}{6}\frac{\nu_{21}E_1h^3\sin\phi\cos\phi}{R^2r^2\varsigma^2\nu_{12}(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}h}$$

$$f_{37}=(-\frac{1}{6}\frac{\nu_{21}E_1h^3}{r^4\varsigma^2(1-\nu_{12}\nu_{21})}-\frac{1}{6}\frac{G_{12}h^3}{r^4\varsigma^2})\frac{1}{\hat{\rho}h}$$

$$f_{38}=(-\frac{1}{2}\frac{\nu_{21}E_1h^2}{r^4\varsigma^2(1-\nu_{12}\nu_{21})}-\frac{1}{6}\frac{\nu_{21}E_1h^3\sin\phi}{Rr^3\varsigma^3(1-\nu_{12}\nu_{21})}-\frac{1}{2}\frac{G_{12}h^2}{r^4\varsigma^2}\frac{\partial h}{\partial\phi}-\frac{1}{6}\frac{G_{12}h^3}{Rr^3\varsigma^3})\frac{1}{\hat{\rho}h}$$

$$f_{39}=-\frac{1}{12}\frac{\nu_{21}E_1h^3}{r^4\varsigma^4\nu_{12}(1-\nu_{12}\nu_{21})}\frac{1}{\hat{\rho}h}$$

$$f_{40}=(-\frac{1}{6}\frac{G_{12}h^3\sin\phi}{R^2r^2\varsigma^4}-\frac{1}{2}\frac{G_{12}h^2}{Rr^3\varsigma^3}\frac{\partial h}{\partial\phi}-\frac{1}{2}\frac{\nu_{21}E_1h}{r^4\varsigma^2(1-\nu_{12}\nu_{21})}(\frac{\partial h}{\partial\phi})^2$$
$$-\frac{1}{2}\frac{\nu_{21}E_1h^2\sin\phi}{Rr^3\varsigma^3(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}-\frac{1}{4}\frac{\nu_{21}E_1h^2}{r^4\varsigma^2(1-\nu_{12}\nu_{21})}\frac{\partial^2 h}{\partial\phi^2}-\frac{1}{6}\frac{\nu_{21}E_1h^3\sin^2\phi}{R^2r^2\varsigma^4(1-\nu_{12}\nu_{21})}$$
$$-\frac{1}{12}\frac{\nu_{21}E_1h^3\cos\phi}{Rr^3\varsigma^3(1-\nu_{12}\nu_{21})}-\frac{1}{6}\frac{\nu_{21}E_1h^3\sin^2\phi}{R^2r^2\varsigma^4\nu_{12}(1-\nu_{12}\nu_{21})}-\frac{1}{4}\frac{\nu_{21}E_1h^2\sin\phi}{Rr^3\varsigma^3\nu_{12}(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}$$
$$-\frac{1}{12}\frac{\nu_{21}E_1h^3\cos\phi}{Rr^3\varsigma^3\nu_{12}(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}h}$$

$$f_{41}=(\frac{\nu_{21}E_1h\cos^2\phi}{R^2\varsigma^2\nu_{12}(1-\nu_{12}\nu_{21})}+\frac{2\nu_{21}E_1h\cos\phi}{Rr\varsigma(1-\nu_{12}\nu_{21})}+\frac{E_1h}{r^2(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}h}$$

APPENDIX C – Listing of coefficients $g_1 - g_{26}$

$$g_1 = \frac{1}{12}\frac{E_1 h^3}{r^3(1-\nu_{12}\nu_{21})}$$

$$g_2 = \frac{1}{4}\frac{E_1 h^2}{r^3(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi} - \frac{1}{12}\frac{\gamma E_1 h^2 \sin\phi}{r^3\varsigma(1-\nu_{12}\nu_{21})}$$

$$g_3 = \frac{G_{12}h^3}{24r^3\varsigma^3} + \frac{G_{12}h^3}{12r^3\varsigma^2}$$

$$g_4 = -\frac{1}{12}\frac{\gamma\nu_{21}E_1 h^3\sin^2\phi}{Rr^2\varsigma^2\nu_{12}(1-\nu_{12}\nu_{21})} + \frac{1}{4}\frac{\nu_{21}E_1 h^2\sin\phi}{Rr^2\varsigma(1-\nu_{12}\nu_{21})}$$

$$g_5 = \frac{1}{12}\frac{\nu_{21}E_1 h^3\cos\phi}{Rr^2\varsigma^2(1-\nu_{12}\nu_{21})} + \frac{1}{12}\frac{\gamma G_{12}h^3\cos\phi}{r^3\varsigma^2} - \frac{1}{24}\frac{G_{12}h^3}{r^3\varsigma^2}$$

$$g_6 = \frac{1}{4}\frac{\nu_{21}E_1 h^2\cos\phi}{Rr^2\varsigma^2(1-\nu_{12}\nu_{21})} + \frac{1}{12}\frac{\gamma^2 G_{12}h^3\sin 2\phi}{r^3\varsigma^3} + \frac{1}{24}\frac{\gamma G_{12}h^3\sin\phi}{r^3\varsigma^3}$$
$$-\frac{1}{12}\frac{\gamma G_{12}h^3\sin\phi}{r^3\varsigma^2} + \frac{1}{12}\frac{\gamma\nu_{21}E_1 h^3\sin\phi\cos\phi}{Rr^2\varsigma^3(1-\nu_{12}\nu_{21})} + \frac{1}{12}\frac{\gamma\nu_{21}E_1 h^3\sin\phi\cos\phi}{Rr^2\varsigma^3(1-\nu_{12}\nu_{21})}$$

$$g_7 = -\frac{1}{12}\frac{E_1 h^3}{r^3(1-\nu_{12}\nu_{21})}$$

$$g_8 = -\frac{1}{4}\frac{E_1 h^2}{r^3(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi} + \frac{1}{12}\frac{\gamma E_1 h^3\sin\phi}{r^3\varsigma(1-\nu_{12}\nu_{21})}$$

$$g_9 = \frac{1}{4}\frac{\nu_{21}E_1 h^2\sin\phi}{Rr^2\varsigma(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi} + \frac{1}{12}\frac{\gamma\nu_{21}E_1 h^3\sin^2\phi}{Rr^2\varsigma^3\nu_{12}(1-\nu_{12}\nu_{21})}$$

$$g_{10} = -\frac{1}{12}\frac{\nu_{21}E_1 h^3}{r^3\varsigma^2(1-\nu_{12}\nu_{21})} - \frac{1}{6}\frac{G_{12}h^3}{r^3\varsigma^2}$$

$$g_{11} = -\frac{1}{4}\frac{\nu_{21}E_1 h^2}{r^3\varsigma^2(1-\nu_{12}\nu_{21})} - \frac{1}{6}\frac{G_{12}h^3}{Rr^2\varsigma^3} - \frac{1}{12}\frac{\gamma\nu_{21}E_1 h^3\sin\phi}{r^3\varsigma^3\nu_{12}(1-\nu_{12}\nu_{21})}$$
$$-\frac{1}{12}\frac{\gamma\nu_{21}E_1 h^3\sin\phi}{r^3\varsigma^3(1-\nu_{12}\nu_{21})}$$

$$g_{12} = \frac{1}{2}\frac{G_{12}h}{r\varsigma} + \frac{1}{48}\frac{\gamma G_{12}h^3\cos\phi}{r^3\varsigma^3} + \frac{1}{24}\frac{\gamma G_{12}h^3\cos\phi}{r^3\varsigma}$$

$$g_{13} = \frac{1}{2}\frac{G_{12}h}{r} + \frac{1}{24}\frac{\gamma^2 G_{12}h^3\cos^2\phi}{r^3\varsigma^2} - \frac{1}{48}\frac{\gamma G_{12}h^3\cos\phi}{r^3\varsigma^2}$$

$$g_{14} = \frac{1}{2}\frac{\gamma G_{12}h\sin\phi}{r\varsigma} + \frac{1}{24}\frac{\gamma^3 G_{12}h^3\sin 2\phi\cos\phi}{r^3\varsigma^3}$$
$$+\frac{1}{48}\frac{\gamma^2 G_{12}h^3\sin\phi\cos\phi}{r^3\varsigma^3} - \frac{1}{24}\frac{\gamma^2 G_{12}h^3\sin\phi\cos\phi}{r^3\varsigma^2}$$

$$g_{15} = -\frac{1}{12}\frac{\gamma G_{12}h^3\cos\phi}{r^3\varsigma^2}$$

$$g_{16} = -\frac{1}{12}\frac{\gamma G_{12}h^3\cos\phi}{Rr^2\varsigma^3}$$

$$g_{17} = \frac{E_1 h}{r(1-\nu_{12}\nu_{21})}$$

$$g_{18} = -\frac{\gamma\nu_{21}E_1 h\sin\phi}{r\varsigma(1-\nu_{12}\nu_{21})}$$

$$g_{19} = \frac{\nu_{21}E_1 h}{r\varsigma(1-\nu_{12}\nu_{21})}$$

$$g_{20} = \frac{E_1 h}{r(1-\nu_{12}\nu_{21})} + \frac{\gamma\nu_{21}E_1 h\cos\phi}{r\varsigma(1-\nu_{12}\nu_{21})}$$

$$g_{21} = \frac{1}{12}\frac{E_1 h^3}{r^2(1-\nu_{12}\nu_{21})}$$

$$g_{22} = -\frac{1}{12}\frac{\nu_{21}E_1 h^3\sin\phi}{Rr\varsigma(1-\nu_{12}\nu_{21})}$$

$$g_{23} = \frac{1}{12}\frac{\nu_{21}E_1 h^3\cos\phi}{Rr\varsigma^2(1-\nu_{12}\nu_{21})}$$

$$g_{24} = -\frac{1}{12}\frac{E_1 h^3}{r^2(1-\nu_{12}\nu_{21})}$$

$$g_{25} = \frac{1}{12}\frac{\nu_{21}E_1 h^3\sin\phi}{Rr\varsigma(1-\nu_{12}\nu_{21})}$$

$$g_{26} = -\frac{1}{12}\frac{\nu_{21}E_1h^3}{r^2\varsigma^2(1-\nu_{12}\nu_{21})}$$

APPENDIX C – Listing of coefficients $h_1 - h_{29}$

$$h_1 = -\left(\frac{E_1 h\varsigma}{r(1-\nu_{12}\nu_{21})} + \frac{1}{12}\frac{E_1 h^3\varsigma}{r^3(1-\nu_{12}\nu_{21})}\right)\frac{1}{\hat{\rho}hr\varsigma}$$

$$h_2 = -\left(\frac{E_1\varsigma}{r(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi} - \frac{E_1 h\sin\phi}{R(1-\nu_{12}\nu_{21})} + \frac{1}{4}\frac{E_1\varsigma h^2}{r^3(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}\right.$$
$$\left.-\frac{1}{12}\frac{E_1\sin\phi h^3}{Rr^2(1-\nu_{12}\nu_{21})}\right)\frac{1}{\hat{\rho}hr\varsigma}$$

$$h_3 = \left(\frac{G_{12}h}{r\varsigma} + \frac{1}{24}\frac{G_{12}h^3}{r^3\varsigma^2} + \frac{1}{24}\frac{G_{12}h^3}{r^3\varsigma} + \frac{1}{96}\frac{G_{12}h^3}{r^3\varsigma^3}\right)\frac{(m\gamma)^2}{\hat{\rho}hr\varsigma}$$
$$-\left(-\frac{\nu_{21}E_1\sin\phi}{R(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi} - \frac{\nu_{21}E_1 h\cos\phi}{R(1-\nu_{12}\nu_{21})} - \frac{\nu_{21}E_1 h\sin^2\phi}{R^2\varsigma\nu_{12}(1-\nu_{12}\nu_{21})}\right.$$
$$\left.-\frac{1}{4}\frac{\nu_{21}E_1\sin\phi h^2}{Rr^2}\frac{\partial h}{\partial\phi} - \frac{1}{12}\frac{\nu_{21}E_1 h^3\cos\phi}{r^2R(1-\nu_{12}\nu_{21})} - \frac{1}{12}\frac{\nu_{21}E_1 h^3\cos\phi}{r^2R(1-\nu_{12}\nu_{21})}\right)\frac{1}{\hat{\rho}hr\varsigma}$$

$$h_4 = -\left(\frac{1}{12}\frac{\nu_{21}E_1 h^3\cos\phi}{Rr^2\varsigma(1-\nu_{12}\nu_{21})} + \frac{G_{12}h}{r} + \frac{1}{24}\frac{G_{12}h^3}{r^3} - \frac{1}{24}\frac{G_{12}h^3}{r^3\varsigma}\right.$$
$$\left.-\frac{1}{32}\frac{G_{12}h^3}{r^3\varsigma^2} + \frac{\nu_{21}E_1 h}{r(1-\nu_{12}\nu_{21})}\right)\frac{m\gamma}{\hat{\rho}hr\varsigma}$$

$$h_5 = -\left(\frac{\nu_{21}E_1}{r(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi} + \frac{\nu_{21}E_1 h\sin\phi}{\nu_{12}R\varsigma(1-\nu_{12}\nu_{21})} + \frac{1}{4}\frac{\nu_{21}E_1 h^2\cos\phi}{Rr^2\varsigma(1-\nu_{12}\nu_{21})}\right.$$
$$-\frac{1}{12}\frac{\nu_{21}E_1 h^3\sin\phi}{Rr^2\varsigma(1-\nu_{12}\nu_{21})} + \frac{1}{24}\frac{G_{12}h^3\sin 2\phi}{R^2r\varsigma^2} + \frac{G_{12}h\sin\phi}{R\varsigma} + \frac{1}{48}\frac{G_{12}h^3\sin 2\phi}{R^2r\varsigma^3}$$
$$-\frac{1}{24}\frac{G_{12}h^3\sin\phi}{Rr^2\varsigma} + \frac{1}{96}\frac{G_{12}h^3\sin\phi}{Rr^2\varsigma^3} + \frac{1}{24}\frac{\nu_{21}E_1 h^3\sin 2\phi}{R^2r\varsigma^2(1-\nu_{12}\nu_{21})}$$
$$\left.+\frac{1}{24}\frac{E_1 h^3\sin 2\phi}{\nu_{12}R^2r\varsigma^2(1-\nu_{12}\nu_{21})}\right)\frac{m\gamma}{\hat{\rho}hr\varsigma}$$

$$h_6 = \frac{1}{12}\frac{E_1 h^3\varsigma}{r^3(1-\nu_{12}\nu_{21})}\frac{1}{\hat{\rho}hr\varsigma}$$

$$h_7 = -\left(-\frac{1}{4}\frac{E_1\varsigma h^2}{r^3(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi} + \frac{1}{12}\frac{E_1 h^3\sin\phi}{Rr^2(1-\nu_{12}\nu_{21})}\right)\frac{1}{\hat{\rho}hr\varsigma}$$

$$h_8 = -(\frac{\nu_{21}E_1 h\cos\phi}{R(1-\nu_{12}\nu_{21})} + \frac{E_1 h\varsigma}{r(1-\nu_{12}\nu_{21})} + \frac{1}{4}\frac{\nu_{21}E_1 \sin\phi h^2}{Rr^2(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}$$
$$+\frac{1}{12}\frac{\nu_{21}E_1 h^3\cos\phi}{Rr^2(1-\nu_{12}\nu_{21})} + \frac{1}{12}\frac{\nu_{21}E_1 h^3\sin^2\phi}{R^3 r\varsigma\nu_{12}(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}hr\varsigma}$$
$$+(-\frac{1}{12}\frac{\nu_{21}E_1 h^3}{r^3\varsigma(1-\nu_{12}\nu_{21})} - \frac{1}{12}\frac{G_{12}h^3}{r^3\varsigma} - \frac{1}{24}\frac{G_{12}h^3}{r^3\varsigma^2})\frac{(m\gamma)^2}{\hat{\rho}hr\varsigma}$$

$$h_9 = (-\frac{1}{4}\frac{\nu_{21}E_1 h^2}{r^3\varsigma(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi} - \frac{1}{12}\frac{\nu_{21}E_1 h^3\sin\phi}{Rr^2\varsigma^2(1-\nu_{12}\nu_{21})}$$
$$-\frac{1}{12}\frac{G_{12}h^3}{Rr^2\varsigma^2} - \frac{1}{24}\frac{G_{12}h^3}{Rr^2\varsigma^3} - \frac{1}{12}\frac{\nu_{21}E_1 h^3\sin\phi}{\nu_{12}Rr^2\varsigma^2(1-\nu_{12}\nu_{21})})\frac{(m\gamma)^2}{\hat{\rho}hr\varsigma}$$
$$-(\frac{\nu_{21}E_1\cos\phi}{R(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi} + \frac{E_1\varsigma}{r(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}$$
$$-\frac{E_1 h\sin\phi}{R(1-\nu_{12}\nu_{21})} + \frac{\nu_{21}E_1 rh\sin\phi\cos\phi}{R^2\varsigma\nu_{12}(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}hr\varsigma}$$

$$h_{10} = -(\frac{1}{12}\frac{\nu_{21}E_1 h^3\cos\phi}{Rr^2\varsigma(1-\nu_{12}\nu_{21})} + \frac{\nu_{21}E_1 h}{r(1-\nu_{12}\nu_{21})}$$
$$+\frac{1}{24}\frac{G_{12}h^3}{r^3} - \frac{1}{24}\frac{G_{12}h^3}{r^3\varsigma} - \frac{1}{32}\frac{G_{12}h^3}{r^3\varsigma^2} + \frac{G_{12}h}{r})\frac{m\gamma}{\hat{\rho}hr\varsigma}$$

$$h_{11} = (-\frac{E_1\nu_{21}h\sin\phi}{R\varsigma\nu_{12}(1-\nu_{12}\nu_{21})} - \frac{G_{12}h\sin\phi}{R\varsigma} - \frac{1}{24}\frac{E_1\nu_{21}\sin 2\phi h^3}{R^2 r\varsigma^2\nu_{12}(1-\nu_{12}\nu_{21})}$$
$$+\frac{G_{12}}{r}\frac{\partial h}{\partial\phi} + \frac{1}{8}\frac{G_{12}h^2}{r^3}\frac{\partial h}{\partial\phi} - \frac{1}{8}\frac{G_{12}h^2}{r^3\varsigma}\frac{\partial h}{\partial\phi} - \frac{3}{32}\frac{G_{12}h^2}{r^3\varsigma^2}\frac{\partial h}{\partial\phi} + \frac{1}{24}\frac{G_{12}h^3\sin\phi}{Rr^2\varsigma}$$
$$-\frac{7}{96}\frac{G_{12}h^3\sin\phi}{Rr^2\varsigma^3} - \frac{1}{48}\frac{G_{12}h^3\sin 2\phi}{R^2 r\varsigma^3} - \frac{1}{24}\frac{G_{12}h^3\sin 2\phi}{R^2 r^2\varsigma^2} - \frac{1}{24}\frac{G_{12}h^3\sin\phi}{Rr^2\varsigma^2})\frac{m\gamma}{\hat{\rho}hr\varsigma}$$
$$h_{12} = -(\frac{G_{12}h\varsigma}{r} + \frac{1}{24}\frac{G_{12}h^3\varsigma}{r^3} - \frac{1}{8}\frac{G_{12}h^3}{r^3} + \frac{3}{32}\frac{G_{12}h^3}{r^3\varsigma})\frac{1}{\hat{\rho}hr\varsigma}$$
$$h_{13} = -(-\frac{G_{12}h\sin\phi}{R} + \frac{1}{8}\frac{G_{12}\varsigma h^2}{r^3}\frac{\partial h}{\partial\phi} - \frac{3}{8}\frac{G_{12}h^2}{r^3}\frac{\partial h}{\partial\phi} + \frac{9}{32}\frac{G_{12}h^2}{r^3\varsigma}\frac{\partial h}{\partial\phi}$$
$$+\frac{3}{32}\frac{G_{12}h^3\sin\phi}{Rr^2\varsigma^2} - \frac{1}{24}\frac{G_{12}h^3\sin\phi}{Rr^2} + \frac{G_{12}\varsigma}{r}\frac{\partial h}{\partial\phi})\frac{1}{\hat{\rho}r\varsigma h}$$

$$h_{14} = \left(\frac{1}{12}\frac{\nu_{21}E_1h^3\cos^2\phi}{R^2r\varsigma^3\nu_{12}(1-\nu_{12}\nu_{21})} + \frac{\nu_{21}E_1h}{r\varsigma\nu_{12}(1-\nu_{12}\nu_{21})}\right)\frac{(m\gamma)^2}{\hat{\rho}hr\varsigma}$$

$$-\left(-\frac{G_{12}rh\sin^2\phi}{R^2\varsigma} + \frac{G_{12}h\cos\phi}{R} + \frac{1}{8}\frac{G_{12}h^2\sin 2\phi}{R^2r\varsigma}\frac{\partial h}{\partial\phi}\right.$$

$$-\frac{3}{16}\frac{G_{12}h^2\sin 2\phi}{Rr^2\varsigma^2}\frac{\partial h}{\partial\phi} - \frac{1}{8}\frac{G_{12}h^2\sin\phi}{Rr^2}\frac{\partial h}{\partial\phi} + \frac{1}{4}\frac{G_{12}h^2\sin\phi}{Rr^2\varsigma}\frac{\partial h}{\partial\phi}$$

$$-\frac{3}{32}\frac{G_{12}h^2\cos\phi}{Rr^2\varsigma^2}\frac{\partial h}{\partial\phi} + \frac{1}{12}\frac{G_{12}h^3\cos 2\phi}{R^2r\varsigma} - \frac{1}{8}\frac{G_{12}h^3\cos 2\phi}{R^2r\varsigma^2}$$

$$-\frac{1}{24}\frac{G_{12}h^3\cos\phi}{Rr^2} + \frac{1}{12}\frac{G_{12}h^3\cos\phi}{Rr^2\varsigma} - \frac{1}{32}\frac{G_{12}h^3\cos\phi}{Rr^2\varsigma^2}$$

$$-\frac{1}{24}\frac{G_{12}h^3\sin^2\phi}{R^2r\varsigma} + \frac{G_{12}\sin\phi}{R}\frac{\partial h}{\partial\phi} - \frac{1}{24}\frac{G_{12}rh^3\sin^2 2\phi}{R^4\varsigma^3} + \frac{1}{8}\frac{G_{12}h^3\sin^2\phi}{R^2r\varsigma^2}$$

$$\left.-\frac{7}{96}\frac{G_{12}h^3\sin^2\phi}{R^2r\varsigma^3} + \frac{1}{8}\frac{G_{12}h^3\sin\phi\sin 2\phi}{R^3\varsigma^2} - \frac{1}{6}\frac{G_{12}h^3\sin\phi\sin 2\phi}{R^3\varsigma^3}\right)\frac{1}{\hat{\rho}hr\varsigma}$$

$$h_{15} = \left(-\frac{1}{12}\frac{\nu_{21}E_1h^3\cos\phi}{Rr^2\varsigma(1-\nu_{12}\nu_{21})} - \frac{1}{12}\frac{G_{12}h^3\cos\phi}{Rr^2\varsigma} + \frac{1}{24}\frac{G_{12}h^3}{r^3\varsigma}\right)\frac{m\gamma}{\hat{\rho}hr\varsigma}$$

$$h_{16} = \left(\frac{1}{12}\frac{\nu_{21}E_1h^3\sin\phi\cos\phi}{\nu_{12}R^2r\varsigma^2(1-\nu_{12}\nu_{21})} - \frac{1}{4}\frac{G_{12}h^2}{r^3}\frac{\partial h}{\partial\phi} + \frac{3}{8}\frac{G_{12}h^2}{r^3\varsigma}\frac{\partial h}{\partial\phi}\right.$$

$$\left.-\frac{1}{12}\frac{G_{12}h^3\sin\phi}{Rr^2\varsigma} - \frac{1}{12}\frac{G_{12}h^3}{Rr^2\varsigma} + \frac{1}{8}\frac{G_{12}h^3}{Rr^2\varsigma^2} + \frac{1}{12}\frac{G_{12}h^3\sin 2\phi}{R^2r\varsigma^2} + \frac{1}{6}\frac{G_{12}h^3\sin\phi}{Rr^2\varsigma^2}\right)\frac{m\gamma}{\hat{\rho}hr\varsigma}$$

$$h_{17} = \frac{1}{12}\frac{\nu_{21}E_1h^3\cos\phi}{Rr^2\varsigma^3\nu_{12}(1-\nu_{12}\nu_{21})}\frac{(m\gamma)^3}{\hat{\rho}hr\varsigma}$$

$$+\left(\frac{\nu_{21}E_1h}{r(1-\nu_{12}\nu_{21})} - \frac{1}{4}\frac{G_{12}h^2}{Rr^2\varsigma}\frac{\partial h}{\partial\phi} + \frac{3}{8}\frac{G_{12}h^2}{Rr^2\varsigma^2}\frac{\partial h}{\partial\phi} - \frac{1}{6}\frac{G_{12}h^3\sin\phi}{R^2r\varsigma^2}\right.$$

$$\left.+\frac{\nu_{21}E_1h\cos\phi}{R\varsigma\nu_{12}(1-\nu_{12}\nu_{21})} + \frac{1}{12}\frac{G_{12}h^3\sin 2\phi}{R^3\varsigma^3} + \frac{7}{24}\frac{G_{12}h^3\sin\phi}{R^2r}\varsigma^3\right)\frac{m\gamma}{\hat{\rho}hr\varsigma}$$

$$h_{18} = \frac{1}{12}\frac{E_1h^3}{r^4(1-\nu_{12}\nu_{21})}\frac{1}{\hat{\rho}h}$$

$$h_{19} = -\left(\frac{1}{12}\frac{\nu_{21}E_1h^3\sin\phi}{Rr^3\varsigma(1-\nu_{12}\nu_{21})} - \frac{1}{2}\frac{E_1h^2}{r^4(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}\right.$$

$$\left.-\frac{1}{12}\frac{\nu_{21}E_1h^3\sin\phi}{Rr^3\varsigma(1-\nu_{12}\nu_{21})} + \frac{1}{6}\frac{E_1h^3\sin\phi}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}\right)\frac{1}{\hat{\rho}h}$$

$$h_{20}=-(\frac{E_1 h}{r^2(1-\nu_{12}\nu_{21})}+\frac{\nu_{21}E_1 h\cos\phi}{Rr\varsigma(1-\nu_{12}\nu_{21})}+\frac{1}{12}\frac{\nu_{21}E_1 h^3\cos\phi}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}$$
$$+\frac{1}{4}\frac{\nu_{21}E_1\sin\phi h^2}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}-\frac{1}{4}\frac{E_1 h^2}{r^4(1-\nu_{12}\nu_{21})}\frac{\partial^2 h}{\partial\phi^2}-\frac{1}{2}\frac{E_1 h}{r^4(1-\nu_{12}\nu_{21})}(\frac{\partial h}{\partial\phi})^2$$
$$+\frac{1}{12}\frac{\nu_{21}E_1 h^3\sin^2\phi}{R^2r^2\varsigma^2\nu_{12}(1-\nu_{12}\nu_{21})}+\frac{1}{2}\frac{E_1\sin\phi h^2}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}+\frac{1}{12}\frac{E_1 h^3\cos\phi}{Rr^3\varsigma(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}h}$$
$$-(\frac{1}{12}\frac{G_{12}h^3}{r^4\varsigma^2}+\frac{1}{24}\frac{G_{12}h^3}{r^4\varsigma^3}+\frac{1}{12}\frac{\nu_{21}E_1 h^3}{r^4\varsigma^2(1-\nu_{12}\nu_{21})})\frac{(m\gamma)^2}{\hat{\rho}h}$$

$$h_{21}=-(\frac{1}{4}\frac{G_{12}h^2}{r^4\varsigma^2}\frac{\partial h}{\partial\phi}+\frac{1}{8}\frac{G_{12}h^3}{r^4\varsigma^3}\frac{\partial h}{\partial\phi}+\frac{1}{24}\frac{G_{12}h^3\sin\phi}{Rr^3\varsigma^4}+\frac{1}{12}\frac{\nu_{21}E_1 h^3\sin\phi}{Rr^3\varsigma^3\nu_{12}(1-\nu_{12}\nu_{21})})\frac{(m\gamma)^2}{\hat{\rho}h}$$
$$+(\frac{\nu_{21}E_1 h\sin\phi}{Rr\varsigma(1-\nu_{12}\nu_{21})}+\frac{\nu_{21}E_1 h\sin\phi\cos\phi}{R^2\varsigma^2\nu_{12}(1-\nu_{12}\nu_{21})}-\frac{1}{2}\frac{\nu_{21}E_1 h\sin\phi}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}(\frac{\partial h}{\partial\phi})^2$$
$$-\frac{1}{4}\frac{\nu_{21}E_1 h^2\sin\phi}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}\frac{\partial^2 h}{\partial\phi^2}-\frac{1}{2}\frac{\nu_{21}E_1 h^2}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}+\frac{1}{12}\frac{\nu_{21}E_1 h^3\sin\phi}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}$$

$$-\frac{1}{4}\frac{\nu_{21}E_1 h^2\sin^2\phi}{R^2r^2\varsigma^2\nu_{12}(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}-\frac{1}{6}\frac{\nu_{21}E_1 h^3\sin\phi\cos\phi}{R^2r^2\varsigma^2\nu_{12}(1-\nu_{12}\nu_{21})}-\frac{1}{12}\frac{\nu_{21}E_1 h^3\sin^3\phi}{R^3r\varsigma^3\nu_{12}(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}h}$$

$$h_{22}=(\frac{1}{12}\frac{G_{12}h^3}{r^4\varsigma}-\frac{1}{8}\frac{G_{12}h^3}{r^4\varsigma^2}+\frac{1}{12}\frac{\nu_{21}E_1 h^3\cos\phi}{Rr^3\varsigma^2(1-\nu_{12}\nu_{21})})\frac{m\gamma}{\hat{\rho}h}$$

$$h_{23}=(-\frac{1}{6}\frac{G_{12}h^3\sin\phi}{Rr^3\varsigma}+\frac{1}{24}\frac{G_{12}h^3\sin\phi}{Rr^3\varsigma^3}+\frac{1}{4}\frac{G_{12}h^2}{r^4\varsigma}\frac{\partial h}{\partial\phi}-\frac{3}{8}\frac{G_{12}h^2}{r^4\varsigma^2}\frac{\partial h}{\partial\phi}$$
$$+\frac{1}{12}\frac{G_{12}h^3\sin 2\phi}{R^2r^2\varsigma^3}+\frac{1}{2}\frac{\nu_{21}E_1 h^2\cos\phi}{Rr^3\varsigma^2}\frac{\partial h}{\partial\phi}-\frac{1}{6}\frac{\nu_{21}E_1 h^3\sin\phi}{Rr^3\varsigma^2(1-\nu_{12}\nu_{21})}$$
$$+\frac{1}{24}\frac{\nu_{21}E_1 h^3\sin 2\phi}{\nu_{12}R^2r^2\varsigma^3(1-\nu_{12}\nu_{21})}+\frac{1}{12}\frac{\nu_{21}E_1 h^3\sin 2\phi}{R^2r^2\varsigma^3(1-\nu_{12}\nu_{21})})\frac{m\gamma}{\hat{\rho}h}$$

$$h_{24}=-\frac{1}{12}\frac{\nu_{21}E_1 h^3\cos\phi}{Rr^3\varsigma^4\nu_{12}(1-\nu_{12}\nu_{21})}\frac{(m\gamma)^3}{\hat{\rho}h}-\frac{1}{12}\frac{E_1 h^3}{r^4(1-\nu_{12}\nu_{21})}\frac{m\gamma}{\hat{\rho}h}$$

$$h_{25}=-\frac{1}{12}\frac{E_1 h^3}{r^4(1-\nu_{12}\nu_{21})}\frac{1}{\hat{\rho}h}$$

$$h_{26}=(-\frac{1}{2}\frac{E_1 h^2}{r^4(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}+\frac{1}{6}\frac{E_1\sin\phi h^3\sin\phi}{Rr^3\varsigma(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}h}$$

$$h_{27}=-(-\frac{1}{12}\frac{\nu_{21}E_1h^3\cos\phi}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}-\frac{1}{4}\frac{\nu_{21}E_1\sin\phi h^2}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}+\frac{1}{4}\frac{E_1h^2}{r^4(1-\nu_{12}\nu_{21})}\frac{\partial^2 h}{\partial\phi^2}$$

$$+\frac{1}{2}\frac{E_1h}{r_4(1-\nu_{12}\nu_{21})}(\frac{\partial h}{\partial\phi})^2-\frac{1}{12}\frac{\nu_{21}E_1h^3\sin^2\phi}{R^2r^2\varsigma^2\nu_{12}(1-\nu_{12}\nu_{21})}$$

$$-\frac{1}{2}\frac{E_1\sin\phi h^2}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}-\frac{1}{12}\frac{E_1h^3\cos\phi}{Rr^3\varsigma(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}h}-(-\frac{1}{6}\frac{\nu_{21}E_1h^3}{r^4\varsigma^2(1-\nu_{12}\nu_{21})}-\frac{1}{6}\frac{G_{12}h^3}{r^4\varsigma^2})\frac{(m\gamma)^2}{\hat{\rho}h}$$

$$h_{28}=(\frac{1}{12}\frac{\nu_{21}E_1h^3\sin\phi}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}-\frac{1}{2}\frac{\nu_{21}E_1\sin\phi t}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}(\frac{\partial h}{\partial\phi})^2-\frac{1}{2}\frac{\nu_{21}E_1\cos\phi h^2}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}$$

$$-\frac{1}{4}\frac{\nu_{21}E_1\sin\phi h^2}{Rr^3\varsigma(1-\nu_{12}\nu_{21})}-\frac{1}{12}\frac{\nu_{21}E_1h^3\sin^3\phi}{R^3r\varsigma^3\nu_{12}(1-\nu_{12}\nu_{21})}$$

$$-\frac{1}{4}\frac{\nu_{21}E_1\sin^2\phi h^2}{R^2r^2\varsigma^2\nu_{12}(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}-\frac{1}{6}\frac{\nu_{21}E_1h^3\sin\phi\cos\phi}{R^2r^2\varsigma^2\nu_{12}(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}h}$$

$$-(-\frac{1}{2}\frac{\nu_{21}E_1h^2}{r^4\varsigma^2(1-\nu_{12}\nu_{21})}-\frac{1}{6}\frac{\nu_{21}E_1h^3\sin\phi}{Rr^3\varsigma^3(1-\nu_{12}\nu_{21})}-\frac{1}{2}\frac{G_{12}h^2}{r^4\varsigma^2}\frac{\partial h}{\partial\phi}-\frac{1}{6}\frac{G_{12}h^3}{Rr^3\varsigma^3})\frac{(m\gamma)^2}{\hat{\rho}h}$$

$$h_{29}=-\frac{1}{12}\frac{\nu_{21}E_1h^3}{r^4\varsigma^4\nu_{12}(1-\nu_{12}\nu_{21})}\frac{(m\gamma)^4}{\hat{\rho}h}$$

$$-(-\frac{1}{6}\frac{G_{12}h^3\sin\phi}{R^2r^2\varsigma^4}-\frac{1}{2}\frac{G_{12}h^2}{Rr^3\varsigma^3}\frac{\partial h}{\partial\phi}-\frac{1}{2}\frac{\nu_{21}E_1h}{r^4\varsigma^2(1-\nu_{12}\nu_{21})}(\frac{\partial h}{\partial\phi})^2$$

$$-\frac{1}{2}\frac{\nu_{21}E_1h^2\sin\phi}{Rr^3\varsigma^3(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}-\frac{1}{4}\frac{\nu_{21}E_1h^2}{r^4\varsigma^2(1-\nu_{12}\nu_{21})}\frac{\partial^2 h}{\partial\phi^2}-\frac{1}{6}\frac{\nu_{21}E_1h^3\sin^2\phi}{R^2r^2\varsigma^4(1-\nu_{12}\nu_{21})}$$

$$-\frac{1}{12}\frac{\nu_{21}E_1h^3\cos\phi}{Rr^3\varsigma^3(1-\nu_{12}\nu_{21})}-\frac{1}{6}\frac{\nu_{21}E_1h^3\sin^2\phi}{R^2r^2\varsigma^4\nu_{12}(1-\nu_{12}\nu_{21})}-\frac{1}{4}\frac{\nu_{21}E_1h^2\sin\phi}{Rr^3\varsigma^3\nu_{12}(1-\nu_{12}\nu_{21})}\frac{\partial h}{\partial\phi}$$

$$-\frac{1}{12}\frac{\nu_{21}E_1h^3\cos\phi}{Rr^3\varsigma^3\nu_{12}(1-\nu_{12}\nu_{21})})\frac{(m\gamma)^2}{\hat{\rho}h}+(\frac{\nu_{21}E_1h\cos^2\phi}{R^2\varsigma^2\nu_{12}(1-\nu_{12}\nu_{21})}+\frac{2\nu_{21}E_1h\cos\phi}{Rr\varsigma(1-\nu_{12}\nu_{21})}+\frac{E_1h}{r^2(1-\nu_{12}\nu_{21})})\frac{1}{\hat{\rho}h}$$

APPENDIX D - Listing of Freqisol.m

```
% Freqisol.m last modification - Feb 31, 2002
% Sanders shell theory using 3 gov eqns comp & struct 43 (1992) 1019-1028.
% natural frequencies and mode shapes of steel torus using dqm
% harmonic trial function used in meridional direction
% Fourier series expansion written in circumferential direction
% ----------------------------------------------------------------------------
% section input
% n=stations in meridional direction;
% m=stations in circumferential direction;
% E1=Young's modulus (Pa); G12=shear moduli (Pa);
% nu12=nu21=Poisson's ratio; den=mass density (kg/m3);
% r=x-section radius (m); R=bend radius (m);
% t1 t2=thickness coefficients
% format long
n=48
m=120
k=1/(0.762/0.25)
r=0.25; R=0.762;  E1=2.07e11; nu12=0.3; nu21=0.3; f1=E1/(1-nu21*nu12);
gammar=r/R; t1=0.009525; t2=0.001; G12=E1/(2*(1+nu12));  den=7770;
%-----------------------------------------------------------------------------
% section wcmd - find the weighting coeffts for meridional direction
% unequal spacing with one delta point at each end for phi direction
% output - b1(i,j),b2(i,j),b3(i,j),b4(i,j) for 1st to 4th derivats
%
n2=n/2;
for i=1:n
tet(i)=2*pi*(i-1)/n; ib(i,1)=.5; ib(i,n2+1)=.5;
end
%
for i=2:2:n; ib(i,n2+1)=-0.5; end
%
for i=2:n2; ib(1,i)=1; end
%
for i=n2+2:n; ib(1,i)=0; end
%
for i=2:n; for j=n2+2:n
ib(i,j)=sin(2*pi*(i-1)*(j-1-n2)/n);
end; end
%
for i=2:n; for j=2:n2
ib(i,j)=cos(2*pi*(i-1)*(j-1)/n);
end; end
%
ib=ib/n2;
for i=1:n; for j=1:n
if abs(ib(i,j))<1e-12; ib(i,j)=0; end
end;end
%
for i=1:n; for j=1:n2+1
rr1(j)=(1-j)*sin((j-1)*tet(i));
rr2(j)=-1*((j-1)^2)*cos((j-1)*tet(i));
rr3(j)=((j-1)^3)*sin((j-1)*tet(i));
rr4(j)=((j-1)^4)*cos((j-1)*tet(i));
end
for j=n2+2:n
```

```
rr1(j)=(j-1-n2)*cos((j-1-n2)*tet(i));
rr2(j)=-1*((j-1-n2)^2)*sin((j-1-n2)*tet(i));
rr3(j)=-1*((j-1-n2)^3)*cos((j-1-n2)*tet(i));
rr4(j)=((j-1-n2)^4)*sin((j-1-n2)*tet(i));
end
%
b1(i,:)=(ib*rr1')';
b2(i,:)=(ib*rr2')';
b3(i,:)=(ib*rr3')';
b4(i,:)=(ib*rr4')';
end
%---------------------------------------------------------------------------
% section domeqn - set up matrices enforcing the governing equations
%    aii  aim  ain    u      0
%    ami  amm  amn    v  =   0
%    ani  anm  ann    w      0
% output - aii(i,j),...,ann(i,j) - coefft matrices for displacements
%
aii(:,:)=zeros(n);aim(:,:)=zeros(n);ain(:,:)=zeros(n);
ami(:,:)=zeros(n);amm(:,:)=zeros(n);amn(:,:)=zeros(n);
ani(:,:)=zeros(n);anm(:,:)=zeros(n);ann(:,:)=zeros(n);
%
% loop over sampling points in the phi direction
for j=1:n
tee=tet(j); ct=cos(tee); st=sin(tee); st2=2*st*ct; ct2=(ct^2)-(st^2);
zeta=1+gammar*ct; t=t1-t2*ct;
% terms which do not involve derivatives
aii(j,j)=((-nu21*f1*(-t2*st)*gammar*st/r-nu21*f1*t*gammar*ct/r
-(gammar^2)*(st^2)*f1*t*nu21/(r*zeta*nu12)
-nu21*f1*(t^2)*(-t2*st)*st/(4*(r^2)*R)-f1*nu21*(t^3)*ct/(12*(r^2)*R)
-f1*(t^3)*gammar*nu21*(st^2)/(12*nu12*(r^2)*R*zeta))*r/t
-((G12*t/(r*zeta)+G12*(t^3)/(24*(r^3)*(zeta^2))+G12*(t^3)/(24*(r^3)*zeta)
+G12*(t^3)/(96*(r^3)*(zeta^3)))*(k^2))*r/t)/((r^2)*zeta);
aim(j,j)=(((f1*nu21*(-t2*st)/r+st*nu21*f1*t/(R*zeta*nu12)
+nu21*f1*(t^2)*ct*(-t2*st)/(4*(r^2)*R*zeta)
-nu21*f1*(t^3)*st/(12*(r^2)*R*zeta)
+G12*(t^3)*st2/(24*(R^2)*r*(zeta^2))+G12*t*st/(R*zeta)
+G12*(t^3)*st2/(48*(R^2)*r*(zeta^3))-
G12*(t^3)*st/(24*R*(r^2)*zeta)+G12*(t^3)*st/(96*R*(r^2)*(zeta^3))
+st2*nu21*f1*(t^3)/(24*(R^2)*r*(zeta^2))
+nu21*f1*(t^3)*st2/(24*(R^2)*r*nu12*(zeta^2)))*k)*r/t)/((r^2)*zeta);
ain(j,j)=((nu21*f1*(-t2*st)*gammar*ct/r+f1*(-t2*st)*zeta/r
-f1*t*gammar*st/r+f1*t*(gammar^2)*st*ct*nu21/(r*zeta*nu12))*r/t
-((-nu21*f1*(t^2)*(-t2*st)/(4*(r^3)*zeta)
-nu21*f1*(t^3)*st/(12*(r^2)*(zeta^2)*R)
-G12*(t^3)/(12*(r^2)*R*(zeta^2))
-G12*(t^3)/(24*(r^2)*R*(zeta^3))
-st*f1*nu21*(t^3)/(12*nu12*R*(r^2)*(zeta^2)))*(k^2))*r/t)/((r^2)*zeta);
ami(j,j)=(((-st*f1*nu21*t/(R*nu12*zeta)-G12*t*st/(R*zeta)
-f1*nu21*(st2)*(t^3)/(24*(R^2)*r*(zeta^2)*nu12)+G12*(-t2*st)/r
+G12*(t^2)*(-t2*st)/(8*(r^3))-G12*(t^2)*(-t2*st)/(8*(r^3)*zeta)
-3*G12*(t^2)*(-t2*st)/(32*(r^3)*(zeta^2))+G12*(t^3)*st/(24*(r^2)*R*zeta)
-7*G12*(t^3)*st/(96*(r^2)*R*(zeta^3))-st2*G12*(t^3)/(48*r*(R^2)*(zeta^3))
-st2*G12*(t^3)/(24*(r^2)*(R^2)*(zeta^2))
-st*G12*(t^3)/(24*(r^2)*R*(zeta^2)))*(-k))*r/t)/((r^2)*zeta);
amm(j,j)=((-2*G12*t*r*(st^2)/((R^2)*zeta)+G12*ct*t/R
+G12*t*r*(st^2)/((R^2)*zeta)+G12*(t^2)*st2*(-t2*st)/(8*(R^2)*r*zeta)
```

```
-3*G12*(t^2)*st2*(-t2*st)/(16*(R^2)*r*(zeta^2))
-G12*(t^2)*st*(-t2*st)/(8*R*(r^2))+G12*(t^2)*st*(-t2*st)/(4*R*(r^2)*zeta)
-3*G12*(t^2)*st*(-t2*st)/(32*R*(r^2)*(zeta^2))
+G12*(t^3)*ct2/(12*(R^2)*r*zeta)-G12*(t^3)*ct2/(8*(R^2)*r*(zeta^2))
-G12*(t^3)*ct/(24*(r^2)*R)+G12*(t^3)*ct/(12*(r^2)*R*zeta)
-G12*(t^3)*ct/(32*(r^2)*R*(zeta^2))-G12*(t^3)*(st^2)/(24*(R^2)*r*zeta)
+G12*st*(-t2*st)/R-r*(st2^2)*G12*(t^3)/(24*(R^4)*(zeta^3))
+G12*(t^3)*(st^2)/(8*(R^2)*r*(zeta^2))
-7*G12*(t^3)*(st^2)/(96*(R^2)*r*(zeta^3))
+G12*(t^3)*st*st2/(8*(R^3)*(zeta^2))-G12*(t^3)*st*st2/(6*(R^3)*(zeta^3)))*r/t
-((ct^2)*nu21*f1*(t^3)/(12*(R^2)*r*(zeta^3)*nu12)
+f1*nu21*t/(r*zeta*nu12))*(k^2)*r/t)/((r^2)*zeta);
amn(j,j)=(((nu21*f1*t/r-G12*(t^2)*(-t2*st)/(4*(r^2)*R*zeta)
+3*G12*(t^2)*(-t2*st)/(8*(r^2)*R*(zeta^2))-G12*(t^3)*st/(6*(R^2)*r*(zeta^2))
+f1*nu21*t*ct/(R*zeta*nu12)+G12*(t^3)*st2/(12*(R^3)*(zeta^3))
+7*G12*(t^3)*st/(24*(R^2)*r*(zeta^3)))*(-k)
+(-f1*nu21*(t^3)*ct/(12*R*(r^2)*nu12*(zeta^3)))*(k^3))*r/t)/((r^2)*zeta);
ani(j,j)=((-f1*t*nu21*gammar*st/((r^2)*zeta)
-f1*t*(gammar^2)*st*ct*nu21/((r^2)*(zeta^2)*nu12)
+f1*nu21*t*((t2*st)^2)*st/(2*(r^3)*R*zeta)
+nu21*f1*(t^2)*(-t2*ct)*st/(4*(r^3)*R*zeta)
+nu21*f1*(-t2*st)*(t^2)*ct/((r^3)*R*zeta*2)-f1*(t^3)*nu21*st/(12*(r^3)*R*zeta)
+f1*(t^2)*(-t2*st)*gammar*(st^2)*nu21/(4*(r^3)*(zeta^2)*R*nu12)
+f1*(t^3)*gammar*st*ct*nu21/(6*(r^3)*(zeta^2)*R*nu12)
+f1*(t^3)*nu21*(gammar^2)*(st^3)/(12*nu12*(r^3)*R*(zeta^3)))*(-(r^2)*zeta)/t
-((-G12*(t^2)*(-t2*st)/(4*(r^4)*(zeta^2))-G12*(t^2)*(-t2*st)/(8*(r^4)*(zeta^3))
-G12*(t^3)*st/(24*(r^3)*(zeta^4)*R)
+f1*nu21*(t^3)*st/(12*nu12*(r^2)*R*(zeta^3)))*(k^2))*(-
(r^2)*zeta)/t)/((r^2)*zeta);
anm(j,j)=(((nu21*f1*t/((r^2)*zeta)
-G12*(t^2)*st2*(-t2*st)/(4*(r^2)*(R^2)*(zeta^3))
+G12*(t^2)*st*(-t2*st)/(4*(r^3)*R*(zeta^2))
-G12*(t^2)*st*(-t2*st)/(8*(r^3)*R*(zeta^3))
-G12*(t^3)*ct2/(6*(r^2)*(R^2)*(zeta^3))
+G12*(t^3)*ct/(12*(r^3)*R*(zeta^2))-G12*(t^3)*ct/(24*(r^3)*R*(zeta^3)
)-G12*(t^3)*st*st2/(12*r*(R^3)*(zeta^4))
-G12*(t^3)*(st^2)/(24*(r^2)*(R^2)*(zeta^4))
-nu21*f1*t*ct*((t2*st)^2)/(2*(r^3)*R*(zeta^2))
+nu21*f1*(t^2)*st*(-t2*st)/(2*(r^3)*R*(zeta^2))
-nu21*f1*(t^2)*st2*(-t2*st)/(4*(r^2)*(R^2)*(zeta^3))
-nu21*f1*(t^2)*ct*(-t2*ct)/(4*(r^3)*R*(zeta^2))
+nu21*f1*(t^3)*ct/(12*(r^3)*R*(zeta^2))
-nu21*f1*(t^3)*ct*(st^2)/(6*r*(R^3)*(zeta^4))
-(ct^2)*nu21*f1*(t^3)/(12*(r^2)*(R^2)*(zeta^3))
-nu21*f1*(t^3)*(st^2)*ct/(6*r*(R^3)*(zeta^4)*nu12)
-nu21*f1*(t^2)*st2*(-t2*st)/(8*nu12*(r^2)*(R^2)*(zeta^3))
+nu21*f1*(t^3)*(st^2)/(6*(r^2)*(R^2)*(zeta^3))
+nu21*f1*ct*t/(nu12*r*R*(zeta^2))-
nu21*f1*(t^3)*ct2/(12*nu12*(r^2)*(R^2)*(zeta^3)))*k
-(-nu21*f1*(t^3)*ct/(12*nu12*(r^3)*R*(zeta^4)))*(k^3))
*(-(r^2)*zeta)/t)/((r^2)*zeta);
ann(j,j)=((f1*t*(ct^2)*nu21/((R^2)*(zeta^2)*nu12)
+2*nu21*f1*t*ct/(r*R*zeta)+f1*t/(r^2))*(-(r^2)*zeta)/t
+(-(G12*(t^3)*st/(6*(r^2)*(R^2)*(zeta^4))
+G12*(t^2)*(-t2*st)/(2*(r^3)*(zeta^3)*R)
+nu21*f1*t*((t2*st)^2)/(2*(r^4)*(zeta^2))
+nu21*f1*(t^2)*st*(-t2*st)/(2*(r^3)*(zeta^3)*R)
```

```
+nu21*f1*(t^2)*(-t2*ct)/(4*(r^4)*(zeta^2))
+nu21*f1*(t^3)*(st^2)/(6*(r^2)*(R^2)*(zeta^4))
+nu21*f1*(t^3)*ct/(12*(r^3)*(zeta^3)*R)
+f1*nu21*(t^3)*(st^2)/(6*nu12*(r^2)*(R^2)*(zeta^4))
+f1*nu21*(t^2)*(-t2*st)*st/(4*nu12*(r^3)*R*(zeta^3))
+nu21*f1*(t^3)*ct/(12*nu12*(r^3)*R*(zeta^3)))*(k^2)
+(nu21*f1*(t^3)/(12*nu12*(r^4)*(zeta^4)))*(k^4))*(-(r^2)*zeta)/t)/((r^2)*zeta);
%
% terms which involve the phi derivatives
for jj=1:n
aii(j,jj)=aii(j,jj)+(((f1*(-t2*st)*zeta/r-f1*t*gammar*st/r
+f1*(t^2)*(-t2*st)*zeta/(4*(r^3))-f1*(t^3)*gammar*st/(12*(r^3)))*b1(j,jj)
+(f1*t*zeta/r+f1*(t^3)*zeta/(12*(r^3)))*b2(j,jj))*r/t)/((r^2)*zeta);
aim(j,jj)=aim(j,jj)+(((nu21*f1*(t^3)*ct/(12*zeta*(r^2)*R)+G12*t/r
+G12*(t^3)/(24*(r^3))-G12*(t^3)/(24*(r^3)*zeta)-G12*(t^3)/(32*(r^3)*(zeta^2))
+nu21*f1*t/r)*k*b1(j,jj))*r/t)/((r^2)*zeta);
ain(j,jj)=ain(j,jj)+(((f1*nu21*t*gammar*ct/r
+f1*t*zeta/r+nu21*f1*(t^2)*(-t2*st)*st/(4*(r^2)*R)
+nu21*f1*(t^3)*ct/(12*(r^2)*R)
+f1*(t^3)*nu21*gammar*(st^2)/(12*nu12*(r^2)*R*zeta))*b1(j,jj)
+(-f1*(t^2)*(-t2*st)*zeta/(4*(r^3))+f1*(t^3)*gammar*st/(12*(r^3)))*b2(j,jj)
+(-f1*(t^3)*zeta/(12*(r^3)))*b3(j,jj))*r/t-((-nu21*f1*(t^3)/(12*(r^3)*zeta)
-G12*(t^3)/(12*(r^3)*zeta)
-G12*(t^3)/(24*(r^3)*(zeta^2)))*(k^2)*b1(j,jj))*r/t)/((r^2)*zeta);
ami(j,jj)=ami(j,jj)-(((nu21*f1*(t^3)*ct/(12*(r^2)*R*zeta)
+nu21*f1*t/r+G12*(t^3)/(24*(r^3))-G12*(t^3)/(24*(r^3)*zeta)
-G12*(t^3)/(32*(r^3)*(zeta^2))+G12*t/r)*k*b1(j,jj))*r/t)/((r^2)*zeta);
amm(j,jj)=amm(j,jj)+(((-st*G12*t/R+G12*(t^2)*zeta*(-t2*st)/(8*(r^3))
-3*G12*(t^2)*(-t2*st)/(8*(r^3))
+9*G12*(t^2)*(-t2*st)/(32*(r^3)*zeta)+3*G12*(t^3)*st/(32*(r^2)*R*(zeta^2))
-st*G12*(t^3)/(24*(r^2)*R)+G12*(-t2*st)*zeta/r)*b1(j,jj)+(G12*t*zeta/r
+G12*(t^3)*zeta/(24*(r^3))-
G12*(t^3)/(8*(r^3))+3*G12*(t^3)/(32*(r^3)*zeta))*b2(j,jj))*r/t)/((r^2)*zeta);
amn(j,jj)=amn(j,jj)-(((-st*f1*nu21*(t^3)*ct/(12*nu12*(R^2)*r*(zeta^2))
-G12*(t^2)*(-t2*st)/(4*(r^3))+3*G12*(t^2)*(-t2*st)/(8*(r^3)*zeta)
-G12*(t^3)*st/(12*(r^2)*R*zeta)-G12*(t^3)/(12*(r^2)*R*zeta)
+G12*(t^3)/(8*(r^2)*R*(zeta^2))+G12*(t^3)*st2/(12*r*(R^2)*(zeta^2))
+G12*(t^3)*st/(6*(r^2)*(zeta^2)*R))*k*b1(j,jj))*r/t
-((-nu21*f1*(t^3)*ct/(12*(r^2)*R*zeta)
-G12*(t^3)*ct/(12*(r^2)*R*zeta)
+G12*(t^3)/(24*(r^3)*zeta))*k*b2(j,jj))*r/t)/((r^2)*zeta);
ani(j,jj)=ani(j,jj)+(((f1*t/(r^2)+ct*nu21*f1*t/(R*r*zeta)
+nu21*f1*(t^3)*ct/(12*(r^3)*R*zeta)+nu21*f1*(t^2)*st*(-t2*st)/(4*(r^3)*R*zeta)
-f1*(t^2)*(-t2*ct)/(4*(r^4))-f1*t*((t2*st)^2)/(2*(r^4))
+f1*(t^3)*(st^2)*nu21/(12*(r^2)*(R^2)*(zeta^2)*nu12)
+f1*(t^2)*(-t2*st)*st/(2*(r^3)*R*zeta)+f1*(t^3)*ct/(12*(r^3)*R*zeta))*b1(j,jj)
+(-f1*(t^2)*(-t2*st)/(2*(r^4))+f1*(t^3)*st/(6*(r^3)*R*zeta))*b2(j,jj)
+(-f1*(t^3)/(12*(r^4)))*b3(j,jj))*(-(r^2)*zeta)/t
+((-G12*(t^3)/(12*(zeta^2)*(r^4))-G12*(t^3)/(24*(zeta^3)*(r^4))
-nu21*f1*(t^3)/(12*(r^4)*(zeta^2)))*(k^2)*b1(j,jj))*((r^2)*zeta)/t)
/((r^2)*zeta);
anm(j,jj)=anm(j,jj)+(((G12*(t^3)*st/(6*(r^3)*R*(zeta^2))
-G12*(t^3)*st/(24*(r^3)*R*(zeta^3))-G12*(t^2)*(-t2*st)/(4*(r^4)*zeta)
+3*G12*(t^2)*(-t2*st)/(8*(r^4)*(zeta^2))
-G12*(t^3)*st2/(12*(r^2)*(R^2)*(zeta^3))
-nu21*f1*(t^2)*ct*(-t2*st)/(2*(r^3)*R*(zeta^2))
+nu21*f1*(t^3)*st/(6*(r^3)*R*(zeta^2))
```

```
 -nu21*f1*(t^3)*st2/(24*nu12*(r^2)*(R^2)*(zeta^3))
 -nu21*f1*(t^3)*st2/(12*(r^2)*(R^2)*(zeta^3)))*k*b1(j,jj)
 +(-G12*(t^3)/(12*(r^4)*zeta)+G12*(t^3)/(8*(r^4)*(zeta^2))
 -nu21*f1*(t^3)*ct/(12*(r^3)*R*(zeta^2)))*k*b2(j,jj))
 *(-(r^2)*zeta)/t)/((r^2)*zeta);
 ann(j,jj)=ann(j,jj)+(((nu21*f1*(t^3)*st/(12*(r^3)*R*zeta)
 -nu21*f1*t*st*((t2*st)^2)/(2*(r^3)*R*zeta)
 -nu21*f1*(t^2)*ct*(-t2*st)/(2*(r^3)*R*zeta)
 -nu21*f1*(t^2)*st*(-t2*ct)/(4*(r^3)*R*zeta)
 -nu21*f1*(t^3)*(st^3)/(12*nu12*r*(R^3)*(zeta^3))
 -f1*(t^2)*(st^2)*(-t2*st)*nu21/(4*(r^2)*(R^2)*(zeta^2)*nu12)
 -f1*(t^3)*st*ct*nu21/(6*(r^2)*(R^2)*(zeta^2)*nu12))*b1(j,jj)
 +(-nu21*f1*(t^3)*ct/(12*(r^3)*R*zeta)-nu21*f1*(t^2)*st*(-
 t2*st)/(4*(r^3)*R*zeta)+(t^2)*(-t2*ct)*f1/(4*(r^4))
 +f1*((t2*st)^2)*t/(2*(r^4))-f1*(t^3)*(st^2)*nu21/(nu12*12*(r^2)*(R^2)*(zeta^2))
 -f1*(t^2)*(-t2*st)*st/(2*(r^3)*R*zeta)-f1*(t^3)*ct/(12*(r^3)*R*zeta))*b2(j,jj)
 +(f1*(t^2)*(-t2*st)/(2*(r^4))-f1*(t^3)*st/(6*(r^3)*R*zeta))*b3(j,jj)
 +(f1*(t^3)/(12*(r^4)))*b4(j,jj))*(-(r^2)*zeta)/t+((nu21*f1*(t^2)*(-
 t2*st)/(2*(r^4)*(zeta^2))+nu21*f1*(t^3)*st/(6*(r^3)*(zeta^3)*R)
 +G12*(t^3)*(-t2*st)/(2*(r^4)*(zeta^2))
 +G12*(t^3)/(6*(r^3)*R*(zeta^3)))*(k^2)*b1(j,jj)
 +(nu21*f1*(t^3)/(6*(r^4)*(zeta^2))+G12*(t^3)/(6*(r^4)*(zeta^2)))*(k^2)*b2(j,jj)
 )*((r^2)*zeta)/t)/((r^2)*zeta);
 end
 end
%-------------------------------------------------------------------------
% section eign
ar=1:n;
stiff=-[aii(ar,ar) aim(ar,ar) ain(ar,ar);
        ami(ar,ar) amm(ar,ar) amn(ar,ar);
        ani(ar,ar) anm(ar,ar) ann(ar,ar)]; clear aii aim ain ami amm amn ani
anm ann;
% find eignvalues
[evc, evl]=eig(stiff);  clear stiff;
tmn=3*n;
kk=0;
for i=1:tmn
if imag(evl(i,i)) == 0
if evl(i,i) > 0; kk=kk+1; ezz(kk)=evl(i,i); gzz(:,kk)=evc(:,i); end
end
end
ezzz=ezz(1:kk);
% sort the eigenvalues and output the ten lowest
sval=sort(ezzz);
for j=1:10; result(j)=sqrt(sval(j)/den); end
% ------------------------------------------------------------------------
% section mode - plot vibration mode shape
% F is a scaling factor
del(:,1)=gzz(:,49); clear evc;
F=0.7;
%
% displacement components at the sampling points, u, v, w
%
% copy in del values
uy=zeros(m+1,n+1); vy=zeros(m+1,n+1); wy=zeros(m+1,n+1);
for i=1:m; teta(i)=2*pi*(i-1)/m; end
teta(m+1)=.0;
```

```
for ar=1:n
for j=1:m+1
uy(j,ar)=uy(j,ar)+del(ar,1)*cos(1*teta(j));
vy(j,ar)=vy(j,ar)+del(ar+n,1)*sin(1*teta(j));
wy(j,ar)=wy(j,ar)+del(2*n+ar,1)*cos(1*teta(j));
end; end
% repeat the 360 deg values with the 0 deg values
ucm(:,1)=uy(:,1);
vcm(:,1)=vy(:,1);
wcm(:,1)=wy(:,1);
uyy=[uy ucm];
vyy=[vy vcm];
wyy=[wy wcm];
%
% determine the coordinates of undisplaced points
for i=1:n; tphi(i)=2*pi*(i-1)/n; end
tphi(n+1)=.0;
for j=1:m+1
thetang=teta(j); sp=sin(thetang); cp=cos(thetang);
for i=1:n+1
phiang=tphi(i); cs=cos(phiang); sn=sin(phiang);
xx(j,i)=(R+r*cs)*sp;
yy(j,i)=(R+r*cs)*cp;
zz(j,i)=r*sn;
end; end
%
% find coord of displaced points
for j=1:m+1
thetang=teta(j); sp=sin(thetang); cp=cos(thetang);
for i=1:n+1
phiang=tphi(i); cs=cos(phiang); sn=sin(phiang);
csn=cos(phiang+pi/2); snn=sin(phiang+pi/2);
xdisp(j,i)=xx(j,i)-F*(sp*(-uyy(j,i)*csn+wyy(j,i)*cs)+vyy(j,i)*cp);
ydisp(j,i)=yy(j,i)-F*(cp*(-uyy(j,i)*csn+wyy(j,i)*cs)-vyy(j,i)*sp);
zdisp(j,i)=zz(j,i)-F*(uyy(j,i)*snn+wyy(j,i)*sn);
end; end
% plot the mode shape
hold on
surf(xdisp,ydisp,zdisp);axis('on');
view(-60,60)
hold off
```

APPENDIX D - Listing of program Afreqisol.m

```
% AFreqisol.m  last modification - Feb. 25, 2002
% Sanders shell theory using 2 gov eqns comp & struct 43 (1992) 1019-1028.
% axisymmetric natural frequencies and mode shapes of steel torus using DQM
% harmonics trial function used in meridional direction
%--------------------------------------------------------------------------------
% section input
% n=sampling points in meridional direction;
% m=sampling points in circumferential direction;
% E1=Young's modulus (Pa); G12=shear moduli (Pa);
% nu12=nu21=Poisson's ratio; den=mass density (kg/m3);
% r=x-section radius (m); R=bend radius (m);
% t1 t2=thickness coefficients;
% format long
n=96
m=180;
r=0.25; R=0.762;  E1=2.07e11; nu12=0.3; nu21=0.3; f1=E1/(1-nu21*nu12);
gammar=r/R; t1=0.009525; t2=0.001; G12=E1/(2*(1+nu12)); den=7770;
%--------------------------------------------------------------------------------
% section wcmd - find the weighting coeffts for meridional direction
% unequal spacing with one delta point at each end for phi direction
% output - b1(i,j),b2(i,j),b3(i,j),b4(i,j) for 1st to 4th derivats
n2=n/2;
for i=1:n
tet(i)=2*pi*(i-1)/n; ib(i,1)=.5; ib(i,n2+1)=.5;
end
%
for i=2:2:n; ib(i,n2+1)=-0.5; end
%
for i=2:n2; ib(1,i)=1; end
%
for i=n2+2:n; ib(1,i)=0; end
%
for i=2:n; for j=n2+2:n
ib(i,j)=sin(2*pi*(i-1)*(j-1-n2)/n);
end; end
%
for i=2:n; for j=2:n2
ib(i,j)=cos(2*pi*(i-1)*(j-1)/n);
end; end
%
ib=ib/n2;
for i=1:n; for j=1:n
if abs(ib(i,j))<1e-12; ib(i,j)=0; end
end;end
%
for i=1:n; for j=1:n2+1
rr1(j)=(1-j)*sin((j-1)*tet(i));
rr2(j)=-1*((j-1)^2)*cos((j-1)*tet(i));
rr3(j)=((j-1)^3)*sin((j-1)*tet(i));
rr4(j)=((j-1)^4)*cos((j-1)*tet(i));
end
for j=n2+2:n
rr1(j)=(j-1-n2)*cos((j-1-n2)*tet(i));
rr2(j)=-1*((j-1-n2)^2)*sin((j-1-n2)*tet(i));
rr3(j)=-1*((j-1-n2)^3)*cos((j-1-n2)*tet(i));
```

```
rr4(j)=((j-1-n2)^4)*sin((j-1-n2)*tet(i));
end
%
b1(i,:)=(ib*rr1')';
b2(i,:)=(ib*rr2')';
b3(i,:)=(ib*rr3')';
b4(i,:)=(ib*rr4')';
end
%---------------------------------------------------------------------------------
% section domeqn - set up matrices enforcing the governing equations
%    aii    ain     u       0
%
%    ani    ann     w       0
% output - aii(i,j),...,ann(i,j) - coefft matrices for displacements
%
aii(:,:)=zeros(n); ain(:,:)=zeros(n);
ani(:,:)=zeros(n); ann(:,:)=zeros(n);
%
%loop over sampling points in the phi direction
for j=1:n
tee=tet(j); ct=cos(tee); st=sin(tee); st2=2*st*ct;
ct2=(ct^2)-(st^2); zeta=1+gammar*ct; t=t1+t2*ct;
% terms which do not involve derivatives
aii(j,j)=(-nu21*f1*(-t2*st)*gammar*st/r-nu21*f1*t*gammar*ct/r
-(gammar^2)*(st^2)*f1*t*nu21/(r*zeta*nu12)
-nu21*f1*(t^2)*(-t2*st)*st/(4*(r^2)*R)-f1*nu21*(t^3)*ct/(12*(r^2)*R)
-f1*(t^3)*gammar*nu21*(st^2)/(12*nu12*(r^2)*R*zeta))*r/((r^2)*zeta*t);
ain(j,j)=(nu21*f1*(-t2*st)*gammar*ct/r+f1*(-t2*st)*zeta/r
-f1*t*gammar*st/r+f1*t*(gammar^2)*st*ct*nu21/(r*zeta*nu12))*r
/((r^2)*zeta*t);
ani(j,j)=(-f1*t*nu21*gammar*st/((r^2)*zeta)
-f1*t*(gammar^2)*st*ct*nu21/((r^2)*(zeta^2)*nu12)
+f1*nu21*t*((t2*st)^2)*st/(2*(r^3)*R*zeta)
+nu21*f1*(t^2)*(-t2*ct)*st/(4*(r^3)*R*zeta)
+nu21*f1*(-t2*st)*(t^2)*ct/((r^3)*R*zeta*2)
-f1*(t^3)*nu21*st/(12*(r^3)*R*zeta)
+f1*(t^2)*(-t2*st)*gammar*(st^2)*nu21/(4*(r^3)*(zeta^2)*R*nu12)
+f1*(t^3)*gammar*st*ct*nu21/(6*(r^3)*(zeta^2)*R*nu12)
+f1*(t^3)*nu21*(gammar^2)*(st^3)/(12*nu12*(r^3)*R*(zeta^3)))
*(-(r^2)*zeta)/((r^2)*zeta*t);
ann(j,j)=(f1*t*(ct^2)*nu21/((R^2)*(zeta^2)*nu12)
+2*nu21*f1*t*ct/(r*R*zeta)+f1*t/(r^2))*(-(r^2)*zeta)/((r^2)*zeta*t);
%
% terms which involve the phi derivatives
for jj=1:n
aii(j,jj)=aii(j,jj)+((f1*(-t2*st)*zeta/r-f1*t*gammar*st/r
+f1*(t^2)*(-t2*st)*zeta/(4*(r^3))-f1*(t^3)*gammar*st/(12*(r^3)))*b1(j,jj)
+(f1*t*zeta/r+f1*(t^3)*zeta/(12*(r^3)))*b2(j,jj))*r/((r^2)*zeta*t);
ain(j,jj)=ain(j,jj)+((f1*nu21*t*gammar*ct/r+f1*t*zeta/r
+nu21*f1*(t^2)*(-t2*st)*st/(4*(r^2)*R)+nu21*f1*(t^3)*ct/(12*(r^2)*R)
+f1*(t^3)*nu21*gammar*(st^2)/(12*nu12*(r^2)*R*zeta))*b1(j,jj)
+(-f1*(t^2)*(-t2*st)*zeta/(4*(r^3))
+f1*(t^3)*gammar*st/(12*(r^3)))*b2(j,jj)
+(-f1*(t^3)*zeta/(12*(r^3)))*b3(j,jj))*r/((r^2)*zeta*t);
ani(j,jj)=ani(j,jj)+((f1*t/(r^2)+ct*nu21*f1*t/(R*r*zeta)
+nu21*f1*(t^3)*ct/(12*(r^3)*R*zeta)
+nu21*f1*(t^2)*st*(-t2*st)/(4*(r^3)*R*zeta)
```

```
-f1*(t^2)*(-t2*ct)/(4*(r^4))-f1*t*((t2*st)^2)/(2*(r^4))
+f1*(t^3)*(st^2)*nu21/(12*(r^2)*(R^2)*(zeta^2)*nu12)
+f1*(t^2)*(-t2*st)*st/(2*(r^3)*R*zeta)
+f1*(t^3)*ct/(12*(r^3)*R*zeta))*b1(j,jj)
+(-f1*(t^2)*(-t2*st)/(2*(r^4))+f1*(t^3)*st/(6*(r^3)*R*zeta))*b2(j,jj)
+(-f1*(t^3)/(12*(r^4)))*b3(j,jj))*(-(r^2)*zeta)/((r^2)*zeta*t);
ann(j,jj)=ann(j,jj)+((nu21*f1*(t^3)*st/(12*(r^3)*R*zeta)
-nu21*f1*t*st*((t2*st)^2)/(2*(r^3)*R*zeta)
-nu21*f1*(t^2)*ct*(-t2*st)/(2*(r^3)*R*zeta)
-nu21*f1*(t^2)*st*(-t2*ct)/(4*(r^3)*R*zeta)-
nu21*f1*(t^3)*(st^3)/(12*nu12*r*(R^3)*(zeta^3))
-f1*(t^2)*(st^2)*(-t2*st)*nu21/(4*(r^2)*(R^2)*(zeta^2)*nu12)
-f1*(t^3)*st*ct*nu21/(6*(r^2)*(R^2)*(zeta^2)*nu12))*b1(j,jj)
+(-nu21*f1*(t^3)*ct/(12*(r^3)*R*zeta)
-nu21*f1*(t^2)*st*(-t2*st)/(4*(r^3)*R*zeta)
+(t^2)*(-t2*ct)*f1/(4*(r^4))+f1*((t2*st)^2)*t/(2*(r^4))
-f1*(t^3)*(st^2)*nu21/(nu12*12*(r^2)*(R^2)*(zeta^2))
-f1*(t^2)*(-t2*st)*st/(2*(r^3)*R*zeta)
-f1*(t^3)*ct/(12*(r^3)*R*zeta))*b2(j,jj)+(f1*(t^2)*(-t2*st)/(2*(r^4))
-f1*(t^3)*st/(6*(r^3)*R*zeta))*b3(j,jj)+(f1*(t^3)/(12*(r^4)))*b4(j,jj))
*(-(r^2)*zeta)/((r^2)*zeta*t);
end
end
clear j; clear jj;
%---------------------------------------------------------------------------
% section eign
ar=1:n;
stiff=-[aii(ar,ar)  ain(ar,ar);
        ani(ar,ar)  ann(ar,ar)]; clear aii  ain  ani  ann;
% find eignvalues
[evc, evl]=eig(stiff);  clear stiff;
tmn=2*n;
kk=0;
for i=1:tmn
if imag(evl(i,i)) == 0
if evl(i,i) > 0; kk=kk+1; ezz(kk)=evl(i,i); gzz(:,kk)=evc(:,i); end
end
end
ezzz=ezz(1:kk);
% sort the eigenvalues and output the five lowest
sval=sort(ezzz);
for j=1:10; result(j)=sqrt(sval(j)/den); end
% -------------------------------------------------------------------------
theta=linspace(0,2*pi,n+1);
for i=1:n
r2(i)=gzz(i+n,132);
r(i)=1+2*r2(i);
end
r(n+1)=1+2*r2(1);
polar(theta,r);
```

APPENDIX D - Listing of program Freqiso2.m

```
% Freqiso2.m last modification - Feb 25, 2002
% Sanders shell theory using 3 gov eqns comp & struct 43 (1992) 1019-1028.
% natural frequencies of steel torus shell using dqm
% polynomial trial function used in meridional direction
% Fourier series expansion written in circumferential direction
%------------------------------------------------------------------------------
% section input
% n=sampling points in meridional direction;
% m=sampling points in circumferential direction;
% E1=Young's modulus (Pa); G12=shear moduli (Pa);
% nu12=nu21=Poisson's ratio; den=mass density (Pa);
% r=x-section radius (m); R=bend radius (m);
% t1 t2=thickness coefficients
% format long
n=44
m=120
k=1/(0.762/0.25)
r=0.25; R=0.762;  E1=2.07e11; nu12=0.3; nu21=0.3; f1=E1/(1-nu21*nu12);
gammar=r/R; t1=0.009525; t2=0.001; G12=E1/(2*(1+nu12))
den=7770; xlo=0; xhi=2*pi; xdif=xhi-xlo;
%------------------------------------------------------------------------------
% section wcmd - find the weighting coeffts for meridional direction
% unequal spacing with one delta point at each end for phi direction
% output - b1(i,j),b2(i,j),b3(i,j),b4(i,j) for 1st to 4th derivats
y(1)=0; y(2)=.00001; y(n-1)=.99999;y(n)=1;
%
for i=3:n-2
tt=(pi*(i-2))/(n-3);y(i)=(1-cos(tt))/2;
end
%
for i=1:n; x(i)=xlo+y(i)*xdif; end
%
for i=1:n; pai(i)=1; for j=1:n
if i~=j; pai(i)=(x(i)-x(j))*pai(i); end
end;end
%
for i=1:n; for j=1:n
if i~=j; b1(i,j)=pai(i)/((x(i)-x(j))*pai(j)); end
end;end
%
for i=1:n; b1(i,i)=0; for j=1:n
if i~=j; b1(i,i)=b1(i,i)-b1(i,j); end
end;end
%
for i=1:n; for j=1:n
if i~=j; b2(i,j)=2*(b1(i,i)*b1(i,j)-b1(i,j)/(x(i)-x(j))); end
end;end
%
for i=1:n; b2(i,i)=0; for j=1:n
if i~=j; b2(i,i)=b2(i,i)-b2(i,j); end
end;end
%
for i=1:n; for j=1:n
if i~=j; b3(i,j)=3*(b2(i,i)*b1(i,j)-b2(i,j)/(x(i)-x(j))); end
end;end
```

```
%
for i=1:n; b3(i,i)=0; for j=1:n
if i~=j; b3(i,i)=b3(i,i)-b3(i,j); end
end;end
%
for i=1:n; for j=1:n
if i~=j; b4(i,j)=4*(b3(i,i)*b1(i,j)-b3(i,j)/(x(i)-x(j))); end
end;end
%
for i=1:n; b4(i,i)=0; for j=1:n
if i~=j; b4(i,i)=b4(i,i)-b4(i,j); end
end;end
%-----------------------------------------------------------------------------
% section domeqn - set up matrices enforcing the governing equations
%    aii  aim  ain     u      0
%    ami  amm  amn     v  =   0
%    ani  anm  ann     w      0
% output - aii(i,j),...,ann(i,j) - coefft matrices for displacements
%
aii(:,:)=zeros(n);aim(:,:)=zeros(n);ain(:,:)=zeros(n);
ami(:,:)=zeros(n);amm(:,:)=zeros(n);amn(:,:)=zeros(n);
ani(:,:)=zeros(n);anm(:,:)=zeros(n);ann(:,:)=zeros(n);
%
%loop over sampling points in the phi direction
for j=2:n-1
tee=x(j); ct=cos(tee); st=sin(tee); st2=2*st*ct;
ct2=(ct^2)-(st^2); zeta=1+gammar*ct; t=t1-t2*ct;
% terms which do not involve derivatives
aii(j,j)=((-nu21*f1*(-t2*st)*gammar*st/r-nu21*f1*t*gammar*ct/r
-(gammar^2)*(st^2)*f1*t*nu21/(r*zeta*nu12)
-nu21*f1*(t^2)*(-t2*st)*st/(4*(r^2)*R)-f1*nu21*(t^3)*ct/(12*(r^2)*R)
-f1*(t^3)*gammar*nu21*(st^2)/(12*nu12*(r^2)*R*zeta))*r/t
-((G12*t/(r*zeta)+G12*(t^3)/(24*(r^3)*(zeta^2))
+G12*(t^3)/(24*(r^3)*zeta)
+G12*(t^3)/(96*(r^3)*(zeta^3)))*(k^2))*r/t)/((r^2)*zeta);
aim(j,j)=(((f1*nu21*(-t2*st)/r+st*nu21*f1*t/(R*zeta*nu12)
+nu21*f1*(t^2)*ct*(-t2*st)/(4*(r^2)*R*zeta)
-nu21*f1*(t^3)*st/(12*(r^2)*R*zeta)
+G12*(t^3)*st2/(24*(R^2)*r*(zeta^2))+G12*t*st/(R*zeta)
+G12*(t^3)*st2/(48*(R^2)*r*(zeta^3))-G12*(t^3)*st/(24*R*(r^2)*zeta)
+G12*(t^3)*st/(96*R*(r^2)*(zeta^3))
+st2*nu21*f1*(t^3)/(24*(R^2)*r*(zeta^2))
+nu21*f1*(t^3)*st2/(24*(R^2)*r*nu12*(zeta^2)))*k)*r/t)/((r^2)*zeta);
ain(j,j)=((nu21*f1*(-t2*st)*gammar*ct/r+f1*(-t2*st)*zeta/r
-f1*t*gammar*st/r+f1*t*(gammar^2)*st*ct*nu21/(r*zeta*nu12))*r/t
-((-nu21*f1*(t^2)*(-t2*st)/(4*(r^3)*zeta)
-nu21*f1*(t^3)*st/(12*(r^2)*(zeta^2)*R)
-G12*(t^3)/(12*(r^2)*R*(zeta^2))-G12*(t^3)/(24*(r^2)*R*(zeta^3))
-st*f1*nu21*(t^3)/(12*nu12*R*(r^2)*(zeta^2)))*(k^2))*r/t)/((r^2)*zeta);
ami(j,j)=(((-st*f1*nu21*t/(R*nu12*zeta)-G12*t*st/(R*zeta)
-f1*nu21*(st2)*(t^3)/(24*(R^2)*r*(zeta^2)*nu12)+G12*(-t2*st)/r
+G12*(t^2)*(-t2*st)/(8*(r^3))-G12*(t^2)*(-t2*st)/(8*(r^3)*zeta)
-3*G12*(t^2)*(-t2*st)/(32*(r^3)*(zeta^2))+G12*(t^3)*st/(24*(r^2)*R*zeta)
-7*G12*(t^3)*st/(96*(r^2)*R*(zeta^3))-st2*G12*(t^3)/(48*r*(R^2)*(zeta^3))
-st2*G12*(t^3)/(24*(r^2)*(R^2)*(zeta^2))
-st*G12*(t^3)/(24*(r^2)*R*(zeta^2)))*(-k))*r/t)/((r^2)*zeta);
amm(j,j)=((-2*G12*t*r*(st^2)/((R^2)*zeta)+G12*ct*t/R
```

```
+G12*t*r*(st^2)/((R^2)*zeta)
+G12*(t^2)*st2*(-t2*st)/(8*(R^2)*r*zeta)
-3*G12*(t^2)*st2*(-t2*st)/(16*(R^2)*r*(zeta^2))
-G12*(t^2)*st*(-t2*st)/(8*R*(r^2))+G12*(t^2)*st*(-t2*st)/(4*R*(r^2)*zeta)
-3*G12*(t^2)*st*(-t2*st)/(32*R*(r^2)*(zeta^2))+G12*(t^3)*ct2/(12*(R^2)*r*zeta)
-G12*(t^3)*ct2/(8*(R^2)*r*(zeta^2))-G12*(t^3)*ct/(24*(r^2)*R)
+G12*(t^3)*ct/(12*(r^2)*R*zeta)-G12*(t^3)*ct/(32*(r^2)*R*(zeta^2))
-G12*(t^3)*(st^2)/(24*(R^2)*r*zeta)+G12*st*(-t2*st)/R
-r*(st2^2)*G12*(t^3)/(24*(R^4)*(zeta^3))+G12*(t^3)*(st^2)/(8*(R^2)*r*(zeta^2))
-7*G12*(t^3)*(st^2)/(96*(R^2)*r*(zeta^3))+G12*(t^3)*st*st2/(8*(R^3)*(zeta^2))
-G12*(t^3)*st*st2/(6*(R^3)*(zeta^3)))*r/t
-((ct^2)*nu21*f1*(t^3)/(12*(R^2)*r*(zeta^3)*nu12)
+f1*nu21*t/(r*zeta*nu12))*(k^2)*r/t)/((r^2)*zeta);
amn(j,j)=(((nu21*f1*t/r-G12*(t^2)*(-t2*st)/(4*(r^2)*R*zeta)
+3*G12*(t^2)*(-t2*st)/(8*(r^2)*R*(zeta^2))
-G12*(t^3)*st/(6*(R^2)*r*(zeta^2))+f1*nu21*t*ct/(R*zeta*nu12)
+G12*(t^3)*st2/(12*(R^3)*(zeta^3))+7*G12*(t^3)*st/(24*(R^2)*r*(zeta^3)))*(-k)
+(-f1*nu21*(t^3)*ct/(12*R*(r^2)*nu12*(zeta^3)))*(k^3))*r/t)/((r^2)*zeta);
ani(j,j)=((-f1*t*nu21*gammar*st/((r^2)*zeta)
-f1*t*(gammar^2)*st*ct*nu21/((r^2)*(zeta^2)*nu12)
+f1*nu21*t*((t2*st)^2)*st/(2*(r^3)*R*zeta)
+nu21*f1*(t^2)*(-t2*ct)*st/(4*(r^3)*R*zeta)
+nu21*f1*(-t2*st)*(t^2)*ct/((r^3)*R*zeta*2)
-f1*(t^3)*nu21*st/(12*(r^3)*R*zeta)
+f1*(t^2)*(-t2*st)*gammar*(st^2)*nu21/(4*(r^3)*(zeta^2)*R*nu12)
+f1*(t^3)*gammar*st*ct*nu21/(6*(r^3)*(zeta^2)*R*nu12)
+f1*(t^3)*nu21*(gammar^2)*(st^3)/(12*nu12*(r^3)*R*(zeta^3)))*(-(r^2)*zeta)/t
-((-G12*(t^2)*(-t2*st)/(4*(r^4)*(zeta^2))
-G12*(t^2)*(-t2*st)/(8*(r^4)*(zeta^3))
-G12*(t^3)*st/(24*(r^3)*(zeta^4)*R)
+f1*nu21*(t^3)*st/(12*nu12*(r^2)*R*(zeta^3)))*(k^2))*(-
(r^2)*zeta)/t)/((r^2)*zeta);
anm(j,j)=(((nu21*f1*t/((r^2)*zeta)
-G12*(t^2)*st2*(-t2*st)/(4*(r^2)*(R^2)*(zeta^3))
+G12*(t^2)*st*(-t2*st)/(4*(r^3)*R*(zeta^2))
-G12*(t^2)*st*(-t2*st)/(8*(r^3)*R*(zeta^3))
-G12*(t^3)*ct2/(6*(r^2)*(R^2)*(zeta^3))
+G12*(t^3)*ct/(12*(r^3)*R*(zeta^2))
-G12*(t^3)*ct/(24*(r^3)*R*(zeta^3))
-G12*(t^3)*st*st2/(12*r*(R^3)*(zeta^4))
-G12*(t^3)*(st^2)/(24*(r^2)*(R^2)*(zeta^4))
-nu21*f1*t*ct*((t2*st)^2)/(2*(r^3)*R*(zeta^2))
+nu21*f1*(t^2)*st*(-t2*st)/(2*(r^3)*R*(zeta^2))
-nu21*f1*(t^2)*st2*(-t2*st)/(4*(r^2)*(R^2)*(zeta^3))
-nu21*f1*(t^2)*ct*(-t2*ct)/(4*(r^3)*R*(zeta^2))
+nu21*f1*(t^3)*ct/(12*(r^3)*R*(zeta^2))
-nu21*f1*(t^3)*ct*(st^2)/(6*r*(R^3)*(zeta^4))
-(ct^2)*nu21*f1*(t^3)/(12*(r^2)*(R^2)*(zeta^3))
-nu21*f1*(t^3)*(st^2)*ct/(6*r*(R^3)*(zeta^4)*nu12)
-nu21*f1*(t^2)*st2*(-t2*st)/(8*nu12*(r^2)*(R^2)*(zeta^3))
+nu21*f1*(t^3)*(st^2)/(6*(r^2)*(R^2)*(zeta^3))
+nu21*f1*ct*t/(nu12*r*R*(zeta^2))
-nu21*f1*(t^3)*ct2/(12*nu12*(r^2)*(R^2)*(zeta^3)))*k
-(-nu21*f1*(t^3)*ct/(12*nu12*(r^3)*R*(zeta^4)))*(k^3))*(-
(r^2)*zeta)/t)/((r^2)*zeta);
ann(j,j)=((f1*t*(ct^2)*nu21/((R^2)*(zeta^2)*nu12)
+2*nu21*f1*t*ct/(r*R*zeta)+f1*t/(r^2))*(-(r^2)*zeta)/t
```

```
+(-(G12*(t^3)*st/(6*(r^2)*(R^2)*(zeta^4))
+G12*(t^2)*(-t2*st)/(2*(r^3)*(zeta^3)*R)
+nu21*f1*t*((t2*st)^2)/(2*(r^4)*(zeta^2))
+nu21*f1*(t^2)*st*(-t2*st)/(2*(r^3)*(zeta^3)*R)
+nu21*f1*(t^2)*(-
t2*ct)/(4*(r^4)*(zeta^2))+nu21*f1*(t^3)*(st^2)/(6*(r^2)*(R^2)*(zeta^4))
+nu21*f1*(t^3)*ct/(12*(r^3)*(zeta^3)*R)
+f1*nu21*(t^3)*(st^2)/(6*nu12*(r^2)*(R^2)*(zeta^4))
+f1*nu21*(t^2)*(-t2*st)*st/(4*nu12*(r^3)*R*(zeta^3))
+nu21*f1*(t^3)*ct/(12*nu12*(r^3)*R*(zeta^3)))*(k^2)
+(nu21*f1*(t^3)/(12*nu12*(r^4)*(zeta^4)))*(k^4))*(-(r^2)*zeta)/t)
/((r^2)*zeta);
%
% terms which involve the phi derivatives
for jj=1:n
aii(j,jj)=aii(j,jj)+(((f1*(-t2*st)*zeta/r-f1*t*gammar*st/r
+f1*(t^2)*(-t2*st)*zeta/(4*(r^3))-f1*(t^3)*gammar*st/(12*(r^3)))*b1(j,jj)
+(f1*t*zeta/r+f1*(t^3)*zeta/(12*(r^3)))*b2(j,jj))*r/t)/((r^2)*zeta);
aim(j,jj)=aim(j,jj)+(((nu21*f1*(t^3)*ct/(12*zeta*(r^2)*R)+G12*t/r
+G12*(t^3)/(24*(r^3))-G12*(t^3)/(24*(r^3)*zeta)-G12*(t^3)/(32*(r^3)*(zeta^2))
+nu21*f1*t/r)*k*b1(j,jj))*r/t)/((r^2)*zeta);
ain(j,jj)=ain(j,jj)+(((f1*nu21*t*gammar*ct/r+f1*t*zeta/r
+nu21*f1*(t^2)*(-t2*st)*st/(4*(r^2)*R)+nu21*f1*(t^3)*ct/(12*(r^2)*R)
+f1*(t^3)*nu21*gammar*(st^2)/(12*nu12*(r^2)*R*zeta))*b1(j,jj)
+(-f1*(t^2)*(-t2*st)*zeta/(4*(r^3))+f1*(t^3)*gammar*st/(12*(r^3)))*b2(j,jj)
+(-f1*(t^3)*zeta/(12*(r^3)))*b3(j,jj))*r/t-((-nu21*f1*(t^3)/(12*(r^3)*zeta)
-G12*(t^3)/(12*(r^3)*zeta)-G12*(t^3)/(24*(r^3)*(zeta^2)))*(k^2)*b1(j,jj))*r/t)
/((r^2)*zeta);
ami(j,jj)=ami(j,jj)-
(((nu21*f1*(t^3)*ct/(12*(r^2)*R*zeta)+nu21*f1*t/r+G12*(t^3)/(24*(r^3))
-G12*(t^3)/(24*(r^3)*zeta)-G12*(t^3)/(32*(r^3)*(zeta^2))
+G12*t/r)*k*b1(j,jj))*r/t)/((r^2)*zeta);
amm(j,jj)=amm(j,jj)+(((-st*G12*t/R+G12*(t^2)*zeta*(-t2*st)/(8*(r^3))
-3*G12*(t^2)*(-t2*st)/(8*(r^3))+9*G12*(t^2)*(-t2*st)/(32*(r^3)*zeta)
+3*G12*(t^3)*st/(32*(r^2)*R*(zeta^2))-st*G12*(t^3)/(24*(r^2)*R)
+G12*(-t2*st)*zeta/r)*b1(j,jj)+(G12*t*zeta/r+G12*(t^3)*zeta/(24*(r^3))
-G12*(t^3)/(8*(r^3))+3*G12*(t^3)/(32*(r^3)*zeta))*b2(j,jj))*r/t)/((r^2)*zeta);
amn(j,jj)=amn(j,jj)-(((st*f1*nu21*(t^3)*ct/(12*nu12*(R^2)*r*(zeta^2))
-G12*(t^2)*(-t2*st)/(4*(r^3))+3*G12*(t^2)*(-t2*st)/(8*(r^3)*zeta)
-G12*(t^3)*st/(12*(r^2)*R*zeta)-G12*(t^3)/(12*(r^2)*R*zeta)
+G12*(t^3)/(8*(r^2)*R*(zeta^2))+G12*(t^3)*st2/(12*r*(R^2)*(zeta^2))
+G12*(t^3)*st/(6*(r^2)*(zeta^2)*R))*k*b1(j,jj))*r/t
-((-nu21*f1*(t^3)*ct/(12*(r^2)*R*zeta)-G12*(t^3)*ct/(12*(r^2)*R*zeta)
+G12*(t^3)/(24*(r^3)*zeta))*k*b2(j,jj))*r/t)/((r^2)*zeta);
ani(j,jj)=ani(j,jj)+(((f1*t/(r^2)+ct*nu21*f1*t/(R*r*zeta)
+nu21*f1*(t^3)*ct/(12*(r^3)*R*zeta)+nu21*f1*(t^2)*st*(-t2*st)/(4*(r^3)*R*zeta)
-f1*(t^2)*(-t2*ct)/(4*(r^4))-f1*t*((t2*st)^2)/(2*(r^4))
+f1*(t^3)*(st^2)*nu21/(12*(r^2)*(R^2)*(zeta^2)*nu12)
+f1*(t^2)*(-t2*st)*st/(2*(r^3)*R*zeta)+f1*(t^3)*ct/(12*(r^3)*R*zeta))*b1(j,jj)
+(-f1*(t^2)*(-t2*st)/(2*(r^4))+f1*(t^3)*st/(6*(r^3)*R*zeta))*b2(j,jj)
+(-f1*(t^3)/(12*(r^4)))*b3(j,jj))*(-(r^2)*zeta)/t
+((-G12*(t^3)/(12*(zeta^2)*(r^4))-G12*(t^3)/(24*(zeta^3)*(r^4))
-nu21*f1*(t^3)/(12*(r^4)*(zeta^2)))*(k^2)*b1(j,jj))*((r^2)*zeta)/t)
/((r^2)*zeta);
anm(j,jj)=anm(j,jj)+(((G12*(t^3)*st/(6*(r^3)*R*(zeta^2))-
G12*(t^3)*st/(24*(r^3)*R*(zeta^3))
-G12*(t^2)*(-t2*st)/(4*(r^4)*zeta)+3*G12*(t^2)*(-t2*st)/(8*(r^4)*(zeta^2))
```

```
-G12*(t^3)*st2/(12*(r^2)*(R^2)*(zeta^3))
-nu21*f1*(t^2)*ct*(-t2*st)/(2*(r^3)*R*(zeta^2))
+nu21*f1*(t^3)*st/(6*(r^3)*R*(zeta^2))
-nu21*f1*(t^3)*st2/(24*nu12*(r^2)*(R^2)*(zeta^3))
-nu21*f1*(t^3)*st2/(12*(r^2)*(R^2)*(zeta^3)))*k*b1(j,jj)
+(-G12*(t^3)/(12*(r^4)*zeta)+G12*(t^3)/(8*(r^4)*(zeta^2))
-nu21*f1*(t^3)*ct/(12*(r^3)*R*(zeta^2)))*k*b2(j,jj))*(-
(r^2)*zeta)/t)/((r^2)*zeta);
ann(j,jj)=ann(j,jj)+(((nu21*f1*(t^3)*st/(12*(r^3)*R*zeta)
-nu21*f1*t*st*((t2*st)^2)/(2*(r^3)*R*zeta)
-nu21*f1*(t^2)*ct*(-t2*st)/(2*(r^3)*R*zeta)
-nu21*f1*(t^2)*st*(-t2*ct)/(4*(r^3)*R*zeta)
-nu21*f1*(t^3)*(st^3)/(12*nu12*r*(R^3)*(zeta^3))
-f1*(t^2)*(st^2)*(-t2*st)*nu21/(4*(r^2)*(R^2)*(zeta^2)*nu12)
-f1*(t^3)*st*ct*nu21/(6*(r^2)*(R^2)*(zeta^2)*nu12))*b1(j,jj)
+(-nu21*f1*(t^3)*ct/(12*(r^3)*R*zeta)-nu21*f1*(t^2)*st*(-t2*st)/(4*(r^3)*R*zeta)
+(t^2)*(-t2*ct)*f1/(4*(r^4))+f1*((t2*st)^2)*t/(2*(r^4))
-f1*(t^3)*(st^2)*nu21/(nu12*12*(r^2)*(R^2)*(zeta^2))
-f1*(t^2)*(-t2*st)*st/(2*(r^3)*R*zeta)-f1*(t^3)*ct/(12*(r^3)*R*zeta))*b2(j,jj)
+(f1*(t^2)*(-t2*st)/(2*(r^4))-f1*(t^3)*st/(6*(r^3)*R*zeta))*b3(j,jj)
+(f1*(t^3)/(12*(r^4)))*b4(j,jj))*(-(r^2)*zeta)/t
+((nu21*f1*(t^2)*(-t2*st)/(2*(r^4)*(zeta^2))
+nu21*f1*(t^3)*st/(6*(r^3)*(zeta^3)*R)
+G12*(t^3)*(-t2*st)/(2*(r^4)*(zeta^2))
+G12*(t^3)/(6*(r^3)*R*(zeta^3)))*(k^2)*b1(j,jj)
+(nu21*f1*(t^3)/(6*(r^4)*(zeta^2))
+G12*(t^3)/(6*(r^4)*(zeta^2)))*(k^2)*b2(j,jj))*((r^2)*zeta)/t)/((r^2)*zeta);
end
end
%-------------------------------------------------------------------------------
% section bound - enforcing the  four displacement relations
% terms which do not involve derivatives
aii(1,1)=-1;   aii(1,n)=1;
amm(1,1)=-1;   amm(1,n)=1;
ann(1,1)=-1;   ann(1,n)=1;
%
% terms which involve the phi derivatives
for jj=1:n
ani(2,jj)=0;
ani(n-1,jj)=0;
anm(2,jj)=0;
anm(n-1,jj)=0;
ann(2,jj)=0;
ann(n-1,jj)=0;
end
for jj=1:n
ann(2,jj)=ann(2,jj)+(-b1(2,jj)+b1(n-1,jj));
end
%-------------------------------------------------------------------------------
% section bound - enforce four force resultants relations
tee=x(n); ct=cos(tee); st=sin(tee); st2=2*st*ct; ct2=(ct^2)-(st^2);
zeta=1+gammar*ct; t=t1-t2*ct;
% terms which do not involve derivatives
aii(n,n)=-gammar*(st^2)*f1*(t^3)*nu21/(12*(r^2)*R*(zeta^2)*nu12)-
f1*nu21*st*(t^2)*(-t2*st)/(4*(r^2)*R*zeta)-
(k^2)*(G12*(t^3)/(24*(r^3)*(zeta^3))+G12*(t^3)/(12*(r^3)*(zeta^2)));
```

```
aim(n,n)=(f1*(t^2)*(-
t2*st)*nu21*ct/(4*(r^2)*R*(zeta^2))+G12*(t^3)*(gammar^2)*st2/(12*(r^3)*(zeta^3)
)+G12*(t^3)*gammar*st/(24*(r^3)*(zeta^3))
-G12*(t^3)*gammar*st/(12*(r^3)*(zeta^2))
+f1*(t^3)*st*ct*nu21*gammar/(12*(r^2)*R*(zeta^3))*(1/nu12+1))*k;
ain(n,n)=-(k^2)*(-f1*(t^2)*(-t2*st)*nu21/(4*(r^3)*(zeta^2))
-G12*(t^3)/(6*(r^2)*R*(zeta^3))
-f1*(t^3)*gammar*st*nu21*(1/nu12+1)/(12*(r^3)*(zeta^3)));
ami(n,n)=(-k)*(G12*t/(2*r*zeta)+G12*(t^3)*gammar*ct/(48*(r^3)*(zeta^3))
+G12*(t^3)*gammar*ct/(24*(r^3)*zeta));
amn(n,n)=(-k)*(-G12*(t^3)*gammar*ct/(12*(r^2)*R*(zeta^3)));
amm(n,n)=G12*gammar*st*t/(2*r*zeta)+G12*(t^3)*st2*ct/(24*(r^3)*(zeta^3))
+G12*(t^3)*(gammar^2)*ct*st/(48*(r^3)*(zeta^3))
-G12*(t^3)*(zeta^2)*ct*st/(24*(r^3)*(zeta^2));
ani(n,n)=-f1*t*nu21*gammar*st/(r*zeta);
anm(n,n)=k*(f1*nu21*t/(r*zeta));
ann(n,n)=f1*t/r+f1*t*nu21*gammar*ct/(r*zeta);
% terms which involve the phi derivatives
for jj=1:n
aii(n,jj)=aii(n,jj)+(f1*(t^2)*(-t2*st)/(4*(r^3))
-f1*(t^3)*gammar*st/(12*(r^3)*zeta))*b1(n,jj)
+(f1*(t^3)/(12*(r^3)))*b2(n,jj);
aim(n,jj)=aim(n,jj)+k*(f1*(t^3)*nu21*ct/(12*(r^2)*R*(zeta^2))
+G12*(t^3)*gammar*ct/(12*(r^3)*(zeta^2))
-G12*(t^3)/(24*(r^3)*(zeta^2)))*b1(n,jj);
ain(n,jj)=ain(n,jj)+(f1*(t^2)*(-t2*st)*nu21*st/(4*(r^2)*R*zeta)
+f1*(t^3)*gammar*(st^2)*nu21/(12*(r^2)*R*(zeta^2)*nu12)
-(k^2)*(-f1*(t^3)*nu21/(12*(r^3)*(zeta^2))
-G12*(t^3)/(6*(r^3)*(zeta^2))))*b1(n,jj)
+(-f1*(t^2)*(-t2*st)/(4*(r^3))+gammar*st*f1*(t^3)/(12*(r^3)*zeta))*b2(n,jj)
+(-f1*(t^3)/(12*(r^3)))*b3(n,jj);
amm(n,jj)=amm(n,jj)+(G12*t/(4*(r^2)*zeta)
+(gammar^2)*(ct^2)*G12*(t^3)/(24*(r^3)*(zeta^2))
-G12*(t^3)*gammar*ct/(48*(r^3)*(zeta^2)))*b1(n,jj);
amn(n,jj)=amn(n,jj)+(-k)*(-G12*(t^3)*gammar*ct/(12*(r^3)*(zeta^2)))*b1(n,jj);
ani(n,jj)=ani(n,jj)+(f1*t/r)*b1(n,jj);
end
tee=x(1); ct=cos(tee); st=sin(tee); st2=2*st*ct; ct2=(ct^2)-(st^2);
zeta=1+gammar*ct; t=t1-t2*ct;
% terms which do not involve derivatives
aii(n,1)=aii(n,1)-(-gammar*(st^2)*f1*(t^3)*nu21/(12*(r^2)*R*(zeta^2)*nu12)
-f1*nu21*st*(t^2)*(-t2*st)/(4*(r^2)*R*zeta)
-(k^2)*(G12*(t^3)/(24*(r^3)*(zeta^3))
+G12*(t^3)/(12*(r^3)*(zeta^2))));
aim(n,1)=aim(n,1)-((f1*(t^2)*(-t2*st)*nu21*ct/(4*(r^2)*R*(zeta^2))
+G12*(t^3)*(gammar^2)*st2/(12*(r^3)*(zeta^3))+G12*(t^3)*gammar*st/(24*(r^3)*(ze
ta^3))-G12*(t^3)*gammar*st/(12*(r^3)*(zeta^2))
+f1*(t^3)*st*ct*nu21*gammar/(12*(r^2)*R*(zeta^3))*(1/nu12+1))*k);
ain(n,1)=ain(n,1)-(-(k^2)*(-f1*(t^2)*(-t2*st)*nu21/(4*(r^3)*(zeta^2))
-G12*(t^3)/(6*(r^2)*R*(zeta^3))
-f1*(t^3)*gammar*st*nu21*(1/nu12+1)/(12*(r^3)*(zeta^3))));
ami(n,1)=ami(n,1)-((-
k)*(G12*t/(2*r*zeta)+G12*(t^3)*gammar*ct/(48*(r^3)*(zeta^3))
+G12*(t^3)*gammar*ct/(24*(r^3)*zeta)));
amn(n,1)=amn(n,1)-((-k)*(-G12*(t^3)*gammar*ct/(12*(r^2)*R*(zeta^3))));
amm(n,1)=amm(n,1)-
(G12*gammar*st*t/(2*r*zeta)+G12*(t^3)*st2*ct/(24*(r^3)*(zeta^3))
```

```
+G12*(t^3)*(gammar^2)*ct*st/(48*(r^3)*(zeta^3))
-G12*(t^3)*(zeta^2)*ct*st/(24*(r^3)*(zeta^2)));
ani(n,1)=ani(n,1)-(-f1*t*nu21*gammar*st/(r*zeta));
anm(n,1)=anm(n,1)-(k*(f1*nu21*t/(r*zeta)));
ann(n,1)=ann(n,1)-(f1*t/r+f1*t*nu21*gammar*ct/(r*zeta));
% terms which invovle the phi derivatives
for jj=1:n
aii(n,jj)=aii(n,jj)-(f1*(t^2)*(-t2*st)/(4*(r^3))
-f1*(t^3)*gammar*st/(12*(r^3)*zeta))*b1(1,jj)+(f1*(t^3)/(12*(r^3)))*b2(1,jj);
aim(n,jj)=aim(n,jj)-k*(f1*(t^3)*nu21*ct/(12*(r^2)*R*(zeta^2))
+G12*(t^3)*gammar*ct/(12*(r^3)*(zeta^2))
-G12*(t^3)/(24*(r^3)*(zeta^2)))*b1(1,jj);
ain(n,jj)=ain(n,jj)-(f1*(t^2)*(-t2*st)*nu21*st/(4*(r^2)*R*zeta)
+f1*(t^3)*gammar*(st^2)*nu21/(12*(r^2)*R*(zeta^2)*nu12)
-(k^2)*(-f1*(t^3)*nu21/(12*(r^3)*(zeta^2))
-G12*(t^3)/(6*(r^3)*(zeta^2))))*b1(1,jj)
+(-f1*(t^2)*(-t2*st)/(4*(r^3))+gammar*st*f1*(t^3)/(12*(r^3)*zeta))*b2(1,jj)
+(-f1*(t^3)/(12*(r^3)))*b3(1,jj);
amm(n,jj)=amm(n,jj)-
(G12*t/(4*(r^2)*zeta)+(gammar^2)*(ct^2)*G12*(t^3)/(24*(r^3)*(zeta^2))
-G12*(t^3)*gammar*ct/(48*(r^3)*(zeta^2)))*b1(1,jj);
amn(n,jj)=amn(n,jj)-(-k)*(-G12*(t^3)*gammar*ct/(12*(r^3)*(zeta^2)))*b1(1,jj);
ani(n,jj)=ani(n,jj)-(f1*t/r)*b1(1,jj);
end
tee=x(n-1); ct=cos(tee); st=sin(tee); st2=2*st*ct; ct2=(ct^2)-(st^2);
zeta=1+gammar*ct; t=t1-t2*ct;
% terms which do not involve the derivatives
ani(n-1,n-1)=-f1*(t^3)*st*nu21/(12*r*R*zeta);
anm(n-1,n-1)=k*(f1*(t^3)*nu21*ct/(12*r*R*(zeta^2)));
ann(n-1,n-1)=-(k^2)*(-f1*(t^3)*nu21/(12*(r^2)*(zeta^2)));
% terms which involve the phi derivatives
for jj=1:n
ani(n-1,jj)=ani(n-1,jj)+(f1*(t^3)/(12*(r^2)))*b1(n-1,jj);
ann(n-1,jj)=ann(n-1,jj)+(f1*(t^3)*st*nu21/(12*r*R*zeta))*b1(n-1,jj)
+(-f1*(t^3)/(12*(r^2)))*b2(n-1,jj);
end
tee=x(2); ct=cos(tee); st=sin(tee); st2=2*st*ct; ct2=(ct^2)-(st^2);
zeta=1+gammar*ct; t=t1-t2*ct;
% terms which do not involve the derivatives
ani(n-1,2)=ani(n-1,2)-(-f1*(t^3)*st*nu21/(12*r*R*zeta));
anm(n-1,2)=anm(n-1,2)-k*(f1*(t^3)*nu21*ct/(12*r*R*(zeta^2)));
ann(n-1,2)=ann(n-1,2)-(k^2)*(-f1*(t^3)*nu21/(12*(r^2)*(zeta^2)));
% terms which involve the phi derivatives
for jj=1:n
ani(n-1,jj)=ani(n-1,jj)-(f1*(t^3)/(12*(r^2)))*b1(2,jj);
ann(n-1,jj)=ann(n-1,jj)-(f1*(t^3)*st*nu21/(12*r*R*zeta))*b1(2,jj)
+(-f1*(t^3)/(12*(r^2)))*b2(2,jj);
end
%-----------------------------------------------------------------------------
% section partition - partition matrices for eigenvalue extraction
% this step necessary for freq analysis to form principle ev problem
% form bb matrix - bdy eqns involving bdy unknows
bb=[aii(1,1) aii(1,n) aim(1,1) aim(1,n) ain(1,1) ain(1,2) ain(1,n-1) ain(1,n);
    aii(n,1) aii(n,n) aim(n,1) aim(n,n) ain(n,1) ain(n,2) ain(n,n-1) ain(n,n);
    ami(1,1) ami(1,n) amm(1,1) amm(1,n) amn(1,1) amn(1,2) amn(1,n-1) amn(1,n);
    ami(n,1) ami(n,n) amm(n,1) amm(n,n) amn(n,1) amn(n,2) amn(n,n-1) amn(n,n);
    ani(1,1) ani(1,n) anm(1,1) anm(1,n) ann(1,1) ann(1,2) ann(1,n-1) ann(1,n);
```

```
    ani(2,1) ani(2,n) anm(2,1) anm(2,n) ann(2,1) ann(2,2) ann(2,n-1) ann(2,n);
    ani(n-1,1) ani(n-1,n) anm(n-1,1) anm(n-1,n) ann(n-1,1) ann(n-1,2) ann(n-
1,n-1) ann(n-1,n);
    ani(n,1) ani(n,n) anm(n,1) anm(n,n) ann(n,1) ann(n,2) ann(n,n-1) ann(n,n)];
% form bd matrix - bdy eqns involving domain unknows
er=2:n-1; cr=3:n-2;
bd=[aii(1,er) aim(1,er) ain(1,cr);
    aii(n,er) aim(n,er) ain(n,cr);
    ami(1,er) amm(1,er) amn(1,cr);
    ami(n,er) amm(n,er) amn(1,cr);
    ani(1,er) anm(1,er) ann(1,cr);
    ani(2,er) anm(2,er) ann(2,cr);
    ani(n-1,er) anm(n-1,er) ann(n-1,cr);
    ani(n,er) anm(n,er) ann(n,cr)];
% form db matrix - domain eqns involving bdy unknows
db=[aii(er,1) aii(er,n) aim(er,1) aim(er,n) ain(er,1) ain(er,2) ain(er,n-1)
ain(er,n);
    ami(er,1) ami(er,n) amm(er,1) amm(er,n) amn(er,1) amn(er,2) amn(er,n-1)
amn(er,n);
    ani(cr,1) ani(cr,n) anm(cr,1) anm(cr,n) ann(cr,1) ann(cr,2) ann(cr,n-1)
ann(cr,n)];
% form dd matrix - domain eqns involving domain unknows
dd=[aii(er,er) aim(er,er) ain(er,cr);
    ami(er,er) amm(er,er) amn(er,cr);
    ani(cr,er) anm(cr,er) ann(cr,cr)];
clear aii aim ain ami amm amn ani anm ann;
%-----------------------------------------------------------------------
% section eign
prod1=-inv(bb)*bd; clear bb bd;
prod2=db*prod1; clear db prod1;
stiff=-(dd+prod2); clear dd prod2;
% find eignvalues
[evc, evl]=eig(stiff);  clear stiff;
tmn=3*n-8;
kk=0;
for i=1:tmn
if imag(evl(i,i)) == 0
if evl(i,i) > 0; kk=kk+1; ezz(kk)=evl(i,i); end
end
end
ezzz=ezz(1:kk);
% sort the eigenvalues and output the ten lowest
sval=sort(ezzz);
for j=1:10; result(j)=sqrt(sval(j)/den); end
%-----------------------------------------------------------------------
% section result
result
```

APPENDIX D - Listing of program Afreqiso2.m

```
% AFreqiso2.m  last modification - Feb. 25, 2002
% Sanders shell theory using 2 gov eqns comp & struct 43 (1992) 1019-1028.
% axisymmetric natural frequencies of steel torus using DQM
% harmonics trial function used in meridional direction
%-------------------------------------------------------------------------------
% section input
% n=sampling points in meridional direction;
% m=sampling points in circumferential direction;
% E1=Young's modulus (Pa); G12=shear moduli (Pa);
% nu12=nu21=Poisson's ratio; den=mass density (kg/m3);
% r=x-section radius (m); R=bend radius (m);
% t1 t2=thickness coefficients;
format long
n=96
m=180;
r=0.25; R=0.762;  E1=2.07e11; nu12=0.3; nu21=0.3; f1=E1/(1-nu21*nu12);
gammar=r/R; t1=0.009525; t2=0.001; G12=E1/(2*(1+nu12)); den=7770;
%-------------------------------------------------------------------------------
% section wcmd - find the weighting coeffts for meridional direction
% unequal spacing with one delta point at each end for phi direction
% output - b1(i,j),b2(i,j),b3(i,j),b4(i,j) for 1st to 4th derivats
y(1)=0; y(2)=.00001; y(n-1)=.99999;y(n)=1;
%
for i=3:n-2
tt=(pi*(i-2))/(n-3);y(i)=(1-cos(tt))/2;
end
%
for i=1:n; x(i)=xlo+y(i)*xdif; end
%
for i=1:n; pai(i)=1; for j=1:n
if i~=j; pai(i)=(x(i)-x(j))*pai(i); end
end;end
%
for i=1:n; for j=1:n
if i~=j; b1(i,j)=pai(i)/((x(i)-x(j))*pai(j)); end
end;end
%
for i=1:n; b1(i,i)=0; for j=1:n
if i~=j; b1(i,i)=b1(i,i)-b1(i,j); end
end;end
%
for i=1:n; for j=1:n
if i~=j; b2(i,j)=2*(b1(i,i)*b1(i,j)-b1(i,j)/(x(i)-x(j))); end
end;end
%
for i=1:n; b2(i,i)=0; for j=1:n
if i~=j; b2(i,i)=b2(i,i)-b2(i,j); end
end;end
%
for i=1:n; for j=1:n
if i~=j; b3(i,j)=3*(b2(i,i)*b1(i,j)-b2(i,j)/(x(i)-x(j))); end
end;end
%
for i=1:n; b3(i,i)=0; for j=1:n
if i~=j; b3(i,i)=b3(i,i)-b3(i,j); end
```

```
end;end
%
for i=1:n; for j=1:n
if i~=j; b4(i,j)=4*(b3(i,i)*b1(i,j)-b3(i,j)/(x(i)-x(j))); end
end;end
%
for i=1:n; b4(i,i)=0; for j=1:n
if i~=j; b4(i,i)=b4(i,i)-b4(i,j); end
end;end
%------------------------------------------------------------------------------------
% section domeqn - set up matrices enforcing the governing equations
%    aii    ain     u       0
%
%    ani    ann     w       0
% output - aii(i,j),...,ann(i,j) - coefft matrices for displacements
%
aii(:,:)=zeros(n); ain(:,:)=zeros(n);
ani(:,:)=zeros(n); ann(:,:)=zeros(n);
%
%loop over sampling points in the phi direction
for j=1:n
tee=tet(j); ct=cos(tee); st=sin(tee); st2=2*st*ct;
ct2=(ct^2)-(st^2);  zeta=1+gammar*ct; t=t1+t2*ct;
% terms which do not involve derivatives
aii(j,j)=(-nu21*f1*(-t2*st)*gammar*st/r-nu21*f1*t*gammar*ct/r
-(gammar^2)*(st^2)*f1*t*nu21/(r*zeta*nu12)
-nu21*f1*(t^2)*(-t2*st)*st/(4*(r^2)*R)-f1*nu21*(t^3)*ct/(12*(r^2)*R)
-f1*(t^3)*gammar*nu21*(st^2)/(12*nu12*(r^2)*R*zeta))*r/((r^2)*zeta*t);
ain(j,j)=(nu21*f1*(-t2*st)*gammar*ct/r+f1*(-t2*st)*zeta/r
-f1*t*gammar*st/r+f1*t*(gammar^2)*st*ct*nu21/(r*zeta*nu12))*r/((r^2)*zeta*t);
ani(j,j)=(-f1*t*nu21*gammar*st/((r^2)*zeta)
-f1*t*(gammar^2)*st*ct*nu21/((r^2)*(zeta^2)*nu12)
+f1*nu21*t*((t2*st)^2)*st/(2*(r^3)*R*zeta)
+nu21*f1*(t^2)*(-t2*ct)*st/(4*(r^3)*R*zeta)
+nu21*f1*(-t2*st)*(t^2)*ct/((r^3)*R*zeta*2)
-f1*(t^3)*nu21*st/(12*(r^3)*R*zeta)
+f1*(t^2)*(-t2*st)*gammar*(st^2)*nu21/(4*(r^3)*(zeta^2)*R*nu12)
+f1*(t^3)*gammar*st*ct*nu21/(6*(r^3)*(zeta^2)*R*nu12)
+f1*(t^3)*nu21*(gammar^2)*(st^3)/(12*nu12*(r^3)*R*(zeta^3)))*(-
(r^2)*zeta)/((r^2)*zeta*t);
ann(j,j)=(f1*t*(ct^2)*nu21/((R^2)*(zeta^2)*nu12)
+2*nu21*f1*t*ct/(r*R*zeta)+f1*t/(r^2))*(-(r^2)*zeta)/((r^2)*zeta*t);
%
% terms which involve the phi derivatives
for jj=1:n
aii(j,jj)=aii(j,jj)+((f1*(-t2*st)*zeta/r-f1*t*gammar*st/r
+f1*(t^2)*(-t2*st)*zeta/(4*(r^3))-f1*(t^3)*gammar*st/(12*(r^3)))*b1(j,jj)
+(f1*t*zeta/r+f1*(t^3)*zeta/(12*(r^3)))*b2(j,jj))*r/((r^2)*zeta*t);
ain(j,jj)=ain(j,jj)+((f1*nu21*t*gammar*ct/r+f1*t*zeta/r
+nu21*f1*(t^2)*(-t2*st)*st/(4*(r^2)*R)+nu21*f1*(t^3)*ct/(12*(r^2)*R)
+f1*(t^3)*nu21*gammar*(st^2)/(12*nu12*(r^2)*R*zeta))*b1(j,jj)
+(-f1*(t^2)*(-t2*st)*zeta/(4*(r^3))+f1*(t^3)*gammar*st/(12*(r^3)))*b2(j,jj)
+(-f1*(t^3)*zeta/(12*(r^3)))*b3(j,jj))*r/((r^2)*zeta*t);
ani(j,jj)=ani(j,jj)+((f1*t/(r^2)+ct*nu21*f1*t/(R*r*zeta)
+nu21*f1*(t^3)*ct/(12*(r^3)*R*zeta)
+nu21*f1*(t^2)*st*(-t2*st)/(4*(r^3)*R*zeta)-f1*(t^2)*(-t2*ct)/(4*(r^4))
-f1*t*((t2*st)^2)/(2*(r^4))
```

```
+f1*(t^3)*(st^2)*nu21/(12*(r^2)*(R^2)*(zeta^2)*nu12)
+f1*(t^2)*(-t2*st)*st/(2*(r^3)*R*zeta)
+f1*(t^3)*ct/(12*(r^3)*R*zeta))*b1(j,jj)
+(-f1*(t^2)*(-t2*st)/(2*(r^4))+f1*(t^3)*st/(6*(r^3)*R*zeta))*b2(j,jj)
+(-f1*(t^3)/(12*(r^4)))*b3(j,jj))*(-(r^2)*zeta)/((r^2)*zeta*t);
ann(j,jj)=ann(j,jj)+((nu21*f1*(t^3)*st/(12*(r^3)*R*zeta)
-nu21*f1*t*st*((t2*st)^2)/(2*(r^3)*R*zeta)
-nu21*f1*(t^2)*ct*(-t2*st)/(2*(r^3)*R*zeta)
-nu21*f1*(t^2)*st*(-t2*ct)/(4*(r^3)*R*zeta)
-nu21*f1*(t^3)*(st^3)/(12*nu12*r*(R^3)*(zeta^3))
-f1*(t^2)*(st^2)*(-t2*st)*nu21/(4*(r^2)*(R^2)*(zeta^2)*nu12)
-f1*(t^3)*st*ct*nu21/(6*(r^2)*(R^2)*(zeta^2)*nu12))*b1(j,jj)
+(-nu21*f1*(t^3)*ct/(12*(r^3)*R*zeta)
-nu21*f1*(t^2)*st*(-t2*st)/(4*(r^3)*R*zeta)
+(t^2)*(-t2*ct)*f1/(4*(r^4))+f1*((t2*st)^2)*t/(2*(r^4))
-f1*(t^3)*(st^2)*nu21/(nu12*12*(r^2)*(R^2)*(zeta^2))
-f1*(t^2)*(-t2*st)*st/(2*(r^3)*R*zeta)
-f1*(t^3)*ct/(12*(r^3)*R*zeta))*b2(j,jj)
+(f1*(t^2)*(-t2*st)/(2*(r^4))
-f1*(t^3)*st/(6*(r^3)*R*zeta))*b3(j,jj)
+(f1*(t^3)/(12*(r^4)))*b4(j,jj))*(-(r^2)*zeta)/((r^2)*zeta*t);
end
end
clear j; clear jj;
%---------------------------------------------------------------------------
% section bound - enforcing the  four displacement relations
% terms which do not involve derivatives
aii(1,1)=-1;   aii(1,n)=1;
ann(1,1)=-1;   ann(1,n)=1;
%
% terms which involve the phi derivatives
for jj=1:n
ani(2,jj)=0;
ani(n-1,jj)=0;
ann(2,jj)=0;
ann(n-1,jj)=0;
end
for jj=1:n
ann(2,jj)=ann(2,jj)+(-b1(2,jj)+b1(n-1,jj));
end
%---------------------------------------------------------------------------
% section bound - enforce four force resultants relations
tee=x(n); ct=cos(tee); st=sin(tee); st2=2*st*ct; ct2=(ct^2)-(st^2);
zeta=1+gammar*ct; t=t1-t2*ct;
% terms which do not involve derivatives
aii(n,n)=-gammar*(st^2)*f1*(t^3)*nu21/(12*(r^2)*R*(zeta^2)*nu12)
-f1*nu21*st*(t^2)*(-t2*st)/(4*(r^2)*R*zeta)
-(k^2)*(G12*(t^3)/(24*(r^3)*(zeta^3))+G12*(t^3)/(12*(r^3)*(zeta^2)));
ain(n,n)=-(k^2)*(-f1*(t^2)*(-t2*st)*nu21/(4*(r^3)*(zeta^2))-
G12*(t^3)/(6*(r^2)*R*(zeta^3))
-f1*(t^3)*gammar*st*nu21*(1/nu12+1)/(12*(r^3)*(zeta^3)));
ani(n,n)=-f1*t*nu21*gammar*st/(r*zeta);
ann(n,n)=f1*t/r+f1*t*nu21*gammar*ct/(r*zeta);
% terms which involve the phi derivatives
for jj=1:n
aii(n,jj)=aii(n,jj)+(f1*(t^2)*(-t2*st)/(4*(r^3))-
f1*(t^3)*gammar*st/(12*(r^3)*zeta))*b1(n,jj)+(f1*(t^3)/(12*(r^3)))*b2(n,jj);
```

```
ain(n,jj)=ain(n,jj)+(f1*(t^2)*(-t2*st)*nu21*st/(4*(r^2)*R*zeta)
+f1*(t^3)*gammar*(st^2)*nu21/(12*(r^2)*R*(zeta^2)*nu12)
-(k^2)*(-f1*(t^3)*nu21/(12*(r^3)*(zeta^2))
-G12*(t^3)/(6*(r^3)*(zeta^2)))))*b1(n,jj)
+(-f1*(t^2)*(-t2*st)/(4*(r^3))+gammar*st*f1*(t^3)/(12*(r^3)*zeta))*b2(n,jj)
+(-f1*(t^3)/(12*(r^3)))*b3(n,jj);
ani(n,jj)=ani(n,jj)+(f1*t/r)*b1(n,jj);
end
tee=x(1); ct=cos(tee); st=sin(tee); st2=2*st*ct; ct2=(ct^2)-(st^2);
zeta=1+gammar*ct; t=t1-t2*ct;
% terms which do not involve derivatives
aii(n,1)=aii(n,1)-(-gammar*(st^2)*f1*(t^3)*nu21/(12*(r^2)*R*(zeta^2)*nu12)
-f1*nu21*st*(t^2)*(-t2*st)/(4*(r^2)*R*zeta)
-(k^2)*(G12*(t^3)/(24*(r^3)*(zeta^3))+G12*(t^3)/(12*(r^3)*(zeta^2))));
ain(n,1)=ain(n,1)-(-(k^2)*(-f1*(t^2)*(-t2*st)*nu21/(4*(r^3)*(zeta^2))
-G12*(t^3)/(6*(r^2)*R*(zeta^3))
-f1*(t^3)*gammar*st*nu21*(1/nu12+1)/(12*(r^3)*(zeta^3))));
ani(n,1)=ani(n,1)-(-f1*t*nu21*gammar*st/(r*zeta));
ann(n,1)=ann(n,1)-(f1*t/r+f1*t*nu21*gammar*ct/(r*zeta));
% terms which invovle the phi derivatives
for jj=1:n
aii(n,jj)=aii(n,jj)-(f1*(t^2)*(-t2*st)/(4*(r^3))
-f1*(t^3)*gammar*st/(12*(r^3)*zeta))*b1(1,jj)+(f1*(t^3)/(12*(r^3)))*b2(1,jj);
ain(n,jj)=ain(n,jj)-(f1*(t^2)*(-t2*st)*nu21*st/(4*(r^2)*R*zeta)
+f1*(t^3)*gammar*(st^2)*nu21/(12*(r^2)*R*(zeta^2)*nu12)
-(k^2)*(-f1*(t^3)*nu21/(12*(r^3)*(zeta^2))
-G12*(t^3)/(6*(r^3)*(zeta^2)))))*b1(1,jj)
+(-f1*(t^2)*(-t2*st)/(4*(r^3))+gammar*st*f1*(t^3)/(12*(r^3)*zeta))*b2(1,jj)
+(-f1*(t^3)/(12*(r^3)))*b3(1,jj);
ani(n,jj)=ani(n,jj)-(f1*t/r)*b1(1,jj);
end
tee=x(n-1); ct=cos(tee); st=sin(tee); st2=2*st*ct; ct2=(ct^2)-(st^2);
zeta=1+gammar*ct; t=t1-t2*ct;
% terms which do not involve the derivatives
ani(n-1,n-1)=-f1*(t^3)*st*nu21/(12*r*R*zeta);
ann(n-1,n-1)=-(k^2)*(-f1*(t^3)*nu21/(12*(r^2)*(zeta^2)));
% terms which involve the phi derivatives
for jj=1:n
ani(n-1,jj)=ani(n-1,jj)+(f1*(t^3)/(12*(r^2)))*b1(n-1,jj);
ann(n-1,jj)=ann(n-1,jj)+(f1*(t^3)*st*nu21/(12*r*R*zeta))*b1(n-1,jj)
+(-f1*(t^3)/(12*(r^2)))*b2(n-1,jj);
end
tee=x(2); ct=cos(tee); st=sin(tee); st2=2*st*ct; ct2=(ct^2)-(st^2);
zeta=1+gammar*ct; t=t1-t2*ct;
% terms which do not involve the derivatives
ani(n-1,2)=ani(n-1,2)-(-f1*(t^3)*st*nu21/(12*r*R*zeta));
ann(n-1,2)=ann(n-1,2)-(k^2)*(-f1*(t^3)*nu21/(12*(r^2)*(zeta^2)));
% terms which involve the phi derivatives
for jj=1:n
ani(n-1,jj)=ani(n-1,jj)-(f1*(t^3)/(12*(r^2)))*b1(2,jj);
ann(n-1,jj)=ann(n-1,jj)-(f1*(t^3)*st*nu21/(12*r*R*zeta))*b1(2,jj)
+(-f1*(t^3)/(12*(r^2)))*b2(2,jj);
end
%-----------------------------------------------------------------------------
% section partition - partition matrices for eigenvalue extraction
% this step necessary for freq analysis to form principle ev problem
% form bb matrix - bdy eqns involving bdy unknows
```

```
bb=[aii(1,1) aii(1,n) ain(1,1) ain(1,2) ain(1,n-1) ain(1,n);
    aii(n,1) aii(n,n) ain(n,1) ain(n,2) ain(n,n-1) ain(n,n);
    ani(1,1) ani(1,n) ann(1,1) ann(1,2) ann(1,n-1) ann(1,n);
    ani(2,1) ani(2,n) ann(2,1) ann(2,2) ann(2,n-1) ann(2,n);
    ani(n-1,1) ani(n-1,n) ann(n-1,1) ann(n-1,2) ann(n-1,n-1) ann(n-1,n);
    ani(n,1) ani(n,n) ann(n,1) ann(n,2) ann(n,n-1) ann(n,n)];
% form bd matrix - bdy eqns involving domain unknows
er=2:n-1; cr=3:n-2;
bd=[aii(1,er) ain(1,cr);
    aii(n,er) ain(n,cr);
    ani(1,er) ann(1,cr);
    ani(2,er) ann(2,cr);
    ani(n-1,er) ann(n-1,cr);
    ani(n,er) ann(n,cr)];
% form db matrix - domain eqns involving bdy unknows
db=[aii(er,1) aii(er,n) ain(er,1) ain(er,2) ain(er,n-1) ain(er,n);
    ani(cr,1) ani(cr,n) ann(cr,1) ann(cr,2) ann(cr,n-1) ann(cr,n)];
% form dd matrix - domain eqns involving domain unknows
dd=[aii(er,er) ain(er,cr);
    ani(cr,er) ann(cr,cr)];
clear aii ain ani ann;
%--------------------------------------------------------------------------
% section eign
prod1=-inv(bb)*bd; clear bb bd;
prod2=db*prod1; clear db prod1;
stiff=-(dd+prod2); clear dd prod2;
% find eignvalues
[evc, evl]=eig(stiff);  clear stiff;
tmn=2*n-6;
kk=0;
for i=1:tmn
if imag(evl(i,i)) == 0
if evl(i,i) > 0; kk=kk+1; ezz(kk)=evl(i,i); end
end
end
ezzz=ezz(1:kk);
% sort the eigenvalues and output the ten lowest
sval=sort(ezzz);
for j=1:10; result(j)=sqrt(sval(j)/den); end
%--------------------------------------------------------------------------
% section result
result
```

APPENDIX D - Listing of program Pvessel.m

```
% Pvessel.m  last modification - March 10, 2002
% Sanders shell theory using 2 gov eqns comp & struct 43 (1992) 1019-1028.
% elastostatics of steel torus - internal pressure using DQM
%---------------------------------------------------------------------------------
% section input
% n=sampling points in meridional direction;
% E1=Young's modulus in meridional direction (Pa);
% G12=shear moduli (Pa); p0=internal pressure (Pa);
% nu12=nu21=Poisson's ratio, den=mass density (kg/m3);
% r=x-section radius (m); R=bend radius(m);
% t1 t2=thickness coefficients,
format long
n=36
r=0.25; R=0.762;  E1=2.07e11; nu12=0.3; nu21=0.3; f1=E1/(1-nu21*nu12); p0=1e6;
gammar=r/R; t1=0.009525; t2=0; G12=E1/(2*(1+nu12))
%---------------------------------------------------------------------------------
% section wcmd - find the weight coeffts for the meridional direction
% sampling points are equally spaced in theta direction
% output - b1(i,j),b2(i,j),b3(i,j),b4(i,j) for 1st to 4th deriv
n2=n/2;
%
for i=1:n
tet(i)=2*pi*(i-1)/n; ib(i,1)=.5; ib(i,n2+1)=.5;
end
%
for i=2:2:n; ib(i,n2+1)=-0.5; end
%
for i=2:n2; ib(1,i)=1; end
%
for i=n2+2:n; ib(1,i)=0; end
%
for i=2:n; for j=n2+2:n
ib(i,j)=sin(2*pi*(i-1)*(j-1-n2)/n);
end; end
%
for i=2:n; for j=2:n2
ib(i,j)=cos(2*pi*(i-1)*(j-1)/n);
end; end
%
ib=ib/n2;
for i=1:n; for j=1:n
if abs(ib(i,j))<1e-12; ib(i,j)=0; end
end;end
%
for i=1:n; for j=1:n2+1
rr1(j)=(1-j)*sin((j-1)*tet(i));
rr2(j)=-1*((j-1)^2)*cos((j-1)*tet(i));
rr3(j)=((j-1)^3)*sin((j-1)*tet(i));
rr4(j)=((j-1)^4)*cos((j-1)*tet(i));
end
for j=n2+2:n
rr1(j)=(j-1-n2)*cos((j-1-n2)*tet(i));
rr2(j)=-1*((j-1-n2)^2)*sin((j-1-n2)*tet(i));
rr3(j)=-1*((j-1-n2)^3)*cos((j-1-n2)*tet(i));
rr4(j)=((j-1-n2)^4)*sin((j-1-n2)*tet(i));
```

```
end
%
b1(i,:)=(ib*rr1')';
b2(i,:)=(ib*rr2')';
b3(i,:)=(ib*rr3')';
b4(i,:)=(ib*rr4')';
end
%-------------------------------------------------------------------------
% section domeqn - set up matrices enforcing the governing equations
%    aii    ain     u       0
%
%    ani    ann     w       0
% output - aii(i,j),...,ann(i,j) - coefft matrices for displacements
%
aii(:,:)=zeros(n); ain(:,:)=zeros(n);
ani(:,:)=zeros(n); ann(:,:)=zeros(n);
%
%loop over sampling points in the phi direction
for j=1:n
tee=tet(j); ct=cos(tee); st=sin(tee); st2=2*st*ct; ct2=(ct^2)-(st^2);
zeta=1+gammar*ct; t=t1+t2*ct;
% terms which do not involve derivatives
aii(j,j)=-nu21*f1*(-t2*st)*gammar*st/r-nu21*f1*t*gammar*ct/r
-(gammar^2)*(st^2)*f1*t*nu21/(r*zeta*nu12)
-nu21*f1*(t^2)*(-t2*st)*st/(4*(r^2)*R)
-f1*nu21*(t^3)*ct/(12*(r^2)*R)
-f1*(t^3)*gammar*nu21*(st^2)/(12*nu12*(r^2)*R*zeta);
ain(j,j)=nu21*f1*(-t2*st)*gammar*ct/r+f1*(-t2*st)*zeta/r
-f1*t*gammar*st/r+f1*t*(gammar^2)*st*ct*nu21/(r*zeta*nu12);
ani(j,j)=-f1*t*nu21*gammar*st/((r^2)*zeta)
-f1*t*(gammar^2)*st*ct*nu21/((r^2)*(zeta^2)*nu12)
+f1*nu21*t*((t2*st)^2)*st/(2*(r^3)*R*zeta)
+nu21*f1*(t^2)*(-t2*ct)*st/(4*(r^3)*R*zeta)
+nu21*f1*(-t2*st)*(t^2)*ct/((r^3)*R*zeta*2)-f1*(t^3)*nu21*st/(12*(r^3)*R*zeta)
+f1*(t^2)*(-t2*st)*gammar*(st^2)*nu21/(4*(r^3)*(zeta^2)*R*nu12)
+f1*(t^3)*gammar*st*ct*nu21/(6*(r^3)*(zeta^2)*R*nu12)
+f1*(t^3)*nu21*(gammar^2)*(st^3)/(12*nu12*(r^3)*R*(zeta^3));
ann(j,j)=f1*t*(ct^2)*nu21/((R^2)*(zeta^2)*nu12)
+2*nu21*f1*t*ct/(r*R*zeta)+f1*t/(r^2);
%
% terms which involve the phi derivatives
for jj=1:n
aii(j,jj)=aii(j,jj)+(f1*(-t2*st)*zeta/r-f1*t*gammar*st/r
+f1*(t^2)*(-t2*st)*zeta/(4*(r^3))-f1*(t^3)*gammar*st/(12*(r^3)))*b1(j,jj)
+(f1*t*zeta/r+f1*(t^3)*zeta/(12*(r^3)))*b2(j,jj);
ain(j,jj)=ain(j,jj)+(f1*nu21*t*gammar*ct/r+f1*t*zeta/r
+nu21*f1*(t^2)*(-t2*st)*st/(4*(r^2)*R)+nu21*f1*(t^3)*ct/(12*(r^2)*R)
+f1*(t^3)*nu21*gammar*(st^2)/(12*nu12*(r^2)*R*zeta))*b1(j,jj)
+(-f1*(t^2)*(-t2*st)*zeta/(4*(r^3))+f1*(t^3)*gammar*st/(12*(r^3)))*b2(j,jj)
+(-f1*(t^3)*zeta/(12*(r^3)))*b3(j,jj);
ani(j,jj)=ani(j,jj)+(f1*t/(r^2)+ct*nu21*f1*t/(R*r*zeta)
+nu21*f1*(t^3)*ct/(12*(r^3)*R*zeta)+nu21*f1*(t^2)*st*(-t2*st)/(4*(r^3)*R*zeta)
-f1*(t^2)*(-t2*ct)/(4*(r^4))-f1*t*((t2*st)^2)/(2*(r^4))
+f1*(t^3)*(st^2)*nu21/(12*(r^2)*(R^2)*(zeta^2)*nu12)
+f1*(t^2)*(-t2*st)*st/(2*(r^3)*R*zeta)
+f1*(t^3)*ct/(12*(r^3)*R*zeta))*b1(j,jj)
+(-f1*(t^2)*(-t2*st)/(2*(r^4))+f1*(t^3)*st/(6*(r^3)*R*zeta))*b2(j,jj)
```

```
+(-f1*(t^3)/(12*(r^4)))*b3(j,jj);
ann(j,jj)=ann(j,jj)+(nu21*f1*(t^3)*st/(12*(r^3)*R*zeta)
-nu21*f1*t*st*((t2*st)^2)/(2*(r^3)*R*zeta)
-nu21*f1*(t^2)*ct*(-t2*st)/(2*(r^3)*R*zeta)
-nu21*f1*(t^2)*st*(-t2*ct)/(4*(r^3)*R*zeta)
-nu21*f1*(t^3)*(st^3)/(12*nu12*r*(R^3)*(zeta^3))
-f1*(t^2)*(st^2)*(-t2*st)*nu21/(4*(r^2)*(R^2)*(zeta^2)*nu12)
-f1*(t^3)*st*ct*nu21/(6*(r^2)*(R^2)*(zeta^2)*nu12))*b1(j,jj)
+(-nu21*f1*(t^3)*ct/(12*(r^3)*R*zeta)
-nu21*f1*(t^2)*st*(-t2*st)/(4*(r^3)*R*zeta)
+(t^2)*(-t2*ct)*f1/(4*(r^4))+f1*((t2*st)^2)*t/(2*(r^4))
-f1*(t^3)*(st^2)*nu21/(nu12*12*(r^2)*(R^2)*(zeta^2))
-f1*(t^2)*(-t2*st)*st/(2*(r^3)*R*zeta)-f1*(t^3)*ct/(12*(r^3)*R*zeta))*b2(j,jj)
+(f1*(t^2)*(-t2*st)/(2*(r^4))-f1*(t^3)*st/(6*(r^3)*R*zeta))*b3(j,jj)
+(f1*(t^3)/(12*(r^4)))*b4(j,jj);
end
end
%-----------------------------------------------------------------------------
% section loadv - set up the load vector for the right hand side
rhss=zeros(2*n,1);
for i=n+1:2*n
rhss(i,1)=-p0;
end
%-----------------------------------------------------------------------------
% section solu - combine submat - solv matrix eqn - find results
% find the vector of displacement components rslt
ar=1:n;
stiff=[aii(ar,ar)  ain(ar,ar);
       ani(ar,ar)  ann(ar,ar)]; clear aii  ain  ani  ann;
rslt=stiff\rhss;   clear rhss;  clear stiff;
%
% displacement components at the sampling points, u,v
ut=zeros(n,1); vt=zeros(n,1); wt=zeros(n,1);
for i=1:n
ut(i,1)=rslt(i,1);  wt(i,1)=rslt(i+n,1);
end
%-----------------------------------------------------------------------------
% section stress
sig1=zeros(n,1);  sig2=zeros(n,1);
for j=1:n
tee=tet(j); st=sin(tee); ct=cos(tee); zeta=1+gammar*ct;
ethetal=wt(j,1)/r; t=t1+t2*ct; kee=E1*t/(1-nu12^2); Dee=E1*(t^3)/(12*(1-
nu12^2));
eetal=(-gammar*st*ut(j,1)+gammar*ct*wt(j,1))/(r*zeta);
kthetal=0;
ketal=-st*ut(j,1)/(r*R*zeta);
for jj=1:n
ethetal=ethetal+ut(jj,1)*b1(j,jj)/r;
kthetal=kthetal+(ut(jj,1)*b1(j,jj)-wt(jj,1)*b2(j,jj))/(r^2);
ketal=ketal+st*wt(jj,1)*b1(j,jj)/(r*R*zeta);
end
sig1(j)=f1*(ethetal+nu12*eetal);
sig2(j)=f1*(eetal+nu12*ethetal);
end
%-----------------------------------------------------------------------------
% section results
sig2
```

APPENDIX D - Listing of program Freqc.m

```
% Freqc.m  last modification - Feb. 25, 2002
% Sanders shell theory using 3 gov eqns comp & struct 43 (1992) 1019-1028.
% natural frequencies and mode shapes of orhtropic toroidal shell using DQM
% Fourier series written in circumferential direction
%--------------------------------------------------------------------------------------
% section input
% n=stations in meridional direction;
% m=stations in circumferential direction;
% k=truncation integer*gammar;
% E1=Young's modulus in meridional direction (Pa);
% G12=shear moduli (Pa);
% nu12=Poisson's ratio, nu21=Poisson's ratio;
% den=mass density (kg/m3);
% r=x-section radius (m), R=bend radius (m);
% t1 t2=thickness coefficients
% format long
n=48
m=60
k=1/(0.38/0.12)
r=0.12; R=0.38;  E1=881570000; nu12=0.551; nu21=0.00525
f1=E1/(1-nu21*nu12); den=1190;gammar=r/R; G12=7850000
t1=0.008; t2=0.0008; xlo=-5/6*pi; xhi=5/6*pi; xdif=xhi-xlo
%--------------------------------------------------------------------------------------
% section wcmd - find the weighting coeffts for meridional direction
% unequal spacing with one delta point at each end for phi direction
% output - b1(i,j),b2(i,j),b3(i,j),b4(i,j) for 1st to 4th derivats
y(1)=0; y(2)=.00001; y(n-1)=.99999;y(n)=1;
%
for i=3:n-2
tt=(pi*(i-2))/(n-3);y(i)=(1-cos(tt))/2;
end
%
for i=1:n; x(i)=xlo+y(i)*xdif; end
%
for i=1:n; pai(i)=1; for j=1:n
if i~=j; pai(i)=(x(i)-x(j))*pai(i); end
end;end
%
for i=1:n; for j=1:n
if i~=j; b1(i,j)=pai(i)/((x(i)-x(j))*pai(j)); end
end;end
%
for i=1:n; b1(i,i)=0; for j=1:n
if i~=j; b1(i,i)=b1(i,i)-b1(i,j); end
end;end
%
for i=1:n; for j=1:n
if i~=j; b2(i,j)=2*(b1(i,i)*b1(i,j)-b1(i,j)/(x(i)-x(j))); end
end;end
%
for i=1:n; b2(i,i)=0; for j=1:n
if i~=j; b2(i,i)=b2(i,i)-b2(i,j); end
end;end
%
for i=1:n; for j=1:n
```

```
if i~=j; b3(i,j)=3*(b2(i,i)*b1(i,j)-b2(i,j)/(x(i)-x(j))); end
end;end
%
for i=1:n; b3(i,i)=0; for j=1:n
if i~=j; b3(i,i)=b3(i,i)-b3(i,j); end
end;end
%
for i=1:n; for j=1:n
if i~=j; b4(i,j)=4*(b3(i,i)*b1(i,j)-b3(i,j)/(x(i)-x(j))); end
end;end
%
for i=1:n; b4(i,i)=0; for j=1:n
if i~=j; b4(i,i)=b4(i,i)-b4(i,j); end
end;end
%-----------------------------------------------------------------------------
% section domeqn - set up matrices enforcing the governing equations
%    aii  aim  ain     u       0
%    ami  amm  amn     v   =   0
%    ani  anm  ann     w       0
% output - aii(i,j),...,ann(i,j) - coefft matrices for displacements
%
aii(:,:)=zeros(n);aim(:,:)=zeros(n);ain(:,:)=zeros(n);
ami(:,:)=zeros(n);amm(:,:)=zeros(n);amn(:,:)=zeros(n);
ani(:,:)=zeros(n);anm(:,:)=zeros(n);ann(:,:)=zeros(n);
%
%loop over sampling points in the phi direction
for j=1:n
tee=x(j); ct=cos(tee); st=sin(tee); st2=2*st*ct; ct2=(ct^2)-(st^2);
zeta=1+gammar*ct; t=t1+t2*ct;
% terms which do not involve derivatives
aii(j,j)=((-nu21*f1*(-t2*st)*gammar*st/r-nu21*f1*t*gammar*ct/r
-(gammar^2)*(st^2)*f1*t*nu21/(r*zeta*nu12)
-nu21*f1*(t^2)*(-t2*st)*st/(4*(r^2)*R)
-f1*nu21*(t^3)*ct/(12*(r^2)*R)
-f1*(t^3)*gammar*nu21*(st^2)/(12*nu12*(r^2)*R*zeta))*r/t
-((G12*t/(r*zeta)+G12*(t^3)/(24*(r^3)*(zeta^2))
+G12*(t^3)/(24*(r^3)*zeta)
+G12*(t^3)/(96*(r^3)*(zeta^3)))*(k^2))*r/t)/((r^2)*zeta);
aim(j,j)=(((f1*nu21*(-t2*st)/r+st*nu21*f1*t/(R*zeta*nu12)
+nu21*f1*(t^2)*ct*(-t2*st)/(4*(r^2)*R*zeta)
-nu21*f1*(t^3)*st/(12*(r^2)*R*zeta)
+G12*(t^3)*st2/(24*(R^2)*r*(zeta^2))
+G12*t*st/(R*zeta)+G12*(t^3)*st2/(48*(R^2)*r*(zeta^3))
-G12*(t^3)*st/(24*R*(r^2)*zeta)
+G12*(t^3)*st/(96*R*(r^2)*(zeta^3))
+st2*nu21*f1*(t^3)/(24*(R^2)*r*(zeta^2))
+nu21*f1*(t^3)*st2/(24*(R^2)*r*nu12*(zeta^2)))*k)*r/t)/((r^2)*zeta);
ain(j,j)=((nu21*f1*(-t2*st)*gammar*ct/r+f1*(-t2*st)*zeta/r
-f1*t*gammar*st/r+f1*t*(gammar^2)*st*ct*nu21/(r*zeta*nu12))*r/t
-((-nu21*f1*(t^2)*(-t2*st)/(4*(r^3)*zeta)
-nu21*f1*(t^3)*st/(12*(r^2)*(zeta^2)*R)-G12*(t^3)/(12*(r^2)*R*(zeta^2))
-G12*(t^3)/(24*(r^2)*R*(zeta^3))
-st*f1*nu21*(t^3)/(12*nu12*R*(r^2)*(zeta^2)))*(k^2))*r/t)/((r^2)*zeta);
ami(j,j)=(((-st*f1*nu21*t/(R*nu12*zeta)-G12*t*st/(R*zeta)
-f1*nu21*(st2)*(t^3)/(24*(R^2)*r*(zeta^2)*nu12)+G12*(-t2*st)/r
+G12*(t^2)*(-t2*st)/(8*(r^3))-G12*(t^2)*(-t2*st)/(8*(r^3)*zeta)
-3*G12*(t^2)*(-t2*st)/(32*(r^3)*(zeta^2))
```

```
+G12*(t^3)*st/(24*(r^2)*R*zeta)-7*G12*(t^3)*st/(96*(r^2)*R*(zeta^3))
-st2*G12*(t^3)/(48*r*(R^2)*(zeta^3))
-st2*G12*(t^3)/(24*(r^2)*(R^2)*(zeta^2))
-st*G12*(t^3)/(24*(r^2)*R*(zeta^2)))*(-k))*r/t)/((r^2)*zeta);
amm(j,j)=((-2*G12*t*r*(st^2)/((R^2)*zeta)+G12*ct*t/R
+G12*t*r*(st^2)/((R^2)*zeta)+G12*(t^2)*st2*(-t2*st)/(8*(R^2)*r*zeta)
-3*G12*(t^2)*st2*(-t2*st)/(16*(R^2)*r*(zeta^2))
-G12*(t^2)*st*(-t2*st)/(8*R*(r^2))+G12*(t^2)*st*(-t2*st)/(4*R*(r^2)*zeta)
-3*G12*(t^2)*st*(-t2*st)/(32*R*(r^2)*(zeta^2))
+G12*(t^3)*ct2/(12*(R^2)*r*zeta)
-G12*(t^3)*ct2/(8*(R^2)*r*(zeta^2))
-G12*(t^3)*ct/(24*(r^2)*R)
+G12*(t^3)*ct/(12*(r^2)*R*zeta)
-G12*(t^3)*ct/(32*(r^2)*R*(zeta^2))
-G12*(t^3)*(st^2)/(24*(R^2)*r*zeta)
+G12*st*(-t2*st)/R-r*(st2^2)*G12*(t^3)/(24*(R^4)*(zeta^3))
+G12*(t^3)*(st^2)/(8*(R^2)*r*(zeta^2))
-7*G12*(t^3)*(st^2)/(96*(R^2)*r*(zeta^3))
+G12*(t^3)*st*st2/(8*(R^3)*(zeta^2))
-G12*(t^3)*st*st2/(6*(R^3)*(zeta^3)))*r/t
-((ct^2)*nu21*f1*(t^3)/(12*(R^2)*r*(zeta^3)*nu12)
+f1*nu21*t/(r*zeta*nu12))*(k^2)*r/t)/((r^2)*zeta);
amn(j,j)=(((nu21*f1*t/r-G12*(t^2)*(-t2*st)/(4*(r^2)*R*zeta)
+3*G12*(t^2)*(-t2*st)/(8*(r^2)*R*(zeta^2))
-G12*(t^3)*st/(6*(R^2)*r*(zeta^2))
+f1*nu21*t*ct/(R*zeta*nu12)+G12*(t^3)*st2/(12*(R^3)*(zeta^3))
+7*G12*(t^3)*st/(24*(R^2)*r*(zeta^3)))*(-k)
+(-f1*nu21*(t^3)*ct/(12*R*(r^2)*nu12*(zeta^3))))*(k^3))*r/t)/((r^2)*zeta);
ani(j,j)=((-f1*t*nu21*gammar*st/((r^2)*zeta)
-f1*t*(gammar^2)*st*ct*nu21/((r^2)*(zeta^2)*nu12)
+f1*nu21*t*((t2*st)^2)*st/(2*(r^3)*R*zeta)
+nu21*f1*(t^2)*(-t2*ct)*st/(4*(r^3)*R*zeta)
+nu21*f1*(-t2*st)*(t^2)*ct/((r^3)*R*zeta*2)
-f1*(t^3)*nu21*st/(12*(r^3)*R*zeta)
+f1*(t^2)*(-t2*st)*gammar*(st^2)*nu21/(4*(r^3)*(zeta^2)*R*nu12)
+f1*(t^3)*gammar*st*ct*nu21/(6*(r^3)*(zeta^2)*R*nu12)
+f1*(t^3)*nu21*(gammar^2)*(st^3)/(nu12*(r^3)*R*(zeta^3)))*(-(r^2)*zeta)/t
-((-G12*(t^2)*(-t2*st)/(4*(r^4)*(zeta^2))
-G12*(t^2)*(-t2*st)/(8*(r^4)*(zeta^3))
-G12*(t^3)*st/(24*(r^3)*(zeta^4)*R)
+f1*nu21*(t^3)*st/(12*nu12*(r^2)*R*(zeta^3)))*(k^2))
*(-(r^2)*zeta)/t)/((r^2)*zeta);
anm(j,j)=(((nu21*f1*t/((r^2)*zeta)
-G12*(t^2)*st2*(-t2*st)/(4*(r^2)*(R^2)*(zeta^3))
+G12*(t^2)*st*(-t2*st)/(4*(r^3)*R*(zeta^2))
-G12*(t^2)*st*(-t2*st)/(8*(r^3)*R*(zeta^3))
-G12*(t^3)*ct2/(6*(r^2)*(R^2)*(zeta^3))
+G12*(t^3)*ct/(12*(r^3)*R*(zeta^2))
-G12*(t^3)*ct/(24*(r^3)*R*(zeta^3))
-G12*(t^3)*st*st2/(12*r*(R^3)*(zeta^4))
-G12*(t^3)*(st^2)/(24*(r^2)*(R^2)*(zeta^4))
-nu21*f1*t*ct*((t2*st)^2)/(2*(r^3)*R*(zeta^2))
+nu21*f1*(t^2)*st*(-t2*st)/(2*(r^3)*R*(zeta^2))
-nu21*f1*(t^2)*st2*(-t2*st)/(4*(r^2)*(R^2)*(zeta^3))
-nu21*f1*(t^2)*ct*(-t2*ct)/(4*(r^3)*R*(zeta^2))
+nu21*f1*(t^3)*ct/(12*(r^3)*R*(zeta^2))
-nu21*f1*(t^3)*ct*(st^2)/(6*r*(R^3)*(zeta^4))
```

```
-(ct^2)*nu21*f1*(t^3)/(12*(r^2)*(R^2)*(zeta^3))
-nu21*f1*(t^3)*(st^2)*ct/(6*r*(R^3)*(zeta^4)*nu12)
-nu21*f1*(t^2)*st2*(-t2*st)/(8*nu12*(r^2)*(R^2)*(zeta^3))
+nu21*f1*(t^3)*(st^2)/(6*(r^2)*(R^2)*(zeta^3))
+nu21*f1*ct*t/(nu12*r*R*(zeta^2))
-nu21*f1*(t^3)*ct2/(12*nu12*(r^2)*(R^2)*(zeta^3)))*k
-(-nu21*f1*(t^3)*ct/(12*nu12*(r^3)*R*(zeta^4)))*(k^3))
*(-(r^2)*zeta)/t)/((r^2)*zeta);
ann(j,j)=((f1*t*(ct^2)*nu21/((R^2)*(zeta^2)*nu12)
+2*nu21*f1*t*ct/(r*R*zeta)+f1*t/(r^2))*(-(r^2)*zeta)/t
+(-(G12*(t^3)*st/(6*(r^2)*(R^2)*(zeta^4))
+G12*(t^2)*(-t2*st)/(2*(r^3)*(zeta^3)*R)
+nu21*f1*t*((t2*st)^2)/(2*(r^4)*(zeta^2))
+nu21*f1*(t^2)*st*(-t2*st)/(2*(r^3)*(zeta^3)*R)
+nu21*f1*(t^2)*(-t2*ct)/(4*(r^4)*(zeta^2))
+nu21*f1*(t^3)*(st^2)/(6*(r^2)*(R^2)*(zeta^4))
+nu21*f1*(t^3)*ct/(12*(r^3)*(zeta^3)*R)
+f1*nu21*(t^3)*(st^2)/(6*nu12*(r^2)*(R^2)*(zeta^4))
+f1*nu21*(t^2)*(-t2*st)*st/(4*nu12*(r^3)*R*(zeta^3))
+nu21*f1*(t^3)*ct/(12*nu12*(r^3)*R*(zeta^3)))*(k^2)
+(nu21*f1*(t^3)/(12*nu12*(r^4)*(zeta^4)))*(k^4))
*(-(r^2)*zeta)/t)/((r^2)*zeta);
%
% terms which involve the phi derivatives
for jj=1:n
aii(j,jj)=aii(j,jj)+(((f1*(-t2*st)*zeta/r-f1*t*gammar*st/r
+f1*(t^2)*(-t2*st)*zeta/(4*(r^3))-f1*(t^3)*gammar*st/(12*(r^3)))*b1(j,jj)
+(f1*t*zeta/r+f1*(t^3)*zeta/(12*(r^3)))*b2(j,jj))*r/t)/((r^2)*zeta);
aim(j,jj)=aim(j,jj)+(((nu21*f1*(t^3)*ct/(12*zeta*(r^2)*R)+G12*t/r
+G12*(t^3)/(24*(r^3))-G12*(t^3)/(24*(r^3)*zeta)
-G12*(t^3)/(32*(r^3)*(zeta^2))
+nu21*f1*t/r)*k*b1(j,jj))*r/t)/((r^2)*zeta);
ain(j,jj)=ain(j,jj)+(((f1*nu21*t*gammar*ct/r+f1*t*zeta/r
+nu21*f1*(t^2)*(-t2*st)*st/(4*(r^2)*R)+nu21*f1*(t^3)*ct/(12*(r^2)*R)
+f1*(t^3)*nu21*gammar*(st^2)/(12*nu12*(r^2)*R*zeta))*b1(j,jj)
+(-f1*(t^2)*(-t2*st)*zeta/(4*(r^3))
+f1*(t^3)*gammar*st/(12*(r^3)))*b2(j,jj)
+(-f1*(t^3)*zeta/(12*(r^3)))*b3(j,jj))*r/t
-((-nu21*f1*(t^3)/(12*(r^3)*zeta)
-G12*(t^3)/(12*(r^3)*zeta)
-G12*(t^3)/(24*(r^3)*(zeta^2)))*(k^2)*b1(j,jj))*r/t)/((r^2)*zeta);
ami(j,jj)=ami(j,jj)-(((nu21*f1*(t^3)*ct/(12*(r^2)*R*zeta)
+nu21*f1*t/r+G12*(t^3)/(24*(r^3))-G12*(t^3)/(24*(r^3)*zeta)
-G12*(t^3)/(32*(r^3)*(zeta^2))+G12*t/r)*k*b1(j,jj))*r/t)/((r^2)*zeta);
amm(j,jj)=amm(j,jj)+(((-st*G12*t/R+G12*(t^2)*zeta*(-t2*st)/(8*(r^3))
-3*G12*(t^2)*(-t2*st)/(8*(r^3))+9*G12*(t^2)*(-t2*st)/(32*(r^3)*zeta)
+3*G12*(t^3)*st/(32*(r^2)*R*(zeta^2))-st*G12*(t^3)/(24*(r^2)*R)
+G12*(-t2*st)*zeta/r)*b1(j,jj)+(G12*t*zeta/r
+G12*(t^3)*zeta/(24*(r^3))-G12*(t^3)/(8*(r^3))
+3*G12*(t^3)/(32*(r^3)*zeta))*b2(j,jj))*r/t)/((r^2)*zeta);
amn(j,jj)=amn(j,jj)-(((st*f1*nu21*(t^3)*ct/(12*nu12*(R^2)*r*(zeta^2))
-G12*(t^2)*(-t2*st)/(4*(r^3))+3*G12*(t^2)*(-t2*st)/(8*(r^3)*zeta)
-G12*(t^3)*st/(12*(r^2)*R*zeta)-G12*(t^3)/(12*(r^2)*R*zeta)
+G12*(t^3)/(8*(r^2)*R*(zeta^2))+G12*(t^3)*st2/(12*r*(R^2)*(zeta^2))
+G12*(t^3)*st/(6*(r^2)*(zeta^2)*R))*k*b1(j,jj))*r/t
-((-nu21*f1*(t^3)*ct/(12*(r^2)*R*zeta)
-G12*(t^3)*ct/(12*(r^2)*R*zeta)
```

```
+G12*(t^3)/(24*(r^3)*zeta))*k*b2(j,jj))*r/t)/((r^2)*zeta);
ani(j,jj)=ani(j,jj)+(((f1*t/(r^2)+ct*nu21*f1*t/(R*r*zeta)
+nu21*f1*(t^3)*ct/(12*(r^3)*R*zeta)
+nu21*f1*(t^2)*st*(-t2*st)/(4*(r^3)*R*zeta)
-f1*(t^2)*(-t2*ct)/(4*(r^4))
-f1*t*((t2*st)^2)/(2*(r^4))
+f1*(t^3)*(st^2)*nu21/(12*(r^2)*(R^2)*(zeta^2)*nu12)
+f1*(t^2)*(-t2*st)*st/(2*(r^3)*R*zeta)
+f1*(t^3)*ct/(12*(r^3)*R*zeta))*b1(j,jj)
+(nu21*f1*(t^3)*st/(12*(r^3)*R*zeta)
-f1*(t^2)*(-t2*st)/(2*(r^4))
-nu21*f1*(t^3)*st/(12*(r^3)*R*zeta)
+f1*(t^3)*st/(6*(r^3)*R*zeta))*b2(j,jj)
+(-f1*(t^3)/(12*(r^4)))*b3(j,jj))*(-(r^2)*zeta)/t
+((-G12*(t^3)/(12*(zeta^2)*(r^4))
-G12*(t^3)/(24*(zeta^3)*(r^4))
-nu21*f1*(t^3)/(12*(r^4)*(zeta^2)))*(k^2)*b1(j,jj))
*((r^2)*zeta)/t)/((r^2)*zeta);
anm(j,jj)=anm(j,jj)+(((G12*(t^3)*st/(6*(r^3)*R*(zeta^2))
-G12*(t^3)*st/(24*(r^3)*R*(zeta^3))-G12*(t^2)*(-t2*st)/(4*(r^4)*zeta)
+3*G12*(t^2)*(-t2*st)/(8*(r^4)*(zeta^2))
-G12*(t^3)*st2/(12*(r^2)*(R^2)*(zeta^3))
-nu21*f1*(t^2)*ct*(-t2*st)/(2*(r^3)*R*(zeta^2))
+nu21*f1*(t^3)*st/(6*(r^3)*R*(zeta^2))
-nu21*f1*(t^3)*st2/(24*nu12*(r^2)*(R^2)*(zeta^3))
-nu21*f1*(t^3)*st2/(12*(r^2)*(R^2)*(zeta^3)))*k*b1(j,jj)
+(-G12*(t^3)/(12*(r^4)*zeta)
+G12*(t^3)/(8*(r^4)*(zeta^2))
-nu21*f1*(t^3)*ct/(12*(r^3)*R*(zeta^2)))*k*b2(j,jj))
*(-(r^2)*zeta)/t)/((r^2)*zeta);
ann(j,jj)=ann(j,jj)+(((nu21*f1*(t^3)*st/(12*(r^3)*R*zeta)
-nu21*f1*t*st*((t2*st)^2)/(2*(r^3)*R*zeta)-nu21*f1*(t^2)*ct
*(-t2*st)/(2*(r^3)*R*zeta)
-nu21*f1*(t^2)*st*(-t2*ct)/(4*(r^3)*R*zeta)
-nu21*f1*(t^3)*(st^3)/(12*nu12*r*(R^3)*(zeta^3))
-f1*(t^2)*(st^2)*(-t2*st)*nu21/(4*(r^2)*(R^2)*(zeta^2)*nu12)
-f1*(t^3)*st*ct*nu21/(6*(r^2)*(R^2)*(zeta^2)*nu12))*b1(j,jj)
+(-nu21*f1*(t^3)*ct/(12*(r^3)*R*zeta)
-nu21*f1*(t^2)*st*(-t2*st)/(4*(r^3)*R*zeta)
+(t^2)*(-t2*ct)*f1/(4*(r^4))+f1*((t2*st)^2)*t/(2*(r^4))
-f1*(t^3)*(st^2)*nu21/(nu12*12*(r^2)*(R^2)*(zeta^2))
-f1*(t^2)*(-t2*st)*st/(2*(r^3)*R*zeta)
-f1*(t^3)*ct/(12*(r^3)*R*zeta))*b2(j,jj)+(f1*(t^2)*(-t2*st)/(2*(r^4))
-f1*(t^3)*st/(6*(r^3)*R*zeta))*b3(j,jj)+(f1*(t^3)/(12*(r^4)))*b4(j,jj))
*(-(r^2)*zeta)/t+((nu21*f1*(t^2)*(-t2*st)/(2*(r^4)*(zeta^2))
+nu21*f1*(t^3)*st/(6*(r^3)*(zeta^3)*R)
+G12*(t^3)*(-t2*st)/(2*(r^4)*(zeta^2))
+G12*(t^3)/(6*(r^3)*R*(zeta^3)))*(k^2)*b1(j,jj)
+(nu21*f1*(t^3)/(6*(r^4)*(zeta^2))
+G12*(t^3)/(6*(r^4)*(zeta^2)))*(k^2)*b2(j,jj))
*((r^2)*zeta)/t)/((r^2)*zeta);
end
end
%---------------------------------------------------------------------------------
% section bb-bd - enforce the fixed boundary conditions
bb=zeros(8);bd=zeros(8,3*n-8);
%
```

```
for i=1:8; bb(i,i)=1; end
% eqns 6 7 for slope condition
bb(6,5)=b1(2,1);      bb(7,5)=b1(n-1,1);
bb(6,6)=b1(2,2);      bb(7,6)=b1(n-1,2);
bb(6,7)=b1(2,n-1);    bb(7,7)=b1(n-1,n-1);
bb(6,8)=b1(2,n);      bb(7,8)=b1(n-1,n);
%
for i=1:n-4;
bd(6,2*n-4+i)=b1(2,i+2);
bd(7,2*n-4+i)=b1(n-1,i+2);
end
%-----------------------------------------------------------------------------
% section db-dd - enforce the governing equations in the domain
dd=zeros(3*n-8);db=zeros(3*n-8,8);
%
for i=1:n-2;
db(i,1)=aii(i+1,1);        db(i,5)=ain(i+1,1);
db(i,2)=aii(i+1,n);        db(i,6)=ain(i+1,2);
db(i,3)=aim(i+1,1);        db(i,7)=ain(i+1,n-1);
db(i,4)=aim(i+1,n);        db(i,8)=ain(i+1,n);
%
db(i+n-2,1)=ami(i+1,1);    db(i+n-2,5)=amn(i+1,1);
db(i+n-2,2)=ami(i+1,n);    db(i+n-2,6)=amn(i+1,2);
db(i+n-2,3)=amm(i+1,1);    db(i+n-2,7)=amn(i+1,n-1);
db(i+n-2,4)=amm(i+1,n);    db(i+n-2,8)=amn(i+1,n);
end
%
for i=1:n-4;
db(i+2*(n-2),1)=ani(i+2,1);      db(i+2*(n-2),5)=ann(i+2,1);
db(i+2*(n-2),2)=ani(i+2,n);      db(i+2*(n-2),6)=ann(i+2,2);
db(i+2*(n-2),3)=anm(i+2,1);      db(i+2*(n-2),7)=ann(i+2,n-1);
db(i+2*(n-2),4)=anm(i+2,n);      db(i+2*(n-2),8)=ann(i+2,n);
end
%
for i=1:n-2; for j=1:n-2
dd(i,j)          =aii(1+i,1+j);     dd(n-2+i,j)     =ami(1+i,1+j);
dd(i,n-2+j)      =aim(1+i,1+j);     dd(n-2+i,n-2+j)=amm(1+i,1+j);
end;end
%
for i=1:n-2; for j=1:n-4
dd(i,2*(n-2)+j)      =ain(i+1,j+2);  dd(2*(n-2)+j,i)      =ani(j+2,i+1);
dd(n-2+i,2*(n-2)+j)=amn(i+1,j+2);  dd(2*(n-2)+j,n-2+i)=anm(j+2,i+1);
end;end
%
for i=1:n-4; for j=1:n-4
dd(2*(n-2)+i,2*(n-2)+j)=ann(2+i,2+j);
end;end
%-----------------------------------------------------------------------------
% section eign
prod1=-inv(bb)*bd; clear bb bd;
prod2=db*prod1; clear db prod1;
stiff=-(dd+prod2); clear dd prod2;
% find eignvalues
[evc, evl]=eig(stiff);  clear stiff;
tmn=3*n-8;
kk=0;
for i=1:tmn
```

```
if imag(evl(i,i)) == 0
if evl(i,i) > 0; kk=kk+1; ezz(kk)=evl(i,i); end
end
end
ezzz=ezz(1:kk);
% sort the eigenvalues and output the ten lowest
sval=sort(ezzz);
for j=1:10; result(j)=sqrt(sval(j)/den); end
% ---------------------------------------------------------------------------
% section mode - plot vibration mode shape
% F is a scaling factor
del(:,1)=evc(:,131); clear evc;
F=0.3;
%
% displacement components at the sampling points, u, v, w
%
% copy in del values
uy=zeros(m+1,n); vy=zeros(m+1,n); wy=zeros(m+1,n);
for i=1:m; teta(i)=2*pi*(i-1)/m; end
teta(m+1)=.0;
for ar=1:n-2
for j=1:m+1
uy(j,ar+1)=del(ar,1)*cos(0*teta(j));
vy(j,ar+1)=del(ar+(n-2),1)*sin(0*teta(j));
end; end
for br=1:n-4
for j=1:m+1
wy(j,br+2)=del(2*(n-2)+br,1)*cos(0*teta(j));
end; end
%
% determine the coordinates of undisplaced points
for j=1:m+1
thetang=teta(j); sp=sin(thetang); cp=cos(thetang);
for i=1:n
phiang=x(i); cs=cos(phiang); sn=sin(phiang);
xx(j,i)=(R+r*cs)*sp;
yy(j,i)=(R+r*cs)*cp;
zz(j,i)=r*sn;
end; end
%
% find coord of displaced points
for j=1:m+1
thetang=teta(j); sp=sin(thetang); cp=cos(thetang);
for i=1:n
phiang=x(i); cs=cos(phiang); sn=sin(phiang);
csn=cos(phiang+pi/2); snn=sin(phiang+pi/2);
xdisp(j,i)=xx(j,i)-F*(sp*(-uy(j,i)*csn+wy(j,i)*cs)+vy(j,i)*cp);
ydisp(j,i)=yy(j,i)-F*(cp*(-uy(j,i)*csn+wy(j,i)*cs)-vy(j,i)*sp);
zdisp(j,i)=zz(j,i)-F*(uy(j,i)*snn+wy(j,i)*sn);
end; end
% plot the mode shape
hold on
surf(xdisp,ydisp,zdisp);axis('on');
view(-60,60)
hold off
```

APPENDIX D - Listing of program Afreqc.m

```
% AFreqc.m  last modification - Feb. 25, 2002
% Sanders shell theory using 2 gov eqns comp & struct 43 (1992) 1019-1028.
% axisymmetric vibration of orhtropic toroidal shell using DQM
%----------------------------------------------------------------------------
% section input
% n=sampling points in meridional direction;
% m=sampling points in circumferential direction;
% E1=Young's modulus in meridional direction (Pa);
% G12=shear moduli (Pa);
% nu12=Poisson's ratio; nu21=Poisson's ratio;
% den=mass density (kg/m3);
% r=x-section radius (m); R=bend radius (m);
% t1 t2=thickness coefficients;
% format long
n=48
m=60
r=0.12; R=0.38;  E1=881570000; nu12=0.551; nu21=0.00525
f1=E1/(1-nu21*nu12); den=1190;
gammar=r/R; t1=0.008; t2=0.0008; G12=7850000
xlo=-5/6*pi; xhi=5/6*pi; xdif=xhi-xlo;
%----------------------------------------------------------------------------
% section wccd - find the weighting coeffts for meridional direction
% unequal spacing with one delta point at each end for phi direction
% output - b1(i,j),b2(i,j),b3(i,j),b4(i,j) for 1st to 4th derivats
y(1)=0; y(2)=.00001; y(n-1)=.99999;y(n)=1;
%
for i=3:n-2
tt=(pi*(i-2))/(n-3);y(i)=(1-cos(tt))/2;
end
%
for i=1:n; x(i)=xlo+y(i)*xdif; end
%
for i=1:n; pai(i)=1; for j=1:n
if i~=j; pai(i)=(x(i)-x(j))*pai(i); end
end;end
%
for i=1:n; for j=1:n
if i~=j; b1(i,j)=pai(i)/((x(i)-x(j))*pai(j)); end
end;end
%
for i=1:n; b1(i,i)=0; for j=1:n
if i~=j; b1(i,i)=b1(i,i)-b1(i,j); end
end;end
%
for i=1:n; for j=1:n
if i~=j; b2(i,j)=2*(b1(i,i)*b1(i,j)-b1(i,j)/(x(i)-x(j))); end
end;end
%
for i=1:n; b2(i,i)=0; for j=1:n
if i~=j; b2(i,i)=b2(i,i)-b2(i,j); end
end;end
%
for i=1:n; for j=1:n
if i~=j; b3(i,j)=3*(b2(i,i)*b1(i,j)-b2(i,j)/(x(i)-x(j))); end
end;end
```

```
%
for i=1:n; b3(i,i)=0; for j=1:n
if i~=j; b3(i,i)=b3(i,i)-b3(i,j); end
end;end
%
for i=1:n; for j=1:n
if i~=j; b4(i,j)=4*(b3(i,i)*b1(i,j)-b3(i,j)/(x(i)-x(j))); end
end;end
%
for i=1:n; b4(i,i)=0; for j=1:n
if i~=j; b4(i,i)=b4(i,i)-b4(i,j); end
end;end
clear i; clear j;
%-----------------------------------------------------------------------------
% section domeqn - set up matrices enforcing the governing equations
%    aii    ain     u       0
%
%    ani    ann     w       0
% output - aii(i,j),...,ann(i,j) - coefft matrices for displacements
%
aii(:,:)=zeros(n); ain(:,:)=zeros(n);
ani(:,:)=zeros(n); ann(:,:)=zeros(n);
%
%loop over sampling points in the phi direction
for j=1:n
tee=x(j); ct=cos(tee); st=sin(tee);
zeta=1+gammar*ct; t=t1+t2*ct;
% terms which do not involve derivatives
aii(j,j)=(-nu21*f1*(-t2*st)*gammar*st/r-nu21*f1*t*gammar*ct/r
-(gammar^2)*(st^2)*f1*t*nu21/(r*zeta*nu12)
-nu21*f1*(t^2)*(-t2*st)*st/(4*(r^2)*R)-f1*nu21*(t^3)*ct/(12*(r^2)*R)
-f1*(t^3)*gammar*nu21*(st^2)/(12*nu12*(r^2)*R*zeta))*r/((r^2)*zeta*t);
ain(j,j)=(nu21*f1*(-t2*st)*gammar*ct/r
+f1*(-t2*st)*zeta/r-f1*t*gammar*st/r
+f1*t*(gammar^2)*st*ct*nu21/(r*zeta*nu12))*r/((r^2)*zeta*t);
ani(j,j)=(-f1*t*nu21*gammar*st/((r^2)*zeta)
-f1*t*(gammar^2)*st*ct*nu21/((r^2)*(zeta^2)*nu12)
+f1*nu21*t*((t2*st)^2)*st/(2*(r^3)*R*zeta)
+nu21*f1*(t^2)*(-t2*ct)*st/(4*(r^3)*R*zeta)
+nu21*f1*(-t2*st)*(t^2)*ct/((r^3)*R*zeta*2)
-f1*(t^3)*nu21*st/(12*(r^3)*R*zeta)
+f1*(t^2)*(-t2*st)*gammar*(st^2)*nu21/(4*(r^3)*(zeta^2)*R*nu12)
+f1*(t^3)*gammar*st*ct*nu21/(6*(r^3)*(zeta^2)*R*nu12)
+f1*(t^3)*nu21*(gammar^2)*(st^3)/(12*nu12*(r^3)*R*(zeta^3)))
*(-(r^2)*zeta)/((r^2)*zeta*t);
ann(j,j)=(f1*t*(ct^2)*nu21/((R^2)*(zeta^2)*nu12)
+2*nu21*f1*t*ct/(r*R*zeta)+f1*t/(r^2))*(-(r^2)*zeta)/((r^2)*zeta*t);
%
% terms which involve the phi derivatives
for jj=1:n
aii(j,jj)=aii(j,jj)+((f1*(-t2*st)*zeta/r-f1*t*gammar*st/r
+f1*(t^2)*(-t2*st)*zeta/(4*(r^3))-f1*(t^3)*gammar*st/(12*(r^3)))*b1(j,jj)
+(f1*t*zeta/r+f1*(t^3)*zeta/(12*(r^3)))*b2(j,jj))*r/((r^2)*zeta*t);
ain(j,jj)=ain(j,jj)+((f1*nu21*t*gammar*ct/r+f1*t*zeta/r
+nu21*f1*(t^2)*(-t2*st)*st/(4*(r^2)*R)+nu21*f1*(t^3)*ct/(12*(r^2)*R)
+f1*(t^3)*nu21*gammar*(st^2)/(12*nu12*(r^2)*R*zeta))*b1(j,jj)
+(-f1*(t^2)*(-t2*st)*zeta/(4*(r^3))+f1*(t^3)*gammar*st/(12*(r^3)))*b2(j,jj)
```

```
+(-f1*(t^3)*zeta/(12*(r^3)))*b3(j,jj))*r/((r^2)*zeta*t);
ani(j,jj)=ani(j,jj)+((f1*t/(r^2)+ct*nu21*f1*t/(R*r*zeta)
+nu21*f1*(t^3)*ct/(12*(r^3)*R*zeta)+nu21*f1*(t^2)*st*(-t2*st)/(4*(r^3)*R*zeta)
-f1*(t^2)*(-t2*ct)/(4*(r^4))-f1*t*((t2*st)^2)/(2*(r^4))
+f1*(t^3)*(st^2)*nu21/(12*(r^2)*(R^2)*(zeta^2)*nu12)
+f1*(t^2)*(-t2*st)*st/(2*(r^3)*R*zeta)+f1*(t^3)*ct/(12*(r^3)*R*zeta))*b1(j,jj)
+(-f1*(t^2)*(-t2*st)/(2*(r^4))+f1*(t^3)*st/(6*(r^3)*R*zeta))*b2(j,jj)
+(-f1*(t^3)/(12*(r^4)))*b3(j,jj))*(-(r^2)*zeta)/((r^2)*zeta*t);
ann(j,jj)=ann(j,jj)+((nu21*f1*(t^3)*st/(12*(r^3)*R*zeta)
-nu21*f1*t*st*((t2*st)^2)/(2*(r^3)*R*zeta)
-nu21*f1*(t^2)*ct*(-t2*st)/(2*(r^3)*R*zeta)
-nu21*f1*(t^2)*st*(-t2*ct)/(4*(r^3)*R*zeta)
-nu21*f1*(t^3)*(st^3)/(12*nu12*r*(R^3)*(zeta^3))
-f1*(t^2)*(st^2)*(-t2*st)*nu21/(4*(r^2)*(R^2)*(zeta^2)*nu12)
-f1*(t^3)*st*ct*nu21/(6*(r^2)*(R^2)*(zeta^2)*nu12))*b1(j,jj)
+(-nu21*f1*(t^3)*ct/(12*(r^3)*R*zeta)
-nu21*f1*(t^2)*st*(-t2*st)/(4*(r^3)*R*zeta)
+(t^2)*(-t2*ct)*f1/(4*(r^4))+f1*((t2*st)^2)*t/(2*(r^4))
-f1*(t^3)*(st^2)*nu21/(nu12*12*(r^2)*(R^2)*(zeta^2))
-f1*(t^2)*(-t2*st)*st/(2*(r^3)*R*zeta)-f1*(t^3)*ct/(12*(r^3)*R*zeta))*b2(j,jj)
+(f1*(t^2)*(-t2*st)/(2*(r^4))-f1*(t^3)*st/(6*(r^3)*R*zeta))*b3(j,jj)
+(f1*(t^3)/(12*(r^4)))*b4(j,jj))*(-(r^2)*zeta)/((r^2)*zeta*t);
end
end
clear j; clear jj;
%-------------------------------------------------------------------------
% section bb-bd - enforce the fixed boundary conditions on two end points
bb=zeros(6);bd=zeros(6,2*n-6);
%
for i=1:6; bb(i,i)=1; end
clear i;
% eqns 4 5 for slope condition
bb(4,3)=b1(2,1);      bb(5,3)=b1(n-1,1);
bb(4,4)=b1(2,2);      bb(5,4)=b1(n-1,2);
bb(4,5)=b1(2,n-1);    bb(5,5)=b1(n-1,n-1);
bb(4,6)=b1(2,n);      bb(5,6)=b1(n-1,n);
%
for i=1:n-4;
bd(4,n-2+i)=b1(2,i+2);
bd(5,n-2+i)=b1(n-1,i+2);
end
clear i;
%-------------------------------------------------------------------------
-
% section db-dd - enforce the governing equations in the domain
dd=zeros(2*n-6);db=zeros(2*n-6,6);
%
for i=1:n-2;
db(i,1)=aii(i+1,1);        db(i,3)=ain(i+1,1);
db(i,2)=aii(i+1,n);        db(i,4)=ain(i+1,2);
                           db(i,5)=ain(i+1,n-1);
                           db(i,6)=ain(i+1,n);
end
clear i;
%
for i=1:n-4;
db(i+(n-2),1)=ani(i+2,1);     db(i+(n-2),3)=ann(i+2,1);
```

```
db(i+(n-2),2)=ani(i+2,n);      db(i+(n-2),4)=ann(i+2,2);
                               db(i+(n-2),5)=ann(i+2,n-1);
                               db(i+(n-2),6)=ann(i+2,n);
end
clear i;
%
for i=1:n-2; for j=1:n-2
dd(i,j)=aii(1+i,1+j);
end;end
clear i; clear j;
%
for i=1:n-2; for j=1:n-4
dd(i,(n-2)+j)=ain(i+1,j+2);  dd((n-2)+j,i)=ani(j+2,i+1);
end;end
clear i; clear j;
%
for i=1:n-4; for j=1:n-4
dd((n-2)+i,(n-2)+j)=ann(2+i,2+j);
end;end
clear i; clear j;
prod1=-inv(bb)*bd; clear bb bd;
prod2=db*prod1; clear db prod1;
stiff=-(dd+prod2); clear dd prod2;
%-----------------------------------------------------------------------------
% section eign
% find eignvalues
[evc, evl]=eig(stiff);  clear stiff;
tmn=2*n-6;
kk=0;
for i=1:tmn
if imag(evl(i,i)) == 0
if evl(i,i) > 0; kk=kk+1; ezz(kk)=evl(i,i); gzz(:,kk)=evc(:,i); end
end
end
ezzz=ezz(1:kk);
% sort the eigenvalues and output the five lowest
sval=sort(ezzz);
for j=1:5; result(j)=sqrt(sval(j)/den); end
%-----------------------------------------------------------------------------
% section mode - plot vibration mode shape
% F is a scaling factor
result
del(:,1)=gzz(:,85); clear evc;
F=0.2;
%
% displacement components at the sampling points, u, v, w
%
% copy in del values
uy=zeros(m+1,n); vy=zeros(m+1,n); wy=zeros(m+1,n);
for i=1:m; teta(i)=2*pi*(i-1)/m; end
teta(m+1)=.0;
for ar=1:n-2
for j=1:m+1
uy(j,ar+1)=del(ar,1)*cos(0*teta(j));
end; end
for br=1:n-4
for j=1:m+1
```

```
wy(j,br+2)=del((n-2)+br,1)*cos(0*teta(j));
end; end
%
% determine the coordinates of undisplaced points
for j=1:m+1
thetang=teta(j); sp=sin(thetang); cp=cos(thetang);
for i=1:n
phiang=x(i); cs=cos(phiang); sn=sin(phiang);
xx(j,i)=(R+r*cs)*sp;
yy(j,i)=(R+r*cs)*cp;
zz(j,i)=r*sn;
end; end
%
% find coord of displaced points
for j=1:m+1
thetang=teta(j); sp=sin(thetang); cp=cos(thetang);
for i=1:n
phiang=x(i); cs=cos(phiang); sn=sin(phiang);
csn=cos(phiang+pi/2); snn=sin(phiang+pi/2);
xdisp(j,i)=xx(j,i)-F*(sp*(-uy(j,i)*csn+wy(j,i)*cs)+vy(j,i)*cp);
ydisp(j,i)=yy(j,i)-F*(cp*(-uy(j,i)*csn+wy(j,i)*cs)-vy(j,i)*sp);
zdisp(j,i)=zz(j,i)-F*(uy(j,i)*snn+wy(j,i)*sn);
end; end
% plot the mode shape
hold on
surf(xdisp,ydisp,zdisp);axis('on');
view(-60,60)
hold off
```

APPENDIX D - Listing of program Orthsta.m

```
% Orthsta.m  last modification - March 10, 2002
% Sanders shell theory using 3 gov eqns comp & struct 43 (1992) 1019-1028.
% elastostatics of orhtropic toroidal shell - pad load using DQM
% Fourier series expansion written in circumferential direction
%-------------------------------------------------------------------------------
% section input
% n=sampling points in meridional direction;
% n1=truncation integer;
% E1=Young's modulus in meridional direction (Pa);
% p0=pad load pressure (Pa);
% nu12=Poisson's ratio; nu21=Poisson's ratio;
% den=mass density (kg/m3);
% r=x-section radius (m); R=bend radius (m);
% t1 t2=thickness coefficients;
% beta=load angle extended in meridional direction;
% format long
n=33
n1=10
r=0.12; R=0.38;  E1=881570000; nu12=0.551; nu21=0.00525; f1=E1/(1-nu21*nu12)
p0=1; beta=1/12*pi; G12=7850000
gammar=r/R; t1=0.008; t2=0.0008
xlo=-5/6*pi; xhi=5/6*pi; xdif=xhi-xlo; theta=0;
%-------------------------------------------------------------------------------
% section wcmd - find the weighting coeffts for meridional direction
% unequal spacing with one delta point at each end for phi direction
% output - b1(i,j),b2(i,j),b3(i,j),b4(i,j) for 1st to 4th derivats
y(1)=0; y(2)=.00001; y(n-1)=.99999;y(n)=1;
%
for i=3:n-2
tt=(pi*(i-2))/(n-3);y(i)=(1-cos(tt))/2;
end
%
for i=1:n; x(i)=xlo+y(i)*xdif; end
%
for i=1:n; pai(i)=1; for j=1:n
if i~=j; pai(i)=(x(i)-x(j))*pai(i); end
end;end
%
for i=1:n; for j=1:n
if i~=j; b1(i,j)=pai(i)/((x(i)-x(j))*pai(j)); end
end;end
%
for i=1:n; b1(i,i)=0; for j=1:n
if i~=j; b1(i,i)=b1(i,i)-b1(i,j); end
end;end
%
for i=1:n; for j=1:n
if i~=j; b2(i,j)=2*(b1(i,i)*b1(i,j)-b1(i,j)/(x(i)-x(j))); end
end;end
%
for i=1:n; b2(i,i)=0; for j=1:n
if i~=j; b2(i,i)=b2(i,i)-b2(i,j); end
end;end
%
for i=1:n; for j=1:n
```

```
if i~=j; b3(i,j)=3*(b2(i,i)*b1(i,j)-b2(i,j)/(x(i)-x(j))); end
end;end
%
for i=1:n; b3(i,i)=0; for j=1:n
if i~=j; b3(i,i)=b3(i,i)-b3(i,j); end
end;end
%
for i=1:n; for j=1:n
if i~=j; b4(i,j)=4*(b3(i,i)*b1(i,j)-b3(i,j)/(x(i)-x(j))); end
end;end
%
for i=1:n; b4(i,i)=0; for j=1:n
if i~=j; b4(i,i)=b4(i,i)-b4(i,j); end
end;end
%+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
% solution for axisymmetric part
%
% section domeqn - set up matrices enforcing the governing equations
%    aii    ain     u       0
%
%    ani    ann     w       0
% output - aii(i,j),...,ann(i,j) - coefft matrices for displacements
%
aii(:,:)=zeros(n); ain(:,:)=zeros(n);
ani(:,:)=zeros(n); ann(:,:)=zeros(n);
%
%loop over sampling points in the phi direction
for j=1:n
tee=x(j); ct=cos(tee); st=sin(tee); st2=2*st*ct; ct2=(ct^2)-(st^2);
zeta=1+gammar*ct; t=t1+t2*ct;
% terms which do not involve derivatives
aii(j,j)=-nu21*f1*(-t2*st)*gammar*st/r-nu21*f1*t*gammar*ct/r
-(gammar^2)*(st^2)*f1*t*nu21/(r*zeta*nu12)
-nu21*f1*(t^2)*(-t2*st)*st/(4*(r^2)*R)-f1*nu21*(t^3)*ct/(12*(r^2)*R)
-f1*(t^3)*gammar*nu21*(st^2)/(12*nu12*(r^2)*R*zeta);
ain(j,j)=nu21*f1*(-t2*st)*gammar*ct/r+f1*(-t2*st)*zeta/r
-f1*t*gammar*st/r+f1*t*(gammar^2)*st*ct*nu21/(r*zeta*nu12);
ani(j,j)=(-f1*t*nu21*gammar*st/((r^2)*zeta)
-f1*t*(gammar^2)*st*ct*nu21/((r^2)*(zeta^2)*nu12)
+f1*nu21*t*((t2*st)^2)*st/(2*(r^3)*R*zeta)
+nu21*f1*(t^2)*(-t2*ct)*st/(4*(r^3)*R*zeta)
+nu21*f1*(-t2*st)*(t^2)*ct/((r^3)*R*zeta*2)
-f1*(t^3)*nu21*st/(12*(r^3)*R*zeta)
+f1*(t^2)*(-t2*st)*gammar*(st^2)*nu21/(4*(r^3)*(zeta^2)*R*nu12)
+f1*(t^3)*gammar*st*ct*nu21/(6*(r^3)*(zeta^2)*R*nu12)
+f1*(t^3)*nu21*(gammar^2)*(st^3)/(nu12*(r^3)*R*(zeta^3)));
ann(j,j)=(f1*t*(ct^2)*nu21/((R^2)*(zeta^2)*nu12)
+2*nu21*f1*t*ct/(r*R*zeta)+f1*t/(r^2));
%
% terms which involve the phi derivatives
for jj=1:n
aii(j,jj)=aii(j,jj)+(f1*(-t2*st)*zeta/r-f1*t*gammar*st/r
+f1*(t^2)*(-t2*st)*zeta/(4*(r^3))-f1*(t^3)*gammar*st/(12*(r^3)))*b1(j,jj)
+(f1*t*zeta/r+f1*(t^3)*zeta/(12*(r^3)))*b2(j,jj);
ain(j,jj)=ain(j,jj)+(f1*nu21*t*gammar*ct/r+f1*t*zeta/r
+nu21*f1*(t^2)*(-t2*st)*st/(4*(r^2)*R)+nu21*f1*(t^3)*ct/(12*(r^2)*R)
+f1*(t^3)*nu21*gammar*(st^2)/(12*nu12*(r^2)*R*zeta))*b1(j,jj)
```

```
+(-f1*(t^2)*(-t2*st)*zeta/(4*(r^3))+f1*(t^3)*gammar*st/(12*(r^3)))*b2(j,jj)
+(-f1*(t^3)*zeta/(12*(r^3)))*b3(j,jj);
ani(j,jj)=ani(j,jj)+((f1*t/(r^2)+ct*nu21*f1*t/(R*r*zeta)
+nu21*f1*(t^3)*ct/(12*(r^3)*R*zeta)+nu21*f1*(t^2)*st*(-t2*st)/(4*(r^3)*R*zeta)
-f1*(t^2)*(-t2*ct)/(4*(r^4))-f1*t*((t2*st)^2)/(2*(r^4))
+f1*(t^3)*(st^2)*nu21/(12*(r^2)*(R^2)*(zeta^2)*nu12)
+f1*(t^2)*(-t2*st)*st/(2*(r^3)*R*zeta)+f1*(t^3)*ct/(12*(r^3)*R*zeta))*b1(j,jj)
+(nu21*f1*(t^3)*st/(12*(r^3)*R*zeta)-f1*(t^2)*(-t2*st)/(2*(r^4))
-nu21*f1*(t^3)*st/(12*(r^3)*R*zeta)+f1*(t^3)*st/(6*(r^3)*R*zeta))*b2(j,jj)
+(-f1*(t^3)/(12*(r^4)))*b3(j,jj));
ann(j,jj)=ann(j,jj)+((nu21*f1*(t^3)*st/(12*(r^3)*R*zeta)
-nu21*f1*t*st*((t2*st)^2)/(2*(r^3)*R*zeta)
-nu21*f1*(t^2)*ct*(-t2*st)/(2*(r^3)*R*zeta)
-nu21*f1*(t^2)*st*(-t2*ct)/(4*(r^3)*R*zeta)
-nu21*f1*(t^3)*(st^3)/(12*nu12*r*(R^3)*(zeta^3))
-f1*(t^2)*(st^2)*(-t2*st)*nu21/(4*(r^2)*(R^2)*(zeta^2)*nu12)
-f1*(t^3)*st*ct*nu21/(6*(r^2)*(R^2)*(zeta^2)*nu12))*b1(j,jj)
+(-nu21*f1*(t^3)*ct/(12*(r^3)*R*zeta)-nu21*f1*(t^2)*st*(-t2*st)/(4*(r^3)*R*zeta)
+(t^2)*(-t2*ct)*f1/(4*(r^4))+f1*((t2*st)^2)*t/(2*(r^4))
-f1*(t^3)*(st^2)*nu21/(nu12*12*(r^2)*(R^2)*(zeta^2))
-f1*(t^2)*(-t2*st)*st/(2*(r^3)*R*zeta)-f1*(t^3)*ct/(12*(r^3)*R*zeta))*b2(j,jj)
+(f1*(t^2)*(-t2*st)/(2*(r^4))-f1*(t^3)*st/(6*(r^3)*R*zeta))*b3(j,jj)
+(f1*(t^3)/(12*(r^4)))*b4(j,jj));
end
end
clear j; clear jj;
%--------------------------------------------------------------------------
% section bb-bd - enforce the fixed boundary conditions on two end points
bb=zeros(6);bd=zeros(6,2*n-6);
%
for i=1:6; bb(i,i)=1; end
clear i;
% eqns 6 7 for slope condition
bb(4,3)=b1(2,1);     bb(5,3)=b1(n-1,1);
bb(4,4)=b1(2,2);     bb(5,4)=b1(n-1,2);
bb(4,5)=b1(2,n-1);   bb(5,5)=b1(n-1,n-1);
bb(4,6)=b1(2,n);     bb(5,6)=b1(n-1,n);
%
for i=1:n-4;
bd(4,n-2+i)=b1(2,i+2);
bd(5,n-2+i)=b1(n-1,i+2);
end
clear i;
%--------------------------------------------------------------------------
% section db-dd - enforce the governing equations in the domain
dd=zeros(2*n-6);db=zeros(2*n-6,6);
%
for i=1:n-2;
db(i,1)=aii(i+1,1);        db(i,3)=ain(i+1,1);
db(i,2)=aii(i+1,n);        db(i,4)=ain(i+1,2);
                           db(i,5)=ain(i+1,n-1);
                           db(i,6)=ain(i+1,n);
end
clear i;
%
for i=1:n-4;
db(i+(n-2),1)=ani(i+2,1);      db(i+(n-2),3)=ann(i+2,1);
```

```
db(i+(n-2),2)=ani(i+2,n);      db(i+(n-2),4)=ann(i+2,2);
                               db(i+(n-2),5)=ann(i+2,n-1);
                               db(i+(n-2),6)=ann(i+2,n);
end
clear i;
%
for i=1:n-2; for j=1:n-2
dd(i,j)=aii(1+i,1+j);
end;end
clear i; clear j;
%
for i=1:n-2; for j=1:n-4
dd(i,(n-2)+j)=ain(i+1,j+2);  dd((n-2)+j,i)=ani(j+2,i+1);
end;end
clear i; clear j;
%
for i=1:n-4; for j=1:n-4
dd((n-2)+i,(n-2)+j)=ann(2+i,2+j);
end;end
clear i; clear j;
prod1=-inv(bb)*bd; clear bb bd;
prod2=db*prod1; clear db prod1;
stiff=dd+prod2; clear dd prod2;
%---------------------------------------------------------------------------
% section loadv - set up the load vector for the right hand side
rhss=zeros(2*n-6,1);
rhss(44,1)=-p0*beta/(2*pi);
rhss(45,1)=-p0*beta/pi;
rhss(46,1)=-p0*beta/pi;
rhss(47,1)=-p0*beta/pi;
rhss(48,1)=-p0*beta/(2*pi);
%---------------------------------------------------------------------------
% section solu - combine submat - solv matrix eqn - find results
% find the vector of displacement components rslt
clear aii;  clear ain; clear ani;  clear ann; clear pval; clear tval; clear
test1;
clear test2;
rslt=stiff\rhss; clear stiff;  clear rhss;
%
% displacement components at the sampling points, u,v
ut=zeros(n,1);  wt=zeros(n,1);  vt=zeros(n,1); vt1=zeros(n,1); wt2=zeros(n,1);
for i=1:n-2
ut(i+1,1)=rslt(i,1);
end;
for j=1:n-4
wt(j+2,1)=rslt(n-2+j,1);
end
clear rslt;
%++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
% solution for nonsymmetric part
%
for k1=1:n1
k=k1/(R/r);
%---------------------------------------------------------------------------
% section domeqn - set up matrices enforcing the governing equations
%    aii  aim  ain     u       0
%    ami  amm  amn     v   =   0
```

```
%    ani  anm  ann     w      0
% output - aii(i,j),...,ann(i,j) - coefft matrices for displacements
%
aii(:,:)=zeros(n);aim(:,:)=zeros(n);ain(:,:)=zeros(n);
ami(:,:)=zeros(n);amm(:,:)=zeros(n);amn(:,:)=zeros(n);
ani(:,:)=zeros(n);anm(:,:)=zeros(n);ann(:,:)=zeros(n);
%
%loop over sampling points in the phi direction
for j=1:n
tee=x(j); ct=cos(tee); st=sin(tee); st2=2*st*ct; ct2=(ct^2)-(st^2);
zeta=1+gammar*ct; t=t1+t2*ct;
% terms which do not involve derivatives
aii(j,j)=((-nu21*f1*(-t2*st)*gammar*st/r-nu21*f1*t*gammar*ct/r
-(gammar^2)*(st^2)*f1*t*nu21/(r*zeta*nu12)-nu21*f1*(t^2)*(-
t2*st)*st/(4*(r^2)*R)-f1*nu21*(t^3)*ct/(12*(r^2)*R)
-f1*(t^3)*gammar*nu21*(st^2)/(12*nu12*(r^2)*R*zeta))
-((G12*t/(r*zeta)+G12*(t^3)/(24*(r^3)*(zeta^2))+G12*(t^3)/(24*(r^3)*zeta)
+G12*(t^3)/(96*(r^3)*(zeta^3)))*(k^2)));
aim(j,j)=(((f1*nu21*(-t2*st)/r+st*nu21*f1*t/(R*zeta*nu12)
+nu21*f1*(t^2)*ct*(-t2*st)/(4*(r^2)*R*zeta)-nu21*f1*(t^3)*st/(12*(r^2)*R*zeta)
+G12*(t^3)*st2/(24*(R^2)*r*(zeta^2))+G12*t*st/(R*zeta)+G12*(t^3)*st2/(48*(R^2)*
r*(zeta^3))-G12*(t^3)*st/(24*R*(r^2)*zeta)+G12*(t^3)*st/(96*R*(r^2)*(zeta^3))
+st2*nu21*f1*(t^3)/(24*(R^2)*r*(zeta^2))+nu21*f1*(t^3)*st2/(24*(R^2)*r*nu12*(ze
ta^2)))*k));
ain(j,j)=((nu21*f1*(-t2*st)*gammar*ct/r+f1*(-t2*st)*zeta/r-f1*t*gammar*st/r
+f1*t*(gammar^2)*st*ct*nu21/(r*zeta*nu12))
-((-nu21*f1*(t^2)*(-t2*st)/(4*(r^3)*zeta)
-nu21*f1*(t^3)*st/(12*(r^2)*(zeta^2)*R)
-G12*(t^3)/(12*(r^2)*R*(zeta^2))-G12*(t^3)/(24*(r^2)*R*(zeta^3))
-st*f1*nu21*(t^3)/(12*nu12*R*(r^2)*(zeta^2)))*(k^2)));
ami(j,j)=(((-st*f1*nu21*t/(R*nu12*zeta)-G12*t*st/(R*zeta)
-f1*nu21*(st2)*(t^3)/(24*(R^2)*r*(zeta^2)*nu12)+G12*(-t2*st)/r
+G12*(t^2)*(-t2*st)/(8*(r^3))-G12*(t^2)*(-t2*st)/(8*(r^3)*zeta)
-3*G12*(t^2)*(-t2*st)/(32*(r^3)*(zeta^2))+G12*(t^3)*st/(24*(r^2)*R*zeta)
-7*G12*(t^3)*st/(96*(r^2)*R*(zeta^3))-st2*G12*(t^3)/(48*r*(R^2)*(zeta^3))
-st2*G12*(t^3)/(24*(r^2)*(R^2)*(zeta^2))
-st*G12*(t^3)/(24*(r^2)*R*(zeta^2)))*(-k)));
amm(j,j)=((-2*G12*t*r*(st^2)/((R^2)*zeta)+G12*ct*t/R
+G12*t*r*(st^2)/((R^2)*zeta)+G12*(t^2)*st2*(-t2*st)/(8*(R^2)*r*zeta)
-3*G12*(t^2)*st2*(-t2*st)/(16*(R^2)*r*(zeta^2))
-G12*(t^2)*st*(-t2*st)/(8*R*(r^2))+G12*(t^2)*st*(-t2*st)/(4*R*(r^2)*zeta)
-3*G12*(t^2)*st*(-t2*st)/(32*R*(r^2)*(zeta^2))+G12*(t^3)*ct2/(12*(R^2)*r*zeta)
-G12*(t^3)*ct2/(8*(R^2)*r*(zeta^2))-G12*(t^3)*ct/(24*(r^2)*R)
+G12*(t^3)*ct/(12*(r^2)*R*zeta)
-G12*(t^3)*ct/(32*(r^2)*R*(zeta^2))
-G12*(t^3)*(st^2)/(24*(R^2)*r*zeta)+G12*st*(-t2*st)/R
-r*(st2^2)*G12*(t^3)/(24*(R^4)*(zeta^3))+G12*(t^3)*(st^2)/(8*(R^2)*r*(zeta^2))
-7*G12*(t^3)*(st^2)/(96*(R^2)*r*(zeta^3))+G12*(t^3)*st*st2/(8*(R^3)*(zeta^2))-
G12*(t^3)*st*st2/(6*(R^3)*(zeta^3)))
-((ct^2)*nu21*f1*(t^3)/(12*(R^2)*r*(zeta^3)*nu12)
+f1*nu21*t/(r*zeta*nu12))*(k^2));
amn(j,j)=(((nu21*f1*t/r-G12*(t^2)*(-t2*st)/(4*(r^2)*R*zeta)
+3*G12*(t^2)*(-t2*st)/(8*(r^2)*R*(zeta^2))-G12*(t^3)*st/(6*(R^2)*r*(zeta^2))
+f1*nu21*t*ct/(R*zeta*nu12)+G12*(t^3)*st2/(12*(R^3)*(zeta^3))+7*G12*(t^3)*st/(2
4*(R^2)*r*(zeta^3)))*(-k)+(-
f1*nu21*(t^3)*ct/(12*R*(r^2)*nu12*(zeta^3)))*(k^3)));
ani(j,j)=((-f1*t*nu21*gammar*st/((r^2)*zeta)
```

```
-f1*t*(gammar^2)*st*ct*nu21/((r^2)*(zeta^2)*nu12)
+f1*nu21*t*((t2*st)^2)*st/(2*(r^3)*R*zeta)
+nu21*f1*(t^2)*(-t2*ct)*st/(4*(r^3)*R*zeta)
+nu21*f1*(-t2*st)*(t^2)*ct/((r^3)*R*zeta*2)
-f1*(t^3)*nu21*st/(12*(r^3)*R*zeta)
+f1*(t^2)*(-t2*st)*gammar*(st^2)*nu21/(4*(r^3)*(zeta^2)*R*nu12)
+f1*(t^3)*gammar*st*ct*nu21/(6*(r^3)*(zeta^2)*R*nu12)
+f1*(t^3)*nu21*(gammar^2)*(st^3)/(nu12*(r^3)*R*(zeta^3)))
-((-G12*(t^2)*(-t2*st)/(4*(r^4)*(zeta^2))-G12*(t^2)*(-t2*st)/(8*(r^4)*(zeta^3))
-G12*(t^3)*st/(24*(r^3)*(zeta^4)*R)
+f1*nu21*(t^3)*st/(12*nu12*(r^2)*R*(zeta^3)))*(k^2)));
anm(j,j)=(((nu21*f1*t/((r^2)*zeta)
-G12*(t^2)*st2*(-t2*st)/(4*(r^2)*(R^2)*(zeta^3))
+G12*(t^2)*st*(-t2*st)/(4*(r^3)*R*(zeta^2))-G12*(t^2)*st*(-
t2*st)/(8*(r^3)*R*(zeta^3))-G12*(t^3)*ct2/(6*(r^2)*(R^2)*(zeta^3))
+G12*(t^3)*ct/(12*(r^3)*R*(zeta^2))-G12*(t^3)*ct/(24*(r^3)*R*(zeta^3))
-G12*(t^3)*st*st2/(12*r*(R^3)*(zeta^4))
-G12*(t^3)*(st^2)/(24*(r^2)*(R^2)*(zeta^4))
-nu21*f1*t*ct*((t2*st)^2)/(2*(r^3)*R*(zeta^2))
+nu21*f1*(t^2)*st*(-t2*st)/(2*(r^3)*R*(zeta^2))
-nu21*f1*(t^2)*st2*(-t2*st)/(4*(r^2)*(R^2)*(zeta^3))
-nu21*f1*(t^2)*ct*(-t2*ct)/(4*(r^3)*R*(zeta^2))
+nu21*f1*(t^3)*ct/(12*(r^3)*R*(zeta^2))
-nu21*f1*(t^3)*ct*(st^2)/(6*r*(R^3)*(zeta^4))
-(ct^2)*nu21*f1*(t^3)/(12*(r^2)*(R^2)*(zeta^3))
-nu21*f1*(t^3)*(st^2)*ct/(6*r*(R^3)*(zeta^4)*nu12)
-nu21*f1*(t^2)*st2*(-t2*st)/(8*nu12*(r^2)*(R^2)*(zeta^3))
+nu21*f1*(t^3)*(st^2)/(6*(r^2)*(R^2)*(zeta^3))
+nu21*f1*ct*t/(nu12*r*R*(zeta^2))
-nu21*f1*(t^3)*ct2/(12*nu12*(r^2)*(R^2)*(zeta^3)))*k
-(-nu21*f1*(t^3)*ct/(12*nu12*(r^3)*R*(zeta^4)))*(k^3)));
ann(j,j)=((f1*t*(ct^2)*nu21/((R^2)*(zeta^2)*nu12)
+2*nu21*f1*t*ct/(r*R*zeta)+f1*t/(r^2))
+(-(G12*(t^3)*st/(6*(r^2)*(R^2)*(zeta^4))
+G12*(t^2)*(-t2*st)/(2*(r^3)*(zeta^3)*R)
+nu21*f1*t*((t2*st)^2)/(2*(r^4)*(zeta^2))
+nu21*f1*(t^2)*st*(-t2*st)/(2*(r^3)*(zeta^3)*R)
+nu21*f1*(t^2)*(-t2*ct)/(4*(r^4)*(zeta^2))
+nu21*f1*(t^3)*(st^2)/(6*(r^2)*(R^2)*(zeta^4))
+nu21*f1*(t^3)*ct/(12*(r^3)*(zeta^3)*R)+f1*nu21*(t^3)*(st^2)/(6*nu12*(r^2)*(R^2
)*(zeta^4))+f1*nu21*(t^2)*(-t2*st)*st/(4*nu12*(r^3)*R*(zeta^3))
+nu21*f1*(t^3)*ct/(12*nu12*(r^3)*R*(zeta^3)))*(k^2)+(nu21*f1*(t^3)/(12*nu12*(r^
4)*(zeta^4)))*(k^4)));
%
% terms which involve the phi derivatives
for jj=1:n
aii(j,jj)=aii(j,jj)+(((f1*(-t2*st)*zeta/r-f1*t*gammar*st/r
+f1*(t^2)*(-t2*st)*zeta/(4*(r^3))-f1*(t^3)*gammar*st/(12*(r^3)))*b1(j,jj)
+(f1*t*zeta/r+f1*(t^3)*zeta/(12*(r^3)))*b2(j,jj)));
aim(j,jj)=aim(j,jj)+(((nu21*f1*(t^3)*ct/(12*zeta*(r^2)*R)
+G12*t/r+G12*(t^3)/(24*(r^3))-G12*(t^3)/(24*(r^3)*zeta)
-G12*(t^3)/(32*(r^3)*(zeta^2))+nu21*f1*t/r)*k*b1(j,jj)));
ain(j,jj)=ain(j,jj)+(((f1*nu21*t*gammar*ct/r+f1*t*zeta/r
+nu21*f1*(t^2)*(-t2*st)*st/(4*(r^2)*R)+nu21*f1*(t^3)*ct/(12*(r^2)*R)
+f1*(t^3)*nu21*gammar*(st^2)/(12*nu12*(r^2)*R*zeta))*b1(j,jj)
+(-f1*(t^2)*(-t2*st)*zeta/(4*(r^3))+f1*(t^3)*gammar*st/(12*(r^3)))*b2(j,jj)
+(-f1*(t^3)*zeta/(12*(r^3)))*b3(j,jj))-((-nu21*f1*(t^3)/(12*(r^3)*zeta)
```

```
-G12*(t^3)/(12*(r^3)*zeta)-G12*(t^3)/(24*(r^3)*(zeta^2)))*(k^2)*b1(j,jj)));
ami(j,jj)=ami(j,jj)-(((nu21*f1*(t^3)*ct/(12*(r^2)*R*zeta)
+nu21*f1*t/r+G12*(t^3)/(24*(r^3))-G12*(t^3)/(24*(r^3)*zeta)
-G12*(t^3)/(32*(r^3)*(zeta^2))+G12*t/r)*k*b1(j,jj)));
amm(j,jj)=amm(j,jj)+(((-st*G12*t/R+G12*(t^2)*zeta*(-t2*st)/(8*(r^3))
-3*G12*(t^2)*(-t2*st)/(8*(r^3))+9*G12*(t^2)*(-t2*st)/(32*(r^3)*zeta)
+3*G12*(t^3)*st/(32*(r^2)*R*(zeta^2))-st*G12*(t^3)/(24*(r^2)*R)
+G12*(-t2*st)*zeta/r)*b1(j,jj)+(G12*t*zeta/r+G12*(t^3)*zeta/(24*(r^3))
-G12*(t^3)/(8*(r^3))+3*G12*(t^3)/(32*(r^3)*zeta))*b2(j,jj)));
amn(j,jj)=amn(j,jj)-(((st*f1*nu21*(t^3)*ct/(12*nu12*(R^2)*r*(zeta^2))
-G12*(t^2)*(-t2*st)/(4*(r^3))+3*G12*(t^2)*(-t2*st)/(8*(r^3)*zeta)
-G12*(t^3)*st/(12*(r^2)*R*zeta)-G12*(t^3)/(12*(r^2)*R*zeta)
+G12*(t^3)/(8*(r^2)*R*(zeta^2))+G12*(t^3)*st2/(12*r*(R^2)*(zeta^2))
+G12*(t^3)*st/(6*(r^2)*(zeta^2)*R))*k*b1(j,jj))-((-
nu21*f1*(t^3)*ct/(12*(r^2)*R*zeta)-G12*(t^3)*ct/(12*(r^2)*R*zeta)
+G12*(t^3)/(24*(r^3)*zeta))*k*b2(j,jj)));
ani(j,jj)=ani(j,jj)+(((f1*t/(r^2)+ct*nu21*f1*t/(R*r*zeta)
+nu21*f1*(t^3)*ct/(12*(r^3)*R*zeta)+nu21*f1*(t^2)*st*(-t2*st)/(4*(r^3)*R*zeta)
-f1*(t^2)*(-t2*ct)/(4*(r^4))-f1*t*((t2*st)^2)/(2*(r^4))
+f1*(t^3)*(st^2)*nu21/(12*(r^2)*(R^2)*(zeta^2)*nu12)
+f1*(t^2)*(-t2*st)*st/(2*(r^3)*R*zeta)+f1*(t^3)*ct/(12*(r^3)*R*zeta))*b1(j,jj)
+(nu21*f1*(t^3)*st/(12*(r^3)*R*zeta)-f1*(t^2)*(-t2*st)/(2*(r^4)
)-nu21*f1*(t^3)*st/(12*(r^3)*R*zeta)+f1*(t^3)*st/(6*(r^3)*R*zeta))*b2(j,jj)
+(-f1*(t^3)/(12*(r^4)))*b3(j,jj))+((-G12*(t^3)/(12*(zeta^2)*(r^4))
-G12*(t^3)/(24*(zeta^3)*(r^4))
-nu21*f1*(t^3)/(12*(r^4)*(zeta^2)))*(-(k^2))*b1(j,jj)));
anm(j,jj)=anm(j,jj)+(((G12*(t^3)*st/(6*(r^3)*R*(zeta^2))
-G12*(t^3)*st/(24*(r^3)*R*(zeta^3))
-G12*(t^2)*(-t2*st)/(4*(r^4)*zeta)+3*G12*(t^2)*(-t2*st)/(8*(r^4)*(zeta^2))
-G12*(t^3)*st2/(12*(r^2)*(R^2)*(zeta^3))-nu21*f1*(t^2)*ct*(-
t2*st)/(2*(r^3)*R*(zeta^2))+nu21*f1*(t^3)*st/(6*(r^3)*R*(zeta^2))
-nu21*f1*(t^3)*st2/(24*nu12*(r^2)*(R^2)*(zeta^3))
-nu21*f1*(t^3)*st2/(12*(r^2)*(R^2)*(zeta^3)))*k*b1(j,jj)
+(-G12*(t^3)/(12*(r^4)*zeta)+G12*(t^3)/(8*(r^4)*(zeta^2))
-nu21*f1*(t^3)*ct/(12*(r^3)*R*(zeta^2)))*k*b2(j,jj)));
ann(j,jj)=ann(j,jj)+(((nu21*f1*(t^3)*st/(12*(r^3)*R*zeta)
-nu21*f1*t*st*((t2*st)^2)/(2*(r^3)*R*zeta)
-nu21*f1*(t^2)*ct*(-t2*st)/(2*(r^3)*R*zeta)
-nu21*f1*(t^2)*st*(-t2*ct)/(4*(r^3)*R*zeta)
-nu21*f1*(t^3)*(st^3)/(12*nu12*r*(R^3)*(zeta^3))
-f1*(t^2)*(st^2)*(-t2*st)*nu21/(4*(r^2)*(R^2)*(zeta^2)*nu12)
-f1*(t^3)*st*ct*nu21/(6*(r^2)*(R^2)*(zeta^2)*nu12))*b1(j,jj)
+(-nu21*f1*(t^3)*ct/(12*(r^3)*R*zeta)-nu21*f1*(t^2)*st*(-
t2*st)/(4*(r^3)*R*zeta)+(t^2)*(-t2*ct)*f1/(4*(r^4))
+f1*((t2*st)^2)*t/(2*(r^4))-f1*(t^3)*(st^2)*nu21/(nu12*12*(r^2)*(R^2)*(zeta^2))
-f1*(t^2)*(-t2*st)*st/(2*(r^3)*R*zeta)
-f1*(t^3)*ct/(12*(r^3)*R*zeta))*b2(j,jj)
+(f1*(t^2)*(-t2*st)/(2*(r^4))-f1*(t^3)*st/(6*(r^3)*R*zeta))*b3(j,jj)
+(f1*(t^3)/(12*(r^4)))*b4(j,jj))+((nu21*f1*(t^2)*(-t2*st)/(2*(r^4)*(zeta^2))
+nu21*f1*(t^3)*st/(6*(r^3)*(zeta^3)*R)
+G12*(t^3)*(-t2*st)/(2*(r^4)*(zeta^2))
+G12*(t^3)/(6*(r^3)*R*(zeta^3)))*(-
(k^2))*b1(j,jj)+(nu21*f1*(t^3)/(6*(r^4)*(zeta^2))
+G12*(t^3)/(6*(r^4)*(zeta^2)))*(-(k^2))*b2(j,jj)));
end
end
clear j; clear jj
```

```
%------------------------------------------------------------------------------
% section bb-bd - enforce the fixed boundary conditions on two end points
bb=zeros(8);bd=zeros(8,3*n-8);
%
for i=1:8; bb(i,i)=1; end; clear i;
% eqns 6 7 for slope condition
bb(6,5)=b1(2,1);     bb(7,5)=b1(n-1,1);
bb(6,6)=b1(2,2);     bb(7,6)=b1(n-1,2);
bb(6,7)=b1(2,n-1);   bb(7,7)=b1(n-1,n-1);
bb(6,8)=b1(2,n);     bb(7,8)=b1(n-1,n);
%
for i=1:n-4;
bd(6,2*n-4+i)=b1(2,i+2);
bd(7,2*n-4+i)=b1(n-1,i+2);
end
clear i;
%------------------------------------------------------------------------------
% section db-dd - enforce the governing equations in the domain
dd=zeros(3*n-8);db=zeros(3*n-8,8);
%
for i=1:n-2;
db(i,1)=aii(i+1,1);       db(i,5)=ain(i+1,1);
db(i,2)=aii(i+1,n);       db(i,6)=ain(i+1,2);
db(i,3)=aim(i+1,1);       db(i,7)=ain(i+1,n-1);
db(i,4)=aim(i+1,n);       db(i,8)=ain(i+1,n);
%
db(i+n-2,1)=ami(i+1,1);   db(i+n-2,5)=amn(i+1,1);
db(i+n-2,2)=ami(i+1,n);   db(i+n-2,6)=amn(i+1,2);
db(i+n-2,3)=amm(i+1,1);   db(i+n-2,7)=amn(i+1,n-1);
db(i+n-2,4)=amm(i+1,n);   db(i+n-2,8)=amn(i+1,n);
end
clear i;
%
for i=1:n-4;
db(i+2*(n-2),1)=ani(i+2,1);      db(i+2*(n-2),5)=ann(i+2,1);
db(i+2*(n-2),2)=ani(i+2,n);      db(i+2*(n-2),6)=ann(i+2,2);
db(i+2*(n-2),3)=anm(i+2,1);      db(i+2*(n-2),7)=ann(i+2,n-1);
db(i+2*(n-2),4)=anm(i+2,n);      db(i+2*(n-2),8)=ann(i+2,n);
end
clear i;
%
for i=1:n-2; for j=1:n-2
dd(i,j)         =aii(1+i,1+j);     dd(n-2+i,j)     =ami(1+i,1+j);
dd(i,n-2+j)     =aim(1+i,1+j);     dd(n-2+i,n-2+j)=amm(1+i,1+j);
end;end
%
for i=1:n-2; for j=1:n-4
dd(i,2*(n-2)+j)     =ain(i+1,j+2);  dd(2*(n-2)+j,i)     =ani(j+2,i+1);
dd(n-2+i,2*(n-2)+j)=amn(i+1,j+2);  dd(2*(n-2)+j,n-2+i)=anm(j+2,i+1);
end;end
clear i; clear j;
%
for i=1:n-4; for j=1:n-4
dd(2*(n-2)+i,2*(n-2)+j)=ann(2+i,2+j);
end;end
clear i; clear j;
prod1=-inv(bb)*bd; clear bb bd;
```

```
prod2=db*prod1; clear db prod1;
stiff=dd+prod2; clear dd prod2;
%--------------------------------------------------------------------------------
% section loadv - set up the load vector for the right hand side
rhss=zeros(3*n-8,1);
rhss(75,1)=-p0*sin(k1*beta)/(k1*pi);
rhss(76,1)=-2*p0*sin(k1*beta)/(k1*pi);
rhss(77,1)=-2*p0*sin(k1*beta)/(k1*pi);
rhss(78,1)=-2*p0*sin(k1*beta)/(k1*pi);
rhss(79,1)=-p0*sin(k1*beta)/(k1*pi);
%--------------------------------------------------------------------------------
% section solu - combine submat - solv matrix eqn - find results
% find the vector of displacement components rslt
clear aii;  clear aim; clear ain; clear ami; clear amm; clear amn; clear ani;
clear anm; clear ann;
clear test2; clear test1; clear pval;
rslt=stiff\rhss; clear stiff;  clear rhss;
%
% displacement components at the sampling points, u,v
u0=zeros(n,1); v0=zeros(n,1); w0=zeros(n,1);
for i=1:n-2
u0(i+1,1)=rslt(i,1);  v0(i+1,1)=rslt(i+n-2,1);
end;
for j=1:n-4
w0(j+2,1)=rslt(2*n-4+j,1);
end
clear rslt;
for ii=1:n
ut(ii,1)=ut(ii,1)+u0(ii,1)*cos(k1*theta);
vt(ii,1)=vt(ii,1)+v0(ii,1)*sin(k1*theta);
wt(ii,1)=wt(ii,1)+w0(ii,1)*cos(k1*theta);
vt1(ii,1)=vt1(ii,1)+k*v0(ii,1)*cos(k1*theta);
wt2(ii,1)=wt2(ii,1)-(k^2)*w0(ii,1)*cos(k1*theta);
end
end
%--------------------------------------------------------------------------------
% section stress
sigphio=zeros(n,1); sigphii=zeros(n,1); sigetao=zeros(n,1); sigetai=zeros(n,1);
for j=1:n
tee=x(j); st=sin(tee); ct=cos(tee); zeta=1+gammar*ct; t=t1+t2*ct;
ephi1=wt(j,1)/r;
eeta1=(-gammar*st*ut(j,1)+gammar*ct*wt(j,1)+vt1(j,1))/(r*zeta);
kphi1=0;
keta1=-st*ut(j,1)/(r*R*zeta)+ct*vt1(j,1)/(r*R*(zeta^2))-wt2(j,1)/((r*zeta)^2);
for jj=1:n
ephi1=ephi1+ut(jj,1)*b1(j,jj)/r;
kphi1=kphi1+(ut(jj,1)*b1(j,jj)-wt(jj,1)*b2(j,jj))/(r^2);
keta1=keta1+st*wt(jj,1)*b1(j,jj)/(r*R*zeta);
end
sigphio(j)=f1*(ephi1+nu21*eeta1)+(t/2)*f1*(kphi1+nu21*keta1);
sigphii(j)=f1*(ephi1+nu21*eeta1)-(t/2)*f1*(kphi1+nu21*keta1);
sigetao(j)=f1*nu21*ephi1+(f1*nu21/nu12)*eeta1+(t/2)*(f1*nu21*kphi1+(f1*nu21/nu1
2)*keta1);
sigetai(j)=f1*nu21*ephi1+(f1*nu21/nu12)*eeta1-
(t/2)*(f1*nu21*kphi1+(f1*nu21/nu12)*keta1);
end
%--------------------------------------------------------------------------------
```

```
% section results
sigphio
wt
```